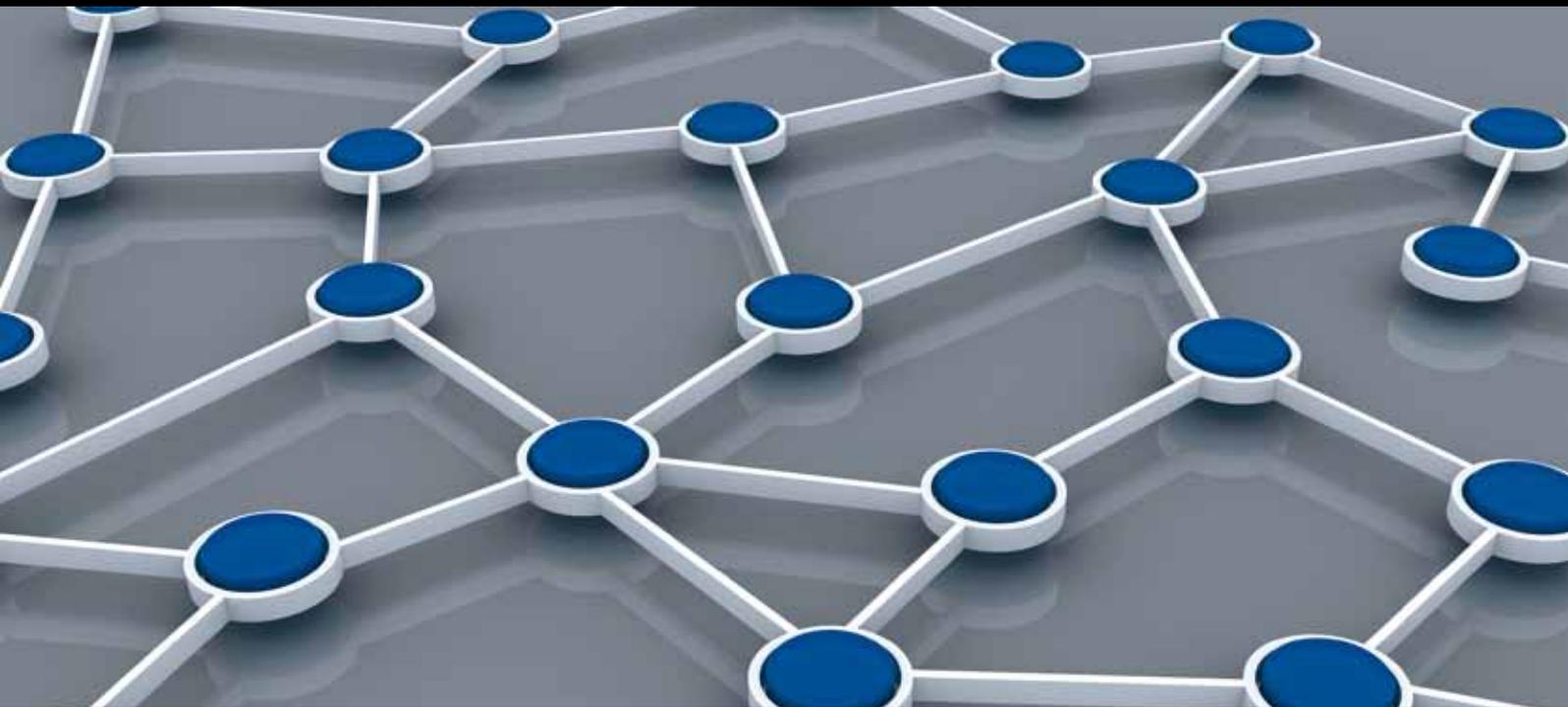


SMART SENSOR NETWORKS: THEORY AND PRACTICE

GUEST EDITORS: YUHANG YANG AND BAHRAM HONARY





Smart Sensor Networks: Theory and Practice

International Journal of Distributed Sensor Networks

Smart Sensor Networks: Theory and Practice

Guest Editors: Yuhang Yang and Bahram Honary



Copyright © 2012 Hindawi Publishing Corporation. All rights reserved.

This is a special issue published in “International Journal of Distributed Sensor Networks.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editorial Board

Prabir Barooah, USA
Richard R. Brooks, USA
Stefano Chessa, Italy
W.-Y. Chung, Republic of Korea
George P. Efthymoglou, Greece
Frank Ehlers, Italy
Paola Flocchini, Canada
Yunghsiang S. Han, Taiwan
Tian He, USA
Baoqi Huang, Australia
Chin-Tser Huang, USA
S. S. Iyengar, USA
Rajgopal Kannan, USA
Miguel A. Labrador, USA
Joo-Ho Lee, Japan
Shijian Li, China
Yingshu Li, USA
Shuai Li, USA

Minglu Li, China
Jing Liang, China
Weifa Liang, Australia
Wen-Hwa Liao, Taiwan
Alvin S. Lim, USA
Zhong Liu, China
Donggang Liu, USA
Yonghe Liu, USA
Seng Loke, Australia
Jun Luo, Singapore
J. R. Martinez-deDios, Spain
Shabbir N. Merchant, India
Aleksandar Milenkovic, USA
Eduardo F. Nakamura, Brazil
Peter Csaba Ölveczky, Norway
M. Palaniswami, Australia
Shashi Phoha, USA
Cristina M. Pinotti, Italy

Hairong Qi, USA
Joel Rodrigues, Portugal
Jorge Sa Silva, Portugal
Sartaj K. Sahni, USA
Weihua Sheng, USA
Zhi Wang, China
Sheng Wang, China
Andreas Willig, New Zealand
Qishi Wu, USA
Qin Xin, Norway
Jianliang Xu, Hong Kong
Yuan Xue, USA
Fan Ye, USA
Ning Yu, China
Tianle Zhang, China
Yanmin Zhu, China

Contents

Smart Sensor Networks: Theory and Practice, Yuhang Yang and Bahra Honary
Volume 2012, Article ID 602893, 2 pages

Enhancing Sink-Location Privacy in Wireless Sensor Networks through k -Anonymity, Guofei Chai, Miao Xu, Wenyuan Xu, and Zhiyun Lin
Volume 2012, Article ID 648058, 16 pages

Performance Analysis of Flow-Based Traffic Splitting Strategy on Cluster-Mesh Sensor Networks, Huimin She, Zhonghai Lu, Axel Jantsch, Dian Zhou, and Li-Rong Zheng
Volume 2012, Article ID 232937, 17 pages

An Integrated Approach to the Design of Wireless Sensor Networks for Structural Health Monitoring, Fabio Federici, Fabio Graziosi, Marco Faccio, Andrea Colarieti, Vincenzo Gattulli, Marco Lepidi, and Francesco Potenza
Volume 2012, Article ID 594842, 16 pages

DI-GEP: A New Lifetime Extending Algorithm for Target Tracking in Wireless Sensor Networks, Shucheng Dai, Chuan Li, and Chun Chen
Volume 2012, Article ID 467497, 9 pages

Enabling Collaborative Musical Activities through Wireless Sensor Networks, Santiago J. Barro, Tiago M. Fernández-Caramés, and Carlos J. Escudero
Volume 2012, Article ID 314078, 13 pages

One-Time Broadcast Encryption Schemes in Distributed Sensor Networks, Pawel Szalachowski and Zbigniew Kotulski
Volume 2012, Article ID 536718, 9 pages

Error-Tolerant and Energy-Efficient Coverage Control Based on Biological Attractor Selection Model in Wireless Sensor Networks, Takuya Iwai, Naoki Wakamiya, and Masayuki Murata
Volume 2012, Article ID 971014, 14 pages

SWR: Smartness Provided by Simple, Efficient, and Self-Adaptive Algorithm, Mujdat Soyuturk and Deniz Turgay Altılar
Volume 2012, Article ID 289683, 21 pages

An Experimental Study of WSN Power Efficiency: MICAz Networks with XMesh, Tyler W. Davis, Xu Liang, Miguel Navarro, Diviyansh Bhatnagar, and Yao Liang
Volume 2012, Article ID 358238, 14 pages

Wireless Sensor Network for Environmental Monitoring: Application in a Coffee Factory, J. Valverde, V. Rosello, G. Mujica, J. Portilla, A. Uriarte, and T. Riesgo
Volume 2012, Article ID 638067, 18 pages

A Mutual Algorithm for Optimizing Distributed Source Coding in Wireless Sensor Networks, Nashat Abughalieh, Kris Steenhaut, Bart Lemmens, and Ann Nowé
Volume 2012, Article ID 783798, 9 pages

Interference-Free Wakeup Scheduling with Consecutive Constraints in Wireless Sensor Networks,

Junchao Ma and Wei Lou

Volume 2012, Article ID 525909, pages

Novel Energy-Efficient Miner Monitoring System with Duty-Cycled Wireless Sensor Networks,

Peng Guo, Tao Jiang, and Kui Zhang

Volume 2012, Article ID 975082, 9 pages

Survey: Discovery in Wireless Sensor Networks, Valerie Galluzzi and Ted Herman

Volume 2012, Article ID 271860, 12 pages

Subjective Logic-Based Anomaly Detection Framework in Wireless Sensor Networks, Jinhui Yuan,

Hongwei Zhou, and Hong Chen

Volume 2012, Article ID 482191, 13 pages

A Self-Organized and Smart-Adaptive Clustering and Routing Approach for Wireless Sensor Networks,

Kyuhong Lee and Heesang Lee

Volume 2012, Article ID 156268, 13 pages

Distributed Algorithm for Real-Time Energy Optimal Routing Based on Dual Decomposition of Linear Programming, Jiří Trdlička and Zdeněk Hanzálek

Volume 2012, Article ID 346163, 13 pages

MAC Protocols Used by Wireless Sensor Networks and a General Method of Performance Evaluation,

Joseph Kabara and Maria Calle

Volume 2012, Article ID 834784, 11 pages

Information Fusion-Based Storage and Retrieve Algorithms for WSNs in Disaster Scenarios, Zhe Xiao,

Ming Huang, Jihong Shi, Wenwei Niu, and Jingjing Yang

Volume 2012, Article ID 524543, 16 pages

A Mobile Computing Framework for Pervasive Adaptive Platforms, Olivier Brousse, Jérémie Guillot,

Gilles Sassatelli, Thierry Gil, François Grize, and Michel Robert

Volume 2012, Article ID 193864, 15 pages

Analysis of Mobility and Sharing of WSNs By IP Applications, Dennis J. A. Bijwaard, Paul J. M. Havinga, and Henk Eertink

Volume 2012, Article ID 923594, 14 pages

Localization Algorithm Based on Maximum a Posteriori in Wireless Sensor Networks, Kezhong Lu,

Xiaohua Xiang, Dian Zhang, Rui Mao, and Yuhong Feng

Volume 2012, Article ID 260302, 7 pages

Adaptive WSN Scheduling for Lifetime Extension in Environmental Monitoring Applications,

Jong Chern Lim and Chris Bleakley

Volume 2012, Article ID 286981, 17 pages

Network-Coding-Based Cooperative ARQ Medium Access Control Protocol for Wireless Sensor Networks, Angelos Antonopoulos and Christos Verikoukis

Volume 2012, Article ID 601321, 9 pages

The Complexity of the Minimum Sensor Cover Problem with Unit-Disk Sensing Regions over a Connected Monitored Region, Ren-Song Ko

Volume 2012, Article ID 918252, 25 pages

Robust Interval-Based Localization Algorithms for Mobile Sensor Networks, Farah Mourad,

Hichem Snoussi, Michel Kieffer, and Cédric Richard

Volume 2012, Article ID 303895, 7 pages

Ion-6: A Positionless Self-Deploying Method for Wireless Sensor Networks, Shih-Chang Huang

Volume 2012, Article ID 940920, 10 pages

Triangular Energy-Saving Cache-Based Routing Protocol by Energy Sieving, Chiu-Ching Tuan and

Yi-Chao Wu

Volume 2012, Article ID 602159, 11 pages

Intelligent Collaborative Event Query Algorithm in Wireless Sensor Networks, Rongbo Zhu

Volume 2012, Article ID 728521, 11 pages

Editorial

Smart Sensor Networks: Theory and Practice

Yuhang Yang¹ and Bahram Honary²

¹ School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

² School of Computing and Communications, Lancaster University, Lancaster LA1 4WA, UK

Correspondence should be addressed to Yuhang Yang, yhyangsjtu@gmail.com

Received 4 June 2012; Accepted 4 June 2012

Copyright © 2012 Y. Yang and B. Honary. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless sensor networks (WSNs) are a new and emerging type of networks. The ubiquitous nature of WSNs has led to a growing interest in rendering them smart and autonomous, which have the potential to enable a large class of applications in multiple fields such as smart power grids, smart health care, smart buildings, and smart industrial process. In spite of the increasing demand for smart services, we are still lacking a clear understanding of analytical and computational techniques, as well as best practices, to design resource allocation schemes, power efficient protocols, and self-organization algorithms for WSNs. Despite recent advances, dynamic spectrum access, cognitive radios, ultra-wideband and cooperative communications, among other techniques, will have a profound impact on smart sensor networks and applications. The objective of this special issue is to present a collection of high-quality research papers that report the latest research advances of WSNs in the area of smart technologies and applications.

There were 107 papers submitted for consideration for publication in this special issue. After two rounds of rigorous peer review and revision processes, only 30 papers were selected for publication. The acceptance rate is 28%. Unfortunately, not all of the excellent papers can be selected into this special issue due to the limited space available. The low acceptance rate reflects the very competitive selection process, and we believe that these papers included in this special issue represent the outstanding quality of the research outcomes in this field. The papers included in this special issue are categorized into the following areas.

The six papers selected in the group have proposed several new methods for multidisciplinary design and optimisation of smart protocols. The papers “A self-organized and smart-adaptive clustering and routing approach for wireless sensor networks,” “Distributed algorithm for real-time

energy optimal routing based on dual decomposition of linear programming,” “triangular energy-saving cache-based routing protocol by energy Sieving”, and “SWR: smartness provided by simple, efficient, and self-adaptive algorithm”, concentrate on smart routing algorithm and protocol design, and the papers “MAC protocols used by wireless sensor networks and a general method of performance evaluation” and “network-coding-based cooperative ARQ medium access control protocol for wireless sensor networks” explore how to design efficient MACs protocols.

Power consumption is a key issue in WSNs. The five papers selected in the group propose several emerging schemes for designing smart power-saving methods. The papers “DI-GEP: a new lifetime extending algorithm for target tracking in wireless sensor Networks” and “adaptive WSN scheduling for lifetime extension in environmental monitoring applications” focus on lifetime extension. The paper “An experimental study of WSN power efficiency: MICAz networks with XMesh” provides the experimental results of power efficiency. The paper “Interference-free wakeup scheduling with consecutive constraints in wireless sensor networks” proposes a new wakeup scheme to reduce power consumption. The survey paper “Survey: discovery in wireless sensor networks” reviews recent progress on the problems of neighbour discovery for WSNs.

Smart coverage and location schemes are very important techniques in WSNs. The three papers selected into this group, “Error-tolerant and energy-efficient coverage control based on biological attractor selection model in wireless sensor networks,” “The complexity of the minimum sensor cover problem with unit-disk sensing regions over a connected Monitored Region” and “Ion-6: A Positionless Self-Deploying Method for Wireless Sensor networks,” propose several efficient design and optimisation schemes for intelligent coverage.

The papers “*Robust interval-based localization algorithms for mobile sensor networks*” and “*Localization algorithm based on maximum a posteriori in wireless sensor networks*” study the smart location algorithm in WSN.

In the fourth group, five papers, “*Information fusion-based storage and retrieve algorithms for WSNs in disaster scenarios*,” “*An efficient data-gathering scheme for heterogeneous sensor networks via mobile sinks*,” “*Intelligent collaborative event query algorithm in wireless sensor networks*,” “*A mutual algorithm for optimizing distributed source coding in wireless sensor networks*” and “*performance analysis of flow-based traffic splitting strategy on cluster-mesh sensor networks*” discuss data processing and performance optimization in WSNs.

Security and privacy is another key problem in WSNs; in this group, three papers “*Enhancing sink-location privacy in wireless sensor networks through k-anonymity*,” “*One-time broadcast encryption schemes in distributed sensor networks*” and “*Subjective logic-based anomaly detection framework in wireless sensor networks*” propose several schemes to improve the security performance in WSNs.

In the last group, there are six papers exploring the smart applications based on WSNs. Three papers, “*An integrated approach to the design of wireless sensor networks for structural health monitoring*,” “*Novel energy-efficient miner monitoring system with duty-cycled wireless sensor networks*” and “*Wireless sensor network for environmental monitoring: application in a coffee factory*,” discuss the monitoring systems based on WSNs in different cases. The paper “*A mobile computing framework for pervasive adaptive platforms*” and “*Analysis of mobility and sharing of WSNs by IP applications*” study mobile applications based on WSNs. The paper “*Enabling collaborative musical activities through wireless sensor networks*” proposes the use of an optimized WSN network for interconnecting MIDI (musical instrument digital interface) devices.

The thirty papers included in this special issue touch on six different topics. They reflect the diversity and the richness of smart WSNs research activities and applications. We hope that this special issue can help readers to get a better understanding about the breadth and depth of current research. We also hope that this special issue can boost further related research and technology improvements in the field of WSNs.

Acknowledgments

The Guest Editors would like to thank all the authors for their contributions. And special thanks go to all reviewers for their great effort, timely responses, and constructive comments and suggestions. Thanks also go to Professor Sundaraja Sitharama Iyengar, the Editor-in-Chief of International Journal of Distributed Sensor Networks (IJDSN), and the journal editorial staff who helped us throughout the entire process.

Yuhang Yang
Bahram Honary

Research Article

Enhancing Sink-Location Privacy in Wireless Sensor Networks through k -Anonymity

Guofei Chai,¹ Miao Xu,² Wenyuan Xu,² and Zhiyun Lin¹

¹College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China

²Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208, USA

Correspondence should be addressed to Wenyuan Xu, wyxu@cse.sc.edu

Received 23 May 2011; Revised 5 January 2012; Accepted 7 January 2012

Academic Editor: Yuhang Yang

Copyright © 2012 Guofei Chai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the shared nature of wireless communication media, a powerful adversary can eavesdrop on the entire radio communication in the network and obtain the contextual communication statistics, for example, traffic volumes, transmitter locations, and so forth. Such information can reveal the location of the sink around which the data traffic exhibits distinctive patterns. To protect the sink-location privacy from a powerful adversary with a global view, we propose to achieve k -anonymity in the network so that at least k entities in the network are indistinguishable to the nodes around the sink with regard to communication statistics. Arranging the location of k entities is complex as it affects two conflicting goals: the routing energy cost and the achievable privacy level, and both goals are determined by a nonanalytic function. We model such a positioning problem as a nonlinearly constrained nonlinear optimization problem. To tackle it, we design a generic-algorithm-based quasi-optimal (GAQO) method that obtains quasi-optimal solutions at quadratic time. The obtained solutions closely approximate the optima with increasing privacy requirements. Furthermore, to solve k -anonymity sink-location problems more efficiently, we develop an artificial potential-based quasi-optimal (APQO) method that is of linear time complexity. Our extensive simulation results show that both algorithms can effectively find solutions hiding the sink among a large number of network nodes.

1. Introduction

With the increasing advances of sensing devices and wireless technology, wireless sensor networks (WSNs) have been interwoven into the fabric of our daily life. In particular, WSNs have been deployed to monitor personal health, track targets, and sense pollutants. Those sensor networks typically consist of many resource-constrained sensor nodes and one sink. Each sensor node monitors the underlying physical phenomenon and reports the measurements to the sink in a multihop manner.

In spite of their popularity, the viability and success of those sensor networks hinge on a variety of security and privacy threats. One of the most challenging threats is location privacy, since it cannot be addressed by traditional cryptographic mechanisms [1]. Due to the shared nature of wireless communication media, an attacker can easily eavesdrop on the radio communication either by purchasing her own sensor devices or by leveraging other radio devices capable of

monitoring message transmission. Thus, no matter whether messages are encrypted or not, an adversary is able to identify contextual information: where the communication has occurred and who has participated in communication, without accessing the content of messages. For example, an adversary can identify the sender of a message by analyzing the angle of arrival [2], or he can determine the receiver in the similar fashion when the receiver relays a message [3].

Since an adversary can locate both the origin and destination of messages (i.e., sinks) purely by observing the contextual information, the WSN location privacy problem can be divided into two categories: *source*-location privacy and *sink*-location privacy. The source-location privacy problem is concerned with preventing attackers from discovering the locations of message sources, which may reveal sensitive position information of assets being monitored, for example, endangered animals. Much effort has been devoted to preserve source-location privacy against a wide variety of attackers, ranging from resource-constrained attackers [2]

to powerful attackers that have a global view of network communications [1, 4].

In this study, we focus on preserving *sink*-location privacy against attackers with a *global* view. The sink node serves as the aggregating point for data collection and is crucial to assure the availability of a WSN. If the sink node is located and destroyed, the sensed data can no longer be relayed to a data center, rendering the entire WSN useless. Despite the great importance of sink node, the sink-location privacy problem has only been studied under the assumption of resource-constrained attackers [3, 5–8]. When a global adversary is involved, those strategies for resource-constrained attackers become inapplicable. Our work aims to fill in the absence in defending against powerful global adversaries.

To achieve the global view, an attacker can either deploy her own sensors [1, 4] or utilize powerful radio receivers with extremely sensitive antennas to pick up communications across the whole network [1]. As such, a global attacker can derive the location of sinks either by traffic-analysis attacks [5] or packet-tracing attacks [2, 3]. Traffic-analysis attacks utilize the fact that the closer a node is located towards the WSN sink, the higher the number of messages it needs to forward. Thus, moving towards a spot that exhibits a higher message volume can eventually lead the adversaries to find the sink. Packet-tracing attacks lead the adversary toward the travel direction of messages hop by hop till he reaches the sink.

Both traffic-analysis attacks and packet-tracing attacks require no access to the message content but message existence. Additionally, a global adversary can identify *every* node that has forwarded a message instantly, while most literature [4, 9] assumes that an adversary with a *local* view can only identify the sender when communication occurs within his observable range. We are unaware of any solutions that can defend against a global adversary, since it is virtually impossible to protect the network against a global eavesdropper [10]. Any local obfuscation created by fake messages cannot confuse a global adversary. For instance, fractional propagation [5] forks a fake message toward a random destination while the real message is forwarded towards the sink, which is likely to mislead an adversary with a local view. However, such an approach cannot deceive the adversary with a global view, since all real messages always arrive at the sink.

One naive defense strategy is to have each node send the same volume of messages as the sink (including both real and fake messages). However, such strategy imposes high energy consumption and is infeasible. To limit the energy conception while enhancing the privacy against a powerful adversary with a global view, we propose to achieve *k*-anonymity in the network so that at least *k* entities exhibit the same characteristics as the nodes located close to the sink. As such, they are indistinguishable even to the powerful attackers with regard to contextual communication information.

The concept of achieving *k*-anonymity [11] was originally proposed to protect personal identity while releasing person-specific data and has been studied extensively in the field of database and data mining. To our best knowledge, our work is the first attempt to apply this concept to

preserving sink-location privacy in wireless sensor networks, and there are no other valid approaches dealing with the attacks of a global adversary. We summarize our contribution as follows.

- (i) We identify the absence of defense strategies to enhance sink-location privacy against global adversaries.
- (ii) To enhance sink-location privacy, we propose to achieve *k*-anonymity via an Euclidean minimum-spanning tree-based routing protocol, that is, create *k* designated nodes in the network.
- (iii) We show that positioning *k* designated nodes is complex as it affects two conflicting goals: the routing energy cost and the achievable privacy level, and both goals are determined by a non-analytic function. To strike a balance between those two goals, we formulate the problem of *k*-anonymity routing protocols as a nonlinearly constrained optimization problem.
- (iv) The nonlinearly constrained optimization problem is extremely challenging to solve. To tackle the problem, we design two quasi-optimal algorithms that can obtain the *k*-node locations closely approximating the optima, and our extensive simulations validate that both algorithms can effectively find solutions hiding the sink among a large number of network nodes.

The rest of the paper is organized as the following. In Section 2, we describe the network model, attack model, and formalize the problem of achieving *k*-anonymity as a nonlinear optimization problem. We present the routing algorithm for achieving *k*-anonymity in Section 3. In Section 4, we discuss two approximate algorithms that can obtain quasi-optimal solutions and show our validation effort. Finally, we discuss related work in Section 5 and provide concluding remarks in Section 6.

2. Problem Overview

A wide variety of WSNs have emerged as monitoring and controlling solutions for numerous applications. It is very hard, if even possible, to design a solution applicable to all types of WSNs and to address all attacks. In this section, we specify a popular type of WSNs, which were adopted by several work [12–16]. We formalize the problem below.

2.1. Network Model. We consider a network of wireless sensor nodes that is distributed throughout a bounded environment $Q \subset \mathbb{R}^2$ at positions n_1, n_2, \dots , and we denote

$$\mathcal{N} = \{n_i \mid i \in I\}, \quad (1)$$

where n_i is indexed using an index set I . The network has the following features.

2.1.1. Periodic Data Reporting. WSNs can be classified as event-driven or periodic. In an event-driven sensor network, only those sensors that have observed events will generate

and deliver messages to sinks in a multihop manner while others remain silent. In a periodic network, each sensor will measure the underlying physical phenomena and will deliver its measurements periodically to sinks. We focus on periodic networks since in such networks, even aggregation cannot eliminate the data traffic accumulation towards the sink [9]. Further, we assume that no aggregation algorithms are applied to the networks.

2.1.2. Homogeneous Network with One Sink. We consider homogeneous sensor networks that consist of sinks and a large number of sensor nodes and are densely deployed in a square. Each sensor node is equipped with an omnidirectional antenna and transmits at the same transmission power level. Without loss of generality, we assume that one sink in the network collects data. We note that our scheme can be easily extended to a network with multiple sinks.

2.1.3. No ACK. We assume that the sensor networks do not rely on acknowledgement packets (ACKs) to achieve reliable communication, since the excessive number of ACKs transmitted by the sink will easily reveal its location. We assume that the sink only passively receives messages. Thus, the sink is hidden, and the adversary cannot pinpoint the location of the sink purely by relying on eavesdropping on ACKs.

2.1.4. End-to-End Data Encryption. We assume that messages are protected by an end-to-end encryption protocol using pairwise keys [17]. Due to the limitation of constrained resources, we do not consider the case where the messages are decrypted and re-encrypted at each hop. Therefore, a message exhibits the same cipher as it travels from the source to the sink.

2.2. Attack Model. We consider a powerful attacker who is able to eavesdrop on all communications across the whole network. The adversary does not actively interfere with regular communications in the network but passively eavesdrops on network communications. Her goal is to find the location of the sink and to compromise the sink via physical contacts. Additionally, according to Kerckhoffs' Principal [18], we assume the adversary is aware of all protocols being used but does not know the established keys of the network and is unable to decrypt messages.

To find the sink physically, the adversary will perform a two-phase search: (1) the *location-mining* phase and (2) the *visual searching* phase. In the location-mining phase, the adversary eavesdrops on the network traffic and identifies a set of nodes that appear to be close to the sink. Given the information on nearby nodes, the adversary will find the sink physically in the visual searching phase.

2.2.1. Phase I: Location Mining. Let m_p be the p th message in the network. When m_p is forwarded from its originator n_{p1} to the sink, the attacker will record a set of communication events represented by three tuples: $\{(m_p, n_{pq}, t_{pq}) \mid q = 1, \dots, h_p\}$, where h_p is the number of hops that m_p has travelled and each three-tuple (m_p, n_{pq}, t_{pq}) maps to an event that the sensor node located at n_{pq} forwards m_p

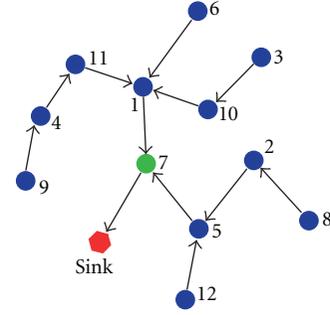


FIGURE 1: An example of a routing tree rooted at the sink. Each arrow points from a child to a parent.

at time t_{pq} . Up to time t , the adversary will obtain the communication event set $M(t) = \{(m_p, n_{pq}, t_{pq}) \mid p \in P, q \in H_p, \max(t_{pq}) \leq t\}$, where P and H_p are index sets of messages and the hop counts for the messages, respectively.

Given $M(t)$, the adversary will perform statistical analysis on message transmission information. Formally, let \mathcal{M} denote the space of all communication events. We describe the statistical analysis method as a composite function $\pi = \psi \circ \rho$: the function ρ maps the communication events to l traffic statistics associated with every node, and the function ψ selects the set of nodes who have unusual traffic statistics. That is,

$$\begin{aligned} \rho : \mathcal{M} &\longrightarrow \mathbb{R}^{|\mathcal{N}|^l}, \\ \psi : \mathbb{R}^{|\mathcal{N}|^l} &\longrightarrow 2^{\mathcal{N}}, \end{aligned} \quad (2)$$

where $|\cdot|$ is the cardinality of a set and $2^{\mathcal{N}}$ is the power set of \mathcal{N} .

We consider a powerful attacker who is able to perform traffic-analysis attacks and traffic-tracing attacks. Particularly, he is able to obtain two traffic statistics (e.g., $l = 2$): the traffic volume $\rho_v^{n_i}$ and the number of messages $\rho_e^{n_i}$ that end at a node n_i . Assume the attacker starts to record communication events at time $t = 0$, and he can obtain the following statistics at time t :

$$\rho_v^{n_i}(M(t)) = \frac{|\{(m_p, n_{pq}, t_{pq}) \mid n_{pq} = n_i, t_{pq} \leq t\}|}{t}, \quad (3)$$

$$\rho_e^{n_i}(M(t)) = \frac{|\{(m_p, n_{ph_p}, t_{ph_p}) \mid n_{ph_p} = n_i, t_{ph_p} \leq t\}|}{t}, \quad (4)$$

where h_p is the hop count for the message m_p . Given $\{\rho_v^{n_i}\}$ and $\{\rho_e^{n_i}\}$, the adversary can identify nodes that have either the maximum traffic volume or the maximum number of messages ending here:

$$\psi(\dots, \rho_v^{n_i}, \rho_e^{n_i}, \dots) = \left\{ \arg \max_{n_i \in \mathcal{N}} (\rho_v^{n_i}) \right\} \cup \left\{ \arg \max_{n_i \in \mathcal{N}} (\rho_e^{n_i}) \right\}. \quad (5)$$

Consider the example depicted in Figure 1, where a tree-based routing protocol is used and a routing tree is formed

with the sink node serving as the root of the tree. After one reporting period t , the adversary will conclude that $\pi(M(t)) = \{n_7\}$, since n_7 transmits 12 messages per period, one for each node.

2.2.2. Phase II: Visual Searching. Although $\pi(M(t))$ only identifies the nodes that are close to the sink and does not pinpoint the sink's location, it does help the adversary to refine the region \mathcal{S} where the sink resides. To find the sink physically, the adversary needs to search \mathcal{S} either visually or using equipment such as a metal detector. Assume the adversary is able to search an area of size ν per second and the area of \mathcal{S} is $A_{\mathcal{S}}$, then the amount of time required for the adversary to identify the sink physically is at most $A_{\mathcal{S}}/\nu$.

Continuing with the example depicted in Figure 1, $\pi(M(t))$ only contains a node n_7 . The region \mathcal{S} is the communication range of n_7 with a size A_c . The amount of time required for the adversary to find the sink is at most A_c/ν .

2.3. k -Anonymity. Our goal is to design a routing strategy that can enhance sink-location privacy. Essentially, the risk of breaching the sink-location privacy is caused by the observable asymmetric traffic pattern of the sensor networks. The message traffic volume is the largest at the nodes close to the sink, and the travel paths of messages always end there as well. The basic idea of our approach is to change the traffic pattern such that at least k nodes located at $p_1, p_2, \dots, p_k, \dots$ may be far away from the sink but behave the same as the nodes around the sink; namely,

$$\pi(M(t)) = \{p_1, \dots, p_k, \dots\}. \quad (6)$$

In particular, we envision that each message is delivered to the sink prior to its last-hop transmission, and thus messages no longer end at the nodes around the sink. Further, a lot more nodes send high volumes of messages other than the ones around the sink. As a result, $|\pi(M(t))| \gg 1$.

The main design goal of the k -anonymity routing protocol is to enhance sink-location privacy, and it should also deliver messages without incurring high energy overhead. Therefore, we define a privacy measure and a network efficiency metric to evaluate a routing strategy.

- (1) The safety period Φ is the *average* amount of time taken for a global attacker to find the sink physically. We use the safety period Φ to quantify the privacy level. A larger safety period maps to a higher level of sink-location privacy. The safety period Φ includes the amount of time needed for *location mining* and for *visual searching*. Because the duration for *location mining* is fixed and short, we consider the safety period equals the duration of *visual searching*. Since at least k nodes located at p_1, \dots, p_k, \dots exhibit the same traffic statistics, the adversary has to visually search all the communication ranges of these nodes. Thus, the safety period is a function of p_1, \dots, p_k, \dots , denoted by $\Phi(p_1, \dots, p_k, \dots)$.
- (2) The energy cost E is the average amount of energy consumed for transmitting one message from each

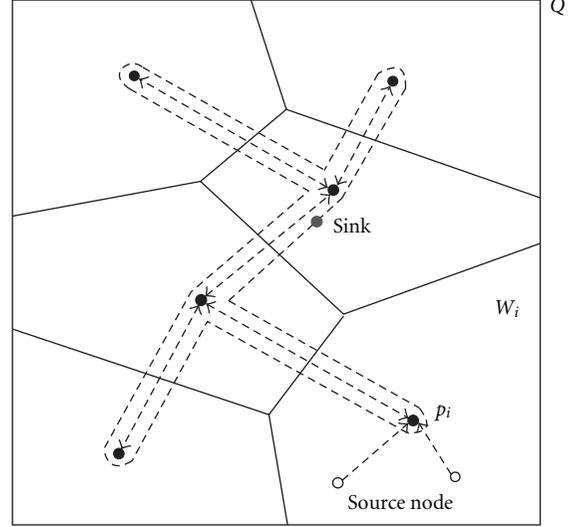


FIGURE 2: An illustration of the two-stage routing protocol: the intra-region routing and the inter-region routing.

sensor to the sink in one measurement period. Since for the routing strategy, the messages delivered to the sink are also transmitted to the nodes p_1, \dots, p_k, \dots , the energy cost also relates to the positions of these nodes, denoted by $E(p_1, \dots, p_k, \dots)$.

An ideal routing protocol should provide a long safety period Φ at small energy cost E . However, typically a longer safety period requires the messages to be transmitted in a longer way to visit p_1, \dots, p_k, \dots and thus imposes a larger energy cost. To find a balance between the safety period Φ and the energy cost E , we define the problem of designing the routing protocol as an optimization problem.

Problem 1.

$$\begin{aligned} & \underset{p_1, \dots, p_k, \dots}{\text{minimize}} && E(p_1, \dots, p_k, \dots) \\ & \text{subject to} && \Phi(p_1, \dots, p_k, \dots) \geq \Phi_o, \\ & && p_i \in \mathcal{N}, \quad i = 1, \dots, k, \dots, \end{aligned} \quad (7)$$

where Φ_o is the required safety period.

3. Routing Algorithm Description

3.1. Algorithm Model. In order to achieve k -anonymity, we propose an Euclidean minimum-spanning tree-based (EMST-based) routing algorithm to create at least k nodes whose traffic volumes are equally high. Consider a network deployed in a square Q , as depicted in Figure 2. The EMST-based routing algorithm partitions the square into k non-overlapping sub-regions W_1, \dots, W_k . Denote the partition by $W = \{W_1, \dots, W_k\}$. In each subregion W_i , a node is chosen to be the *designated node*, which locates at p_i and collects all messages originating from the sub-region W_i .

Each message is forwarded in two stages, *intra-region forwarding* and *interregion forwarding*. During intra-region

forwarding, messages originating from W_i are routed to the designated node p_i through a routing tree rooted at p_i . Once the designated node p_i receives a message generated inside W_i , it starts the inter-region forwarding by sending the message to all other designated nodes through an EMST that connects those nodes. We envision that as the message travels through the EMST, it will reach the sink that is located at most one communication range r away from the EMST. Such an arrangement can be achieved by positioning the sink after the EMST is determined. We note that we adopt an EMST because, by definition, an EMST is a spanning tree with a weight less than or equal to the weight of all other spanning trees.

Interestingly, as a result of constructing an EMST connecting k designated nodes, the number of nodes that exhibit similar traffic statistics as these k designated nodes is larger than k ; that is, $|\pi(M(t))| \geq k$. Typically, the distance between any pair of designated nodes p_i and p_j is larger than one communication range, and additional sensor nodes are needed to form a complete EMST for message relaying. As a result, additional nodes are added to $\pi(M(t))$ as a side effect of the proposed two-stage routing. To make the problem model simple yet representative, for the rest of the paper we denote k as the number of partitions, for example, the number of designated nodes, and denote K as the position vector of k designated nodes; that is,

$$K = \{p_1, \dots, p_k\}, \quad (8)$$

even though the total degree of anonymity is larger than k . The selection of the partition number k is affected by many factors. For instance, a larger k suggests constructing larger number of routing trees rooted at p_j for each region and thus larger overhead, while a smaller k may not meet the requirement of the safe period, Φ_0 . As a general rule, the value of k should be small so that it reduces the overhead of constructing multiple routing trees yet satisfies the constraint of Φ_0 . We postpone the detailed discussion on the selection of k to Section 4.

3.2. Problem Elaboration. Before updating the problem definition according to two-stage routing, we define the length of the EMST as

$$\text{EMST}(K) = \sum_{(i,j) \in \text{EMST}} \|p_i - p_j\|, \quad (9)$$

where (i, j) is the edge that connects p_i and p_j and $\|\cdot\|$ is the Euclidean distance.

According to the two-stage routing protocol, we elaborate the definition of the privacy and network efficiency metrics based on EMST(K) and hop counts

3.2.1. Safety Period Φ Quantified by EMST(K). In one reporting period, the number of messages transmitted by all nodes that are part of the EMST equals the total number of nodes in the network. Therefore, $\pi(M(t))$ contains all nodes belonging to the EMST. To further find the sink physically, the adversary has to search along the EMST. Assume that

the adversary can travel at a very high speed when he is not performing visual search such that the time he spends traveling from one location to another can be ignored. Let v denote the adversary's searching speed, and let r be the node communication range. Then as Figure 2 illustrates, the searching time is approximately

$$\Phi(K) = \frac{\text{EMST}(K) \times r}{v}. \quad (10)$$

For the rest of the paper, we will use EMST(K) as an indicator for the safety period to avoid possible confusion that might be caused by an inappropriately selected v .

3.2.2. Energy Cost E Quantified by Hop Counts. We define energy cost as the unit of hop counts. Assume the average hop size across the network is λ_h . Then, in a network consisting of uniformly distributed nodes, the average energy cost of routing a message from n_i to a designated node p_j can be approximated by the hop count [7]:

$$e_i \approx \frac{\|n_i - p_j\|}{\lambda_h}. \quad (11)$$

We note that this energy representation is sufficient to model energy spent both at the sending end and at the receiving end, since we can scale up e_i by multiplying by a coefficient α . The coefficient α can include the energy consumed both as the sender transmits the message and as its neighbors overhear and process the message.

The average total energy cost for each sensor node consists of intra-region communication E_a and inter-region communication E_e . Since every sensor node will generate one message per reporting period, the average intra-region energy cost per period per node is

$$E_a(K, W) \approx \frac{1}{\lambda_h |\mathcal{N}|} \sum_{j=1}^k \sum_{n_i \in W_j} \|n_i - p_j\|, \quad (12)$$

and the average inter-region energy cost per period per node is

$$E_e(K) \approx \frac{\text{EMST}(K)}{\lambda_h}. \quad (13)$$

Accordingly, the routing optimization problem defined as Problem 1 can be precisely formulated as follows.

Problem 2.

$$\begin{aligned} & \underset{K, W}{\text{minimize}} && E = E_a(K, W) + E_e(K) \\ & \text{subject to} && \text{EMST}(K) \geq \bar{\gamma}, \end{aligned} \quad (14)$$

where $\bar{\gamma} = v\Phi_0/r$ is the threshold value to satisfy the safety period requirement, Φ_0 .

3.3. Problem Reduction. Problem 2 defines a non-linear optimization problem that contains two variables: the locations of k designated nodes, that is, K , and the partition W .

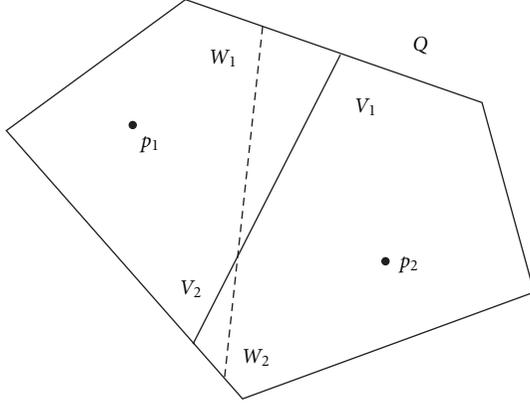


FIGURE 3: An illustration that the Voronoi partition minimizes E_a .

Solving such a nonlinear optimization problem is difficult. Thus, in this subsection we focus on reducing the problem to a simpler version.

We observe that the locations of k designated nodes will affect the inter-region communication energy cost E_c and the intra-region energy cost E_a while the partition W only affects E_a . Thus, we first examine the principle of the partition W that minimizes $E_a(K, W) + E_c(K)$. Intuitively, knowing the partitioning principle enables us to solve the problem defined in Problem 2 in two steps. (1) Finding the optimal locations of k designated nodes. (2) Applying the optimal partition W to further reduce $E_a(K, W)$.

Next, we present a result showing that, for given locations K , the Voronoi partition is the optimal partition for Problem 2.

Lemma 1. *If (K^*, W^*) is the global optimum that minimizes $E_a(K, W) + E_c(K)$, then W^* is the Voronoi partition $\mathcal{V}(K) = \{V_1, \dots, V_k\}$, where*

$$V_i = \{n_l \in \mathcal{N} \mid \|n_l - p_i\| \leq \|n_l - p_j\|, \forall j \neq i\}. \quad (15)$$

Proof. We prove the lemma by contradiction. Without loss of generality, we examine the case $k = 2$ as shown in Figure 3, and let p_1 and p_2 be the locations of the two designated nodes. The solid line located in the middle of the network region Q represents the Voronoi partition, and it perpendicularly bisects the line connecting p_1 and p_2 . Let $W = \{W_1, W_2\}$ be the optimal partition that minimizes $E_a(K^*, W) + E_c(K^*)$, shown by the dashed line. Then,

$$E_a(K^*, \mathcal{V}(K^*)) > E_a(K^*, W); \quad (16)$$

that is, for $j = \{1, 2\}$,

$$\begin{aligned} & \sum_{n_l \in V_1} \|n_l - p_j\| + \sum_{n_l \in V_2} \|n_l - p_j\| \\ & > \sum_{n_l \in W_1} \|n_l - p_j\| + \sum_{n_l \in W_2} \|n_l - p_j\|. \end{aligned} \quad (17)$$

Let $\mathcal{X}_V^{n_l}$ denote the characteristic of n_l with regard to the set V ; that is,

$$\mathcal{X}_V^{n_l} = \begin{cases} 1, & \text{if } n_l \in V, \\ 0, & \text{otherwise.} \end{cases} \quad (18)$$

Then, (17) is equivalent to

$$\begin{aligned} & \sum_{n_l \in Q} \|n_l - p_j\| \mathcal{X}_{V_1}^{n_l} + \sum_{n_l \in Q} \|n_l - p_j\| \mathcal{X}_{V_2}^{n_l} \\ & > \sum_{n_l \in Q} \|n_l - p_j\| \mathcal{X}_{W_1}^{n_l} + \sum_{n_l \in Q} \|n_l - p_j\| \mathcal{X}_{W_2}^{n_l}. \end{aligned} \quad (19)$$

For each $n_l \in Q$, it belongs to one of the following four cases. According to the definition of Voronoi partition, we have

- (1) $n_l \in V_1$ and $n_l \in W_1$: $\|n_l - p_1\| \mathcal{X}_{V_1}^{n_l} = \|n_l - p_1\| \mathcal{X}_{W_1}^{n_l}$,
- (2) $n_l \in V_1$ and $n_l \in W_2$: $\|n_l - p_1\| \mathcal{X}_{V_1}^{n_l} \leq \|n_l - p_2\| \mathcal{X}_{W_2}^{n_l}$,
- (3) $n_l \in V_2$ and $n_l \in W_1$: $\|n_l - p_2\| \mathcal{X}_{V_2}^{n_l} \leq \|n_l - p_1\| \mathcal{X}_{W_1}^{n_l}$,
- (4) $n_l \in V_2$ and $n_l \in W_2$: $\|n_l - p_2\| \mathcal{X}_{V_2}^{n_l} = \|n_l - p_2\| \mathcal{X}_{W_2}^{n_l}$.

Combining the above four cases, we have

$$\begin{aligned} & \|n_l - p_1\| \mathcal{X}_{V_1}^{n_l} + \|n_l - p_2\| \mathcal{X}_{V_2}^{n_l} \\ & \leq \|n_l - p_1\| \mathcal{X}_{W_1}^{n_l} + \|n_l - p_2\| \mathcal{X}_{W_2}^{n_l}, \end{aligned} \quad (20)$$

which contradicts to (19). Thus, the optimal partition is the Voronoi partition. \square

For the rest of the paper, we will use the following notation:

$$E_{a\mathcal{V}}(K) = E_a(K, \mathcal{V}(K)). \quad (21)$$

Additionally, to reflect the fact that E_e depends on EMST(K), we reform Problem 2 to the following.

Problem 3.

$$\begin{aligned} & \underset{K}{\text{minimize}} \quad E = E_{a\mathcal{V}}(K) + E_e(\text{EMST}(K)) \\ & \text{subject to} \quad \text{EMST}(K) \geq \bar{y}. \end{aligned} \quad (22)$$

As a result, the sets of variables for the routing optimization problem have been reduced to K , the positions of k designated nodes.

4. Quasi-Optimal Solutions

Solving Problem 3 gives us the optimal solution of k -anonymity, that is, the positions of k designated nodes that minimize the total routing energy and guarantee the safety period requirement. However, solving Problem 3 is challenging. First, Problem 3 is related to the problem of finding a set of k points in a constrained planar region such that its Euclidean minimum spanning tree has the length of a given value. To the best of knowledge, such a problem has not been addressed in the literature so far, and it is unknown

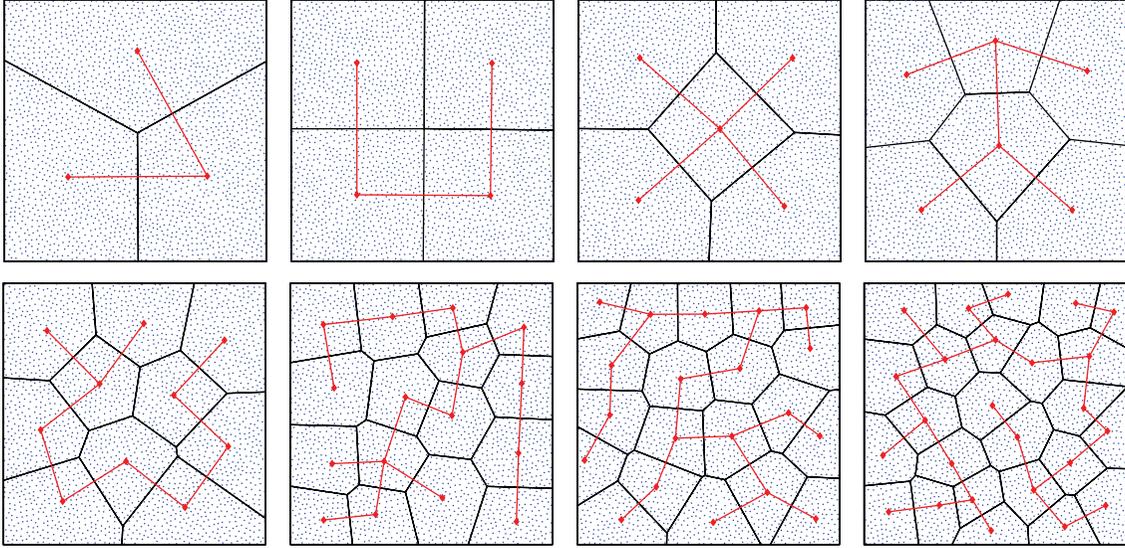


FIGURE 4: An illustration of the optimal locations K_l^* that minimize E_{av} , obtained by $GA4(k)$. The tiny dots denote sensor nodes, the black solid lines delimit the Voronoi partition, and the solid red lines denote the EMST for k designated nodes. From left to right, $k = \{3, 4, 5, 6, 10, 16, 20, 25\}$.

TABLE 1: A summary of notations.

Notations	Explanation	Problem
K_l^*	Global optimum minimizing E_{av}	4
K^*	Global optimum minimizing $E_{av} + E_e$ subject to $EMST(K) \geq \bar{\gamma}$	3
K_q^*	Quasi-optimum minimizing E_{av} subject to $EMST(K) = \bar{\gamma}$ using GAQO algorithm	5
K_a^*	Quasi-optimum minimizing E_{av} subject to $EMST(K) = \bar{\gamma}$ using APQO algorithm	5

whether the problem is NP hard. Second, our Problem 3 seeks optimized locations for an energy cost function subject to an EMST constraint and thus creates more difficulties.

Popular methods for solving nonlinear optimization problems, such as the generalized reduced gradient [19], are inapplicable to solve Problem 3, because those methods leverage the first or second derivative of the objective function to search for the optimal solution and the derivative of $EMST(K)$ is complicated to formulate. Searching for the optimal positions of designated nodes through every conceivable value is computationally infeasible. To tackle the problem, we first analyze Problem 3 by finding a K that minimizes $E_{av}(K)$ using genetic algorithms (GA) and then propose quasi-optimal algorithms to obtain a solution approximating the optimal one.

To facilitate discussion, we summarize the notation convention of optimal solutions to Problem 3 and its reduced subproblems in Table 1.

4.1. Minimizing $E_{av}(K)$. The objective function consists of two components: $E_{av}(K)$ and $E_e(K)$, and we start by

searching for a K that minimizes the first component $E_{av}(K)$, namely, solving the following problem:

Problem 4.

$$\underset{K}{\text{minimize}} E_{av}(K) \quad (23)$$

Problem 4 is still a nonlinear optimization problem with an objective function whose derivative is difficult to calculate. We choose to exploit the widely adopted genetic algorithms (GAs) to find the optimal solution. GA mimics Darwin's theory about evolution. It iteratively generates a set of solutions known as a population and selects a subset of solutions to form a new population based on each solution's "fitness." The fitness level of a solution can be evaluated using the objective function of the optimization problem. "Fitter" solutions will be selected with higher probability while "weaker" solutions will still have chances to be selected. As a result, GA is likely to escape from local optima and evolves to the global optima with high probability. Thus, we call the solutions obtained by GA as optimal solutions.

We call our customized genetic algorithm that searches for optimal solutions of Problem 4 as $GA4(k)$, and we built our $GA4(k)$ using Matlab toolbox GAtool and searched for optimal designated node locations in a 2500-node network that is deployed in a $1000\text{m} \times 1000\text{m}$ square with a uniform density. The node communication range r was set to 40 m, which resulted in an average hop size λ_h of $(2/3) \times 40\text{m}$. We constructed the "chromosome" as K , that is, k coordinates of designated nodes and performed multiple runs of experiments while changing the value of k . For each k , we ran the experiments about 10 times, and we set the population size to approximately $k \times 100$, the crossover fraction to 0.8, and the maximum number of generations to 100. Figure 4 shows the typical patterns for optimal

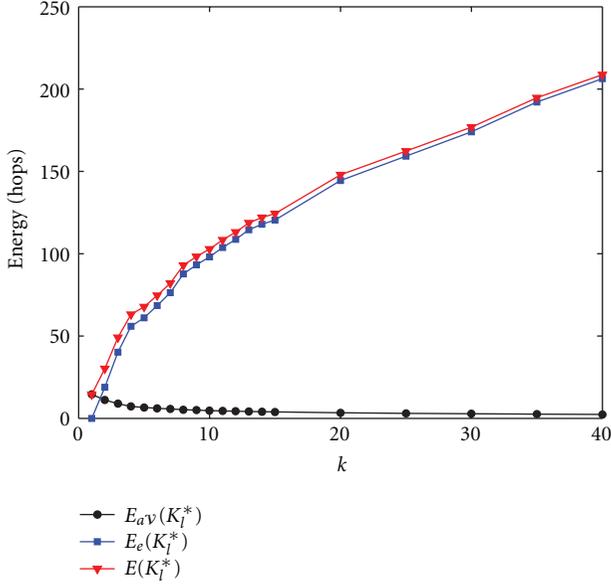


FIGURE 5: The routing efficiency measure: the intra-region energy $E_{aV}(K_i^*)$, inter-region energy $E_e(K_i^*)$, and total energy $E(K_i^*)$ with regard to k . K_i^* is the optimal locations that minimize $E_{aV}(K)$ obtained via GA4(k), and $E(K_i^*) = E_{aV}(K_i^*) + E_e(K_i^*)$. This plot shows that $E_e(K_i^*)$ dominates $E(K_i^*)$.

designated nodes' positions K that minimize $E_{aV}(K)$ and the corresponding EMST(K), when $k = \{3, 4, 5, 6, 10, 16, 20, 25\}$.

Remark 2. From Figure 4, we observe that for each optimal layout the designated nodes are distributed almost uniformly across the network, and the network area Q is partitioned into regions with similar sizes. This observation can be intuitively explained by rewriting (12) as

$$E_{aV}(K) = \frac{\bar{d}}{\lambda_h}, \quad (24)$$

where \bar{d} is the average distance between every sensor node and its nearest designated node. To minimize $E_{aV}(K)$ the designated nodes have to be deployed in such a way that \bar{d} is minimized.

Remark 3. We depict $E_{aV}(K_i^*)$, $E_e(K_i^*)$, and $E(K_i^*)$ in Figure 5 and EMST(K_i^*) in Figure 6, which show that both $E_e(K_i^*)$ and EMST(K_i^*) increase with k while $E_{aV}(K_i^*)$ decreases with k . Intuitively, when the number of partitioned regions increases, the average distance between a sensor node and its nearest designated node \bar{d} decreases and so does $E_{aV}(K_i^*)$. However, the increase of k causes the designated nodes to further spread out and thus increases EMST(K_i^*). A slight change of EMST(K_i^*) will cause a larger level of ΔE_e than $\Delta E_{aV}(K)$, because $\Delta \text{EMST}(K)$ creates an equivalent level of ΔE_e while amortized among all nodes with regard to $E_{aV}(K)$. Thus, we observe that as k increases, E_e grows quickly, and soon $E_e(K_i^*) \gg E_{aV}(K_i^*)$.

To estimate the relationship between EMST(K_i^*) and k , we performed a regression analysis on the empirical results

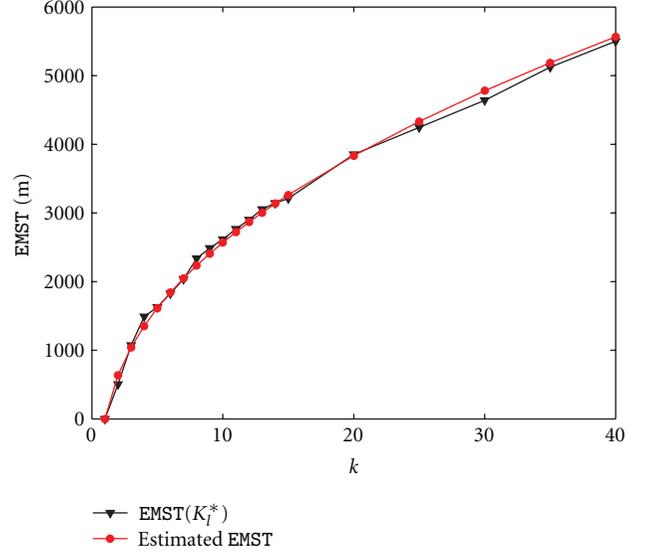


FIGURE 6: The privacy measure: a comparison of estimated EMST and EMST(K_i^*) with respect to k . The estimated EMST is calculated according to a regression formula (27), which turns out to be a close fit to the empirical one obtained using GA4(k).

of EMST(K_i^*) and k . Rather than choosing a polynomial, we construct the regression function according to Remark 2; that is, the network area Q is very likely to be partitioned into regions of similar sizes and the distances between every two neighboring designated nodes (two designated nodes that are connected by an edge in the EMST) are roughly the same. Let \bar{r}_w be the average distance between neighboring designated nodes. Then

$$\text{EMST}(K_i^*) = (k-1)\bar{r}_w. \quad (25)$$

Additionally, we can use a disk with radius $\bar{r}_w/2$ to approximate the area of each region, and

$$k\pi\left(\frac{\bar{r}_w}{2}\right)^2 = A_Q \times \beta, \quad (26)$$

where A_Q is the area of the square Q and $0 < \beta < 1$ is a coefficient describing how close the disk approximates each region on average. Thus, the length of EMST(K_i^*) can be estimated by the following equation:

$$\text{EMST}(K_i^*) = 2(k-1)\sqrt{\frac{\beta A_Q}{k\pi}}. \quad (27)$$

Our regression analysis showed that the fitting error is minimized when $\beta = 0.64$. As shown in Figure 6, the comparison between the estimated EMST(K_i^*) with $\beta = 0.64$ and the empirical one obtained by GA show that the regression line is a close fit.

4.2. GA-Based Quasi-Optimal Algorithm. Analyzing Problem 4 utilizing GA provides important insights towards solving the original routing optimization problem defined in Problem 3. In this subsection, we introduce a GA-based quasi-optimal algorithm (GAQO) that can obtain an approximate

optimal solution for Problem 3. In particular, the GAQO algorithm provides the quasi-optimal solution K_q^* to the following problem:

Problem 5.

$$\begin{aligned} & \underset{K}{\text{minimize}} && E_{av}(K) \\ & \text{subject to} && \text{EMST}(K) = \bar{\gamma}. \end{aligned} \quad (28)$$

We will show that the quasi-optimal solutions for Problem 5 closely approximate the solutions for Problem 3 empirically. Intuitively, according to Remark 3, a slight change of $\text{EMST}(K)$ will cause a larger level of increase of E_e than decrease of $E_{av}(K)$. Thus, our approach is to minimize $E_e(K)$ as much as possible. Note that $E_e(K)$ achieves its minimum when $\text{EMST}(K) = \bar{\gamma}$. Thus, ensuring that $\text{EMST}(K_q^*) = \bar{\gamma}$ will produce a solution approximating the optimal solution for Problem 3.

4.2.1. Approximation Evaluation Metric. To evaluate how close the solutions obtained by the GAQO algorithm approximates the optima, we define the approximation evaluation metric μ as the energy difference between $E(K_q^*)$ and $E(K^*)$:

$$\mu = E(K_q^*) - E(K^*). \quad (29)$$

We will show that μ is bounded by the difference between the intra-region energy E_{av} of K_q^* and K_l^* :

$$\mu \leq E_{av}(K_q^*) - E_{av}(K_l^*). \quad (30)$$

We now justify (30) by proving the following lemma.

Lemma 4. $E_{av}(K_l^*) + E_e(K_q^*) \leq E_{av}(K^*) + E_e(K^*) \leq E_{av}(K_q^*) + E_e(K_q^*)$.

Proof. (Second inequality.) By definition, for a given k , K^* is the global optimum which minimizes $E_{av}(K) + E_e(K)$, so $E_{av}(K^*) + E_e(K^*) \leq E_{av}(K_q^*) + E_e(K_q^*)$.

(First inequality.) For a given k , K_l^* minimizes $E_{av}(K)$. Thus, $E_{av}(K_l^*) \leq E_{av}(K^*)$. Additionally, by definition, $\text{EMST}(K_q^*) = \bar{\gamma}$ and $\text{EMST}(K^*) \geq \bar{\gamma}$. Thus,

$$E_e(K_q^*) \leq E_e(K^*). \quad (31)$$

Combining both facts, we conclude that $E_{av}(K_l^*) + E_e(K_q^*) \leq E_{av}(K^*) + E_e(K^*)$. Therefore, the lemma is proved. \square

4.2.2. Algorithm Walk-Through. Searching optimum K_l^* for Problem 4 using GA has provided insights of K^* (We did not apply GA to solve Problem 5, because the constraint of $\text{EMST}(K) = \bar{\gamma}$ makes it prohibitively time consuming to obtain a feasible solution.). In particular, for a given k , if the required $\bar{\gamma}$ happens to equal $\text{EMST}(K_l^*)$, then K_l^* is the global optimum for Problem 3 that is, $K^* = K_l^*$. We take the hypothesis that optimal solutions for different threshold values $\bar{\gamma}$ are continuous and design our GA-based quasi-optimal (GAQO) algorithm with steps shown in Algorithm 1:

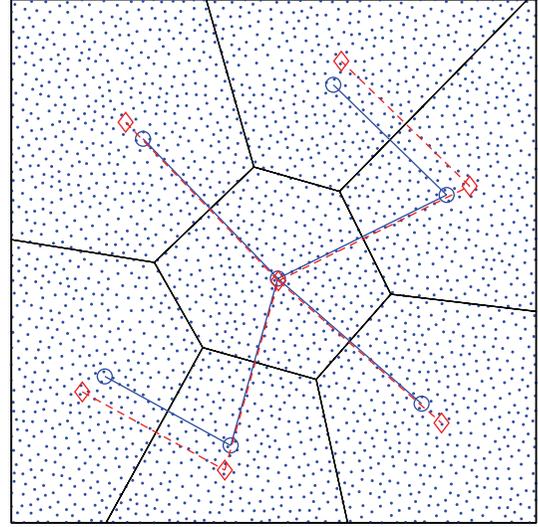


FIGURE 7: The red “ \diamond ” points are the optimal designated node locations that minimize $E_{av}(K_l^*)$ for $k = 7$, derived via $\text{GA4}(k)$, and the blue “ \circ ” points are the quasi-optimal result derived with our GAQO algorithm.

Require:INPUT:

$\bar{\gamma}$;

OUTPUT:

K_q^* ;

(1) **PROCEDURES:**

$k = \text{Closest_EMST}(\bar{\gamma})$

(2) $K_l^* = \text{GA4}(k)$

(3) $\alpha = \bar{\gamma}/\text{EMST}(K_l^*)$

(4) $K_q^* = \alpha K_l^*$

ALGORITHM 1: GA-based quasi-optimal algorithm for the k -anonymity sink-location privacy problem.

Step 1. Call `Closest_EMST` to find k whose $\text{EMST}(K_l^*)$ is closest to the given $\bar{\gamma}$, according to (27).

Step 2. For the given k , find an optimal layout K_l^* for Problem 4 using genetic algorithm $\text{GA4}(k)$.

Step 3. Shrink or expand K_l^* with regard to the center of the network area Q until $\text{EMST}(K_q^*) = \bar{\gamma}$. Let the center of Q be the origin of the coordinate, and let $\alpha = \bar{\gamma}/\text{EMST}(K_l^*)$. Then $K_q^* = \alpha K_l^*$.

We note that the aforementioned GAQO algorithm, though not optimal, does approximate optimal solutions.

Example 5. Here, we illustrate how the GAQO algorithm achieves k -anonymity for a given safety period $\bar{\gamma}$ in Figure 7. We use the same parameters of the sensor network described in Section 4.1 and set the required safety period $\bar{\gamma} = 2000$ m. In the first step, based on (27), GAQO concluded that the closest $\text{EMST}(K_l^*) = 2035.76$ m when $k = 7$. Then, GAQO utilized the genetic algorithm $\text{GA4}(k)$ to search for

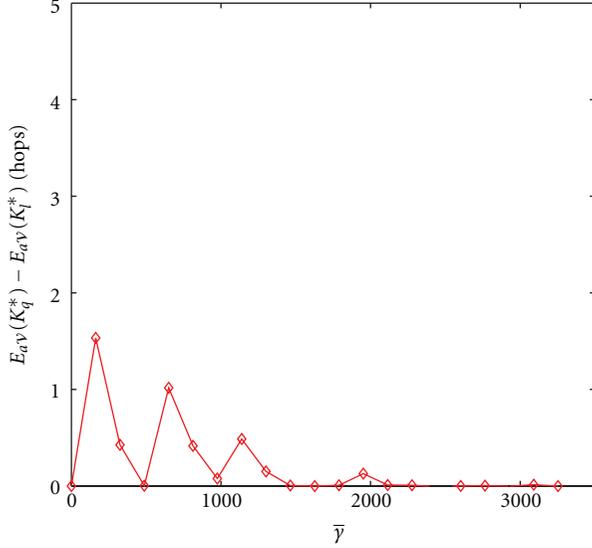


FIGURE 8: The algorithm approximation measure: the upper bound of the difference between the quasi-optimum (using the GAQO algorithm) and the global optimum (using GA4(k)), as $\bar{\gamma}$ varies.

the optimal positioning of 7 designated nodes. An example layout of K_l^* when $k = 7$ is denoted by the red “ \diamond ” points in Figure 7. Since $\text{EMST}(K_l^*) > \bar{\gamma}$, GAQO shrank K_l^* to the quasi-optimal layout of the designated nodes K_q^* , as marked by blue “ \circ ” points in Figure 7.

4.2.3. Evaluation. To evaluate how close the solutions obtained by the GAQO algorithm approximate the optimal solution, we performed an empirical study. In particular, we used the same network setup as before and searched for the quasi-optimal solutions in a 2500-node network deployed in the 1000×1000 m square. We changed the constraint of Problem 5 by varying the length of $\text{EMST}(K)$. To capture the statistical character of GAQO, for each $\text{EMST}(K)$ value, we ran the algorithm at least 10 times over randomly generated network topologies, and calculated the upper bound of the difference between the quasi-optimal solution and global optimal solution, that is, $E_{av}(K_q^*) - E_{av}(K_l^*)$. The plot in Figure 8 has confirmed that for the quasi-optimal solution obtained by the GAQO algorithm, K_q^* approaches K^* as $\bar{\gamma}$ increases.

4.3. Artificial Potential-Based Quasi-Optimal Algorithm. The GAQO algorithm can obtain quasi-optimal solutions of the k -anonymity sink-location problem. However, our simulation study shows that the run time of GA4(k), that is, the algorithm that searches for K_l^* that minimizes $E_{av}(K)$ using genetic algorithms, increases quadratically as the constraint $\bar{\gamma}$ increases. To efficiently solve the k -anonymity sink-location problem, we design an artificial potential-based algorithm named AP4(k) to substitute GA4(k), and we call the new quasi-optimal algorithm leveraging AP4(k) an APQO algorithm.

Artificial potential (AP) [20] (aka. artificial physics in some literature as opposed to natural physics) was originally developed for the purpose of obstacle avoidance. Later, it

was used as a distributed control strategy to solve self-deployment problems of WSNs. The approach is simple enough to let each entity exert forces on other nearby entities and respond to forces from them; yet a uniform distribution will eventually emerge. Since the approach is largely independent of the number of entities, it scales well for large sets of entities. We take advantage of the linear time complexity of an AP-based method to solve the k -anonymity sink-location problem, since searching for optimal solutions of k designated nodes is equivalent to deploying nodes uniformly across the network (according to Remark 2).

We built our APQO algorithm on the AP-based self-deployment algorithm proposed by Ding et al. [21], whereby sensors are deployed into uniform lattices inside a bounded region. We start by assuming the k designated nodes can move to any position inside the network area and we denote $\mathbf{z} = [z_1^T, z_2^T, \dots, z_k^T]^T$ the aggregate position vector of k mobile nodes. Once the AP-based algorithm converges and finds the final position \mathbf{z}^* , we select those sensor nodes that are closest to \mathbf{z}^* to be the designated nodes.

4.3.1. AP Definition. Two types of artificial potential functions are defined for every node i : V_{ij}^1 , which is the potential between node i and another node j ($j \neq i$), and V_{is}^2 , which is the potential between node i and the boundary. The artificial potential has the following characteristics. When node i is located close to another node j or to the boundary, the potential is high and has a tendency to push node i away. When node i is very far away from another node or the boundary, the potential reduces to zero. V_{ij}^1 is defined as

$$V_{ij}^1 = \begin{cases} (l_{ij} - r_e)^2 + \frac{1}{l_{ij}^2}, & 0 < l_{ij} \leq r_e, \\ 0, & \text{else,} \end{cases} \quad (32)$$

where $l_{ij} = \|z_i - z_j\|$ is the distance between these two mobile nodes and r_e is the effective radius of the potential.

We define V_{is}^2 as the potential between mobile node i and the nearest point on the boundary $q_s \in N_i$, where N_i is the set of all the nearest points, and $N_i = \{q \mid \arg \min_{q \in B} \|z_i - q\|\}$, B being the set of all points on the boundary. We note that N_i may not be a singleton. For example, when the z_i is on the diagonal of the square, there exist two nearest points with each on one edge of the square. V_{is}^2 is defined as

$$V_{is}^2 = \begin{cases} (l_{is} - r'_e)^2 + \frac{1}{l_{is}^2}, & 0 < l_{is} \leq r'_e, \\ 0, & \text{else,} \end{cases} \quad (33)$$

where $l_{is} = \|z_i - q_s\|$ and r'_e is the effective radius of the boundary potential. Here we set $r'_e = r_e/2$.

The relationships between V_{ij}^1 and the distance of l_{ij} and between V_{is}^2 and l_{is} are depicted in Figure 9, which exhibit desired characteristics.

In addition, we define the total potential as

$$V(\mathbf{z}) = \frac{1}{2} \sum_{i=1}^k \sum_{j=1}^k V_{ij}^1 + \sum_{i=1}^k \sum_{q_s \in N_i} V_{is}^2. \quad (34)$$

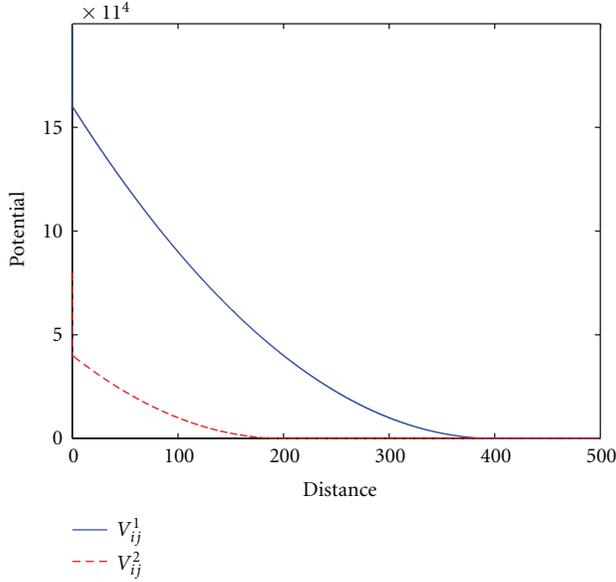


FIGURE 9: V_{ij}^1 and V_{is}^2 with regard to l_{ij} and l_{is} when $r_e = 400$.

To distribute k nodes approximately uniformly inside the network area is equivalent to finding \mathbf{z} that minimize V :

$$\mathbf{z}^* = \underset{\mathbf{z}}{\operatorname{arg\,min}} V(\mathbf{z}). \quad (35)$$

We consider the gradient descent method to find the minimum for $V(\mathbf{z})$ and define the following position update scheme for mobile node i :

$$\dot{z}_i = -\frac{\partial V}{\partial z_i} = -\left(\sum_{j=1}^k \frac{\partial V_{ij}^1}{\partial z_i} + \sum_{q_s \in N_i} \frac{\partial V_{is}^2}{\partial z_i} \right), \quad (36)$$

that is, we let the mobile nodes move towards the negative of the gradient to minimize the total potential V .

4.3.2. Algorithm Walk-Through. Overall, the APQO follows the similar framework as shown in Algorithm 1. For a given \bar{y} , the function `Closest_EMST()` returns k whose corresponding EMST(K_l^*) is closest to \bar{y} , according to the line fitting equation (27). Different from GAQO, APQO utilizes the AP-based function AP4(k) to find the quasi-optimal layout K_a that minimizes $E_{av}(K)$. Similar to GAQO, APQO also shrinks or expands K_a with regard to the center of Q until $\operatorname{EMST}(K_a^*) = \bar{y}$; that is, $K_a^* = (\bar{y}/\operatorname{EMST}(K_a))K_a$.

We listed the pseudocode of AP4(k) in Algorithm 2, which contains the following steps.

Step 1. Initialize the locations of the k nodes \mathbf{z} to be around the center of the network square Q without overlapping.

Step 2. Obtain the gradients $\dot{\mathbf{z}}$, and update the location vector \mathbf{z} according to the gradients $\dot{\mathbf{z}}$ and the step size Δ (a small constant we choose) iteratively until convergence. Denote the converged position as \mathbf{z}^* .

Require: INPUT:

k ;

OUTPUT:

K_a ;

(1) **PROCEDURES:**

$\mathbf{z}(0) = \operatorname{Initialize_z}(k)$;

(2) **repeat**

(3) $\mathbf{z}(n\Delta) = \mathbf{z}((n-1)\Delta) + \Delta \cdot \dot{\mathbf{z}}((n-1)\Delta)$;

(4) Error = $\|\mathbf{z}(n\Delta) - \mathbf{z}((n-1)\Delta)\|$;

(5) **Until** Error < Error_Threshold

(6) $K_a = \operatorname{Closest_nodes}(\mathbf{z}(n\Delta))$

ALGORITHM 2: AP4(k): AP-based method for solving Problem 4.

Step 3. Select the sensor nodes that are closest to \mathbf{z}^* to be the designated nodes, and we call their positions as K_a .

We use the following lemma to show that the AP4(k) algorithm must converge.

Lemma 6. *The AP-based algorithm is convergent; that is, $z_i(t)$ asymptotically approaches the location where $\dot{z}_i = -\partial V/\partial z_i = 0$.*

Proof. Taking the derivative of V , we obtain

$$\begin{aligned} \dot{V} &= \left[\frac{\partial V}{\partial z_1}, \frac{\partial V}{\partial z_2}, \dots, \frac{\partial V}{\partial z_k} \right] \dot{\mathbf{z}} \\ &= -\dot{\mathbf{z}}^T \dot{\mathbf{z}} = -\|\dot{\mathbf{z}}\|^2 \leq 0. \end{aligned} \quad (37)$$

Therefore, $V(\mathbf{z}(t)) \leq V(\mathbf{z}(0)) < \infty$ and $V(\mathbf{z}(t))$ is bounded for $t \geq 0$. Further, note from (33) that V tends to ∞ if l_{is} approaches 0. Thus, the boundedness of $V(\mathbf{z}(t))$ implies that l_{is} will never become 0 and $z_i(t)$ remains inside the network region Q all the time.

Let $\Omega = \{\mathbf{z} \in Q^k \mid V(\mathbf{z}(t)) \leq V(\mathbf{z}(0))\}$. Then by LaSalle's invariance principle [22], the trajectory $\mathbf{z}(t)$ converges to the largest invariant set in $\mathcal{M} = \{\mathbf{z} \in \Omega \mid \dot{V} = -\|\dot{\mathbf{z}}\|^2 = 0\}$, which completes the proof. \square

4.3.3. Evaluation. Similar to the GAQO algorithm, we have defined an approximation evaluation metric

$$\mu = E(K_a^*) - E(K^*), \quad (38)$$

and μ is bounded by the difference between the intra-region energy E_{av} of K_a^* and K_l^* :

$$\mu \leq E_{av}(K_a^*) - E_{av}(K_l^*). \quad (39)$$

To evaluate the APQO algorithm, we performed an empirical study using the same network setup as before: a 2500-node network deployed in the 1000 \times 1000 m square. Figure 10 shows the result, and for the quasi-optimal solution obtained by the APQO algorithm, K_a^* approaches K^* as \bar{y} increases. Additionally, the steady-state locations of the k designated nodes K_a , obtained by AP4(k), are affected by the value of r_e .

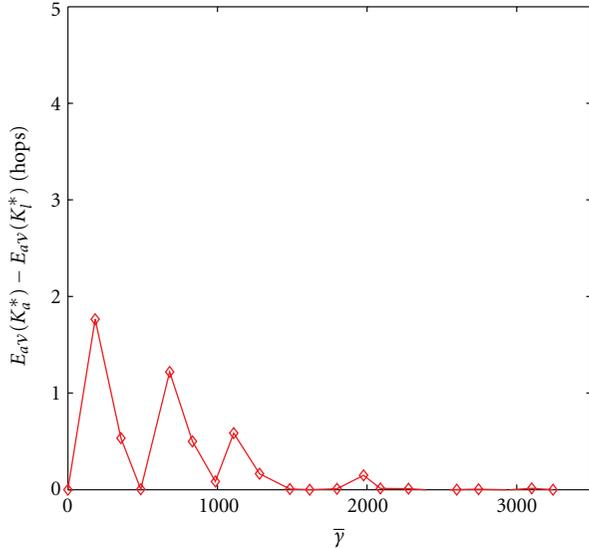


FIGURE 10: The algorithm approximation measure: the upper bound of the difference between the quasi optimum (using the APQO algorithm) and the global optimum (using GA4(k)), as \bar{y} varies.

If r_e is small and k disks (with a radius of $r_e/2$) are not enough to fill the region Q , then in the steady state, each designated node is at least r_e away from its nearest designated node [23]. In comparison, if r_e is large and k disks are more than enough to fill the region, the distances from any pairs of nearest designated nodes in the steady state are less than r_e . For a given k , to ensure that the length of the EMST(K_a) obtained by AP4(k) is similar to the one obtained by GA4(k), we set r_e to \bar{r}_w , the average distance between neighboring designated nodes obtained by empirical equation (27). Additionally, we adopted the same setups as the one for the GAQO algorithm evaluation and used the same topologies to evaluate the APQO algorithm.

Performance Comparison. The length of EMSTs obtained using GA4(k) and AP4(k) is presented in Figure 11(a), and the locations K_a derived by AP4(k) for various k are demonstrated in Figure 12. We note that the resulting EMSTs shown in Figure 12 appear slightly different from the ones that are obtained via GA4(k) (shown in Figure 4). This is because k designated nodes are scattered roughly evenly across the network and a slight variation of their locations will cause the EMST to go through edges connecting different pairs of nodes. However, the numerical results of EMST length show that the AP-based AP4(k) algorithm can acquire EMSTs of similar length as the ones derived by GA4(k). Further, as shown in Figure 11(b), for a given \bar{y} , the total energy levels obtained by the APQO algorithm fit closely with what the GAQO algorithm derives, which indicates that the APQO algorithm can also obtain quasi-optimal solutions for Problem 3.

Time Complexity Comparison. Since the majority of the run-time for the GAQO and APQO algorithms is contributed

by executing GA4(k) and AP4(k), we measure the run-time of GA4(k) and AP4(k) only. We tested both GA4(k) and AP4(k) on a computer equipped with a 2.1 GHz AMD dual-core CPU and 3 GB RAM and depicted the run-time of these two algorithms when varying k in Figure 11(c). Figure 11(c) shows that the run-time of GA4(k) increases quickly as k increases while the run-time of AP4(k) remains short. This is because the time complexity for GA4(k) is $O(nk^2)$, where n is the total number of nodes in the network, and the time complexity of AP4(k) is $O(k)$.

GA4(k) involves calculating multiple generations, and each generation has a population size of $k \times 100$. Computing the fitness function $E_{av}(K)$ for each individual requires calculating the distance between k designated nodes and all n network nodes. Considering that the maximum number of generations is at most 1000 in our simulation, the time complexity of GA4(k) is $O(nk^2)$. In comparison, each iteration of AP4(k) only involves updating k locations z_i . Since the total number of iteration is independent of the number k , the time complexity of AP is $O(k)$. In our simulation, AP4(k) converged around 1s to 5s. Thus, APQO performs better than GAQO as the number of nodes in the network increases.

k-Anonymity Evaluation. We evaluated how effective the EMST-based routing protocol can change the traffic pattern around the sink. Let the node that is closest to the sink be n_{cs} . We are interested in the number of nodes exhibiting the same traffic statistics as n_{cs} . Denote N_{ρ_v} as the number of nodes whose traffic volumes $\rho_v^{n_i}$ (3) are the same as that of n_{cs} , and denote N_{ρ_e} as the number of nodes which has the same number of messages ended there $\rho_e^{n_i}$ (4) as n_{cs} . Figure 13 shows the trend of N_{ρ_v} and N_{ρ_e} when \bar{y} and k increase. It indicates that the EMST-based two-stage routing algorithm can effectively hide the location of the sink. Almost all nodes in the network appear to have the same $\rho_e^{n_i}$ as that of n_{cs} , and a lot more network nodes other than k designated nodes forward the same amount of traffic as n_{cs} .

5. Related Work

Protecting the identity of traffic sources has been extensively studied in the context of general networks, where the usage of a series of intermediate mixes and onion routing [24] was proposed to cope with traffic analysis. The problems of tracking users' paths in wireless networks with location-oriented services were studied by Gruteser and Grunwald [25] and Hoh and Gruteser [26], and they proposed a path perturbation algorithm to increase source location anonymity. Since sensor networks have constrained resources, those methods are not applicable there.

In the context of wireless sensor networks, both source-location privacy and sink-location privacy have attracted attention from the research community. Source location privacy focuses on protecting the message source, since such information can reveal sensitive position information of the target that is close to the message source. Preserving source-location privacy against a local adversary was first studied by Kamat et al. [2], where fake message injection and phantom

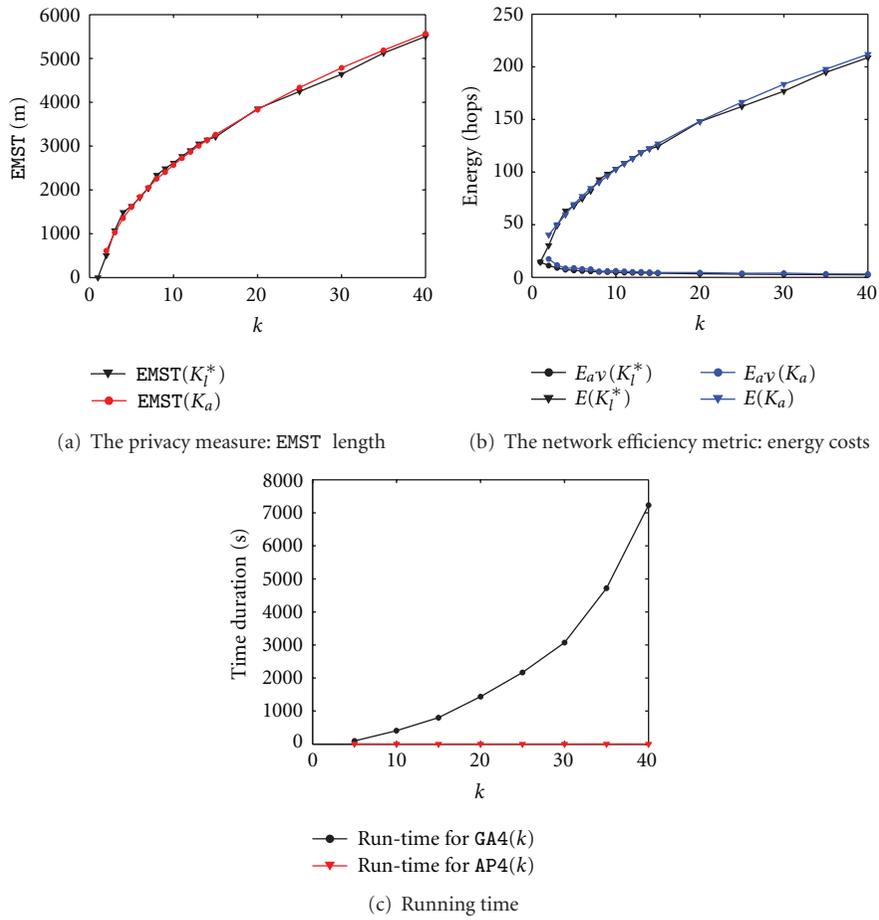


FIGURE 11: Comparison of GA4(k) and AP4(k). For both methods, we used the same network setup and searched for the quasi-optimal solutions in a 2500-node network deployed in the 1000 × 1000 m square.

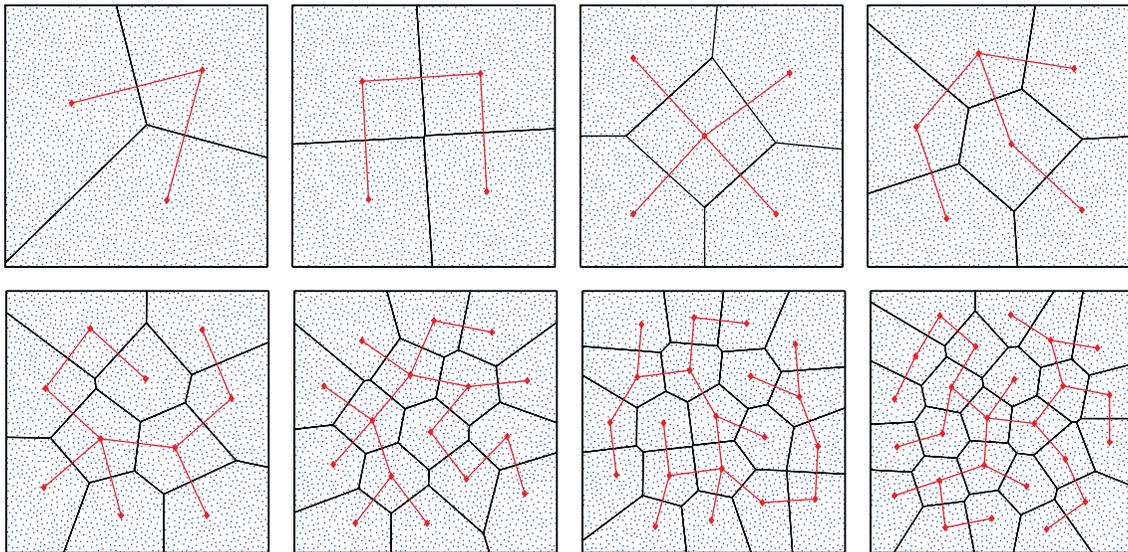


FIGURE 12: The quasi-optimal locations of k designated nodes which approximately minimize E_{av} , derived via AP4(k). From left to right and top to down $k = \{3, 4, 5, 6, 10, 16, 20, 25\}$.

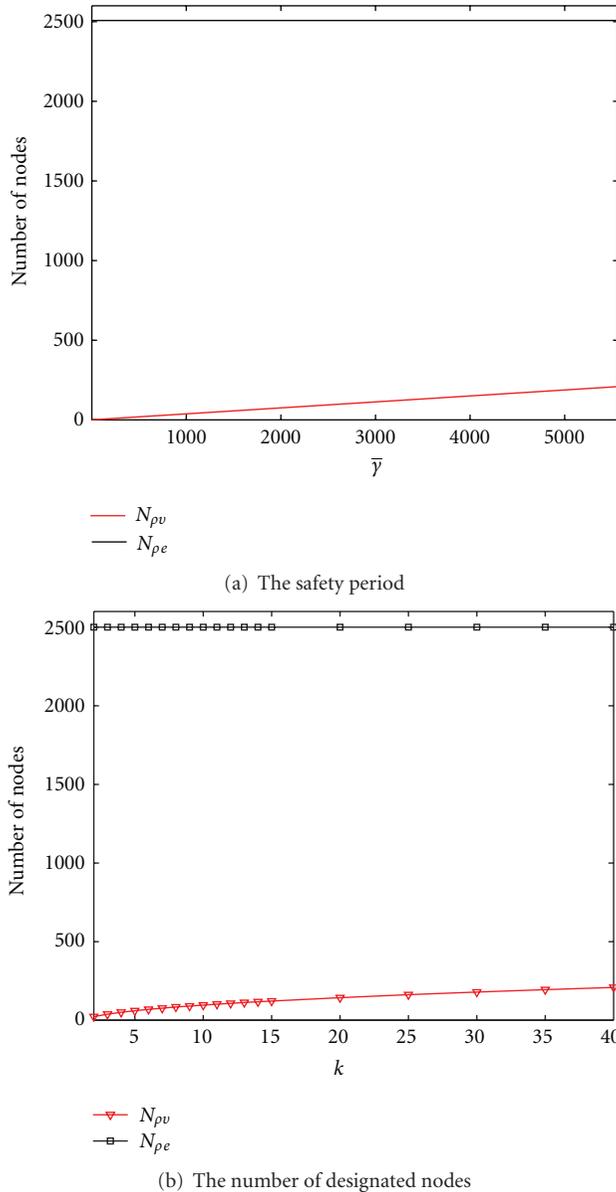


FIGURE 13: The number of nodes that exhibit the same traffic statistics as the nodes around the sink. N_{pv} is the number of nodes which has the same traffic volume, and N_{pe} is the number of nodes which has the same number of messages ended there.

routing are proposed to prevent a local eavesdropper from discovering the message source through hop-by-hop traces.

The problem of preserving source-location privacy under a global eavesdropper has been studied extensively [1, 4, 27, 28]. Mehta et al. [4] have proposed periodic collection and source simulation techniques to prevent the leakage of message source location, and Yang et al. [1] have introduced dummy traffic to hide the real message source. Ouyang et al. [27] have devised a set of privacy-preserving algorithms involving sending periodic maintainable messages to address a laptop-class attacker who has longer radio range and can eavesdrop on all communications in a sensor network. A notion of statistically strong source anonymity is proposed

by Shao et al. [28], and a strategy called FitProbRate has been proposed to achieve statistically strong source anonymity with a reduced real event report latency.

In the areas of enhancing sink-location privacy, Deng et al. [9] have shown that traffic analysis can reveal the location of sinks and proposed several antitraffic analysis countermeasures to hide the direction of data flow and create fake sink locations that exhibit artificially high traffic. In their follow-up work [5], multiple parent routing, controlled random walk, random fake paths, and combinations of all three routing algorithms have been studied to generate randomness against traffic rate monitoring and traffic path direction attacks. Location privacy routing (LPR) [3] utilizes probabilistic routing and fake message injection to deceive an adversary from tracking the direction of traffic flow. Conner et al. [29] proposed the decoy sink protocol, whereby data are forwarded to a decoy sink for aggregation before they are relayed to the real sink. As a result, the traffic volume near the sink is reduced while decoy sinks exhibit high traffic volume, which makes traffic analysis attacks difficult. Liu and Xu [7] presented a zeroing-in attack that can be launched by resource constraint adversaries and proposed a random walk-based defense strategy. Gu et al. [6] proposed a privacy-preserving scheme which obfuscates the sink's location with dummy sink nodes and can help secure existing mobility control protocols against attacks. However, those strategies cannot cope with a global adversary.

To deal with global adversaries, Ngai [8] proposed randomized routing with hidden address (RRHA), whereby packets are routed from the source to the sink along a random path and the destination field is not included in the header of the packets. Such a routing protocol does provide sink anonymity, but the packet may not reach the sink at all. Additionally, Nezhad et al. [10] designed an anonymous routing protocol to preserve the sink-location privacy against a global adversary. However, their global adversaries are only capable of packet-tracing attacks not traffic-analysis attacks. In this paper, we focused on addressing the problem of enhancing sink-location privacy against a global adversary capable of both attacks, while assuring that messages will arrive at the sink.

Artificial potential was originally developed in Khatib [20] for the purpose of obstacle avoidance. Later, it was used as a distributed control strategy for a large number of entities to achieve certain geometric configurations, such as in coverage and connectivity problems of WSNs [21, 30] and formation and flocking problems of collective artificial agents [31]. Since the approach is largely independent of the size and number of entities, the results scale well to larger sets of entities. We take advantage of the linear time complexity of this method to solve a nonlinear optimization problem that defines the k -anonymity sink-location problem.

6. Concluding Remarks

Wireless sensor networks rely on the sink to collect the measurements across the entire network; thus it is essential to protect the location information of the sink. However, the traffic around the sink typically exhibits distinctive patterns,

and an adversary with a global view can identify the location of the sink by measuring the traffic statistics of the entire network. In this study, we addressed such a threat, and we proposed an EMST-based two-phase routing algorithm that can achieve k -anonymity of the sink. In particular, the network is partitioned into k regions with each containing one designated node. Messages are first delivered to one designated node and then forwarded onto the EMST that interconnects all other designated nodes. The two-phase routing algorithms can effectively create many entities that exhibit the same traffic pattern as the nodes located close to the sink.

The positioning of k designated nodes affects two conflicting goals: the routing energy cost and the privacy level of the sink's location, and thus we formulated it as a nonlinear optimization problem. To tackle this problem, we first utilized a genetic algorithm to search for quasi-optimal solutions and developed a genetic algorithm-based quasi-optimal (GAQO) algorithm that can obtain solutions which closely approximate global optimal solutions. Further motivated by the observation that the quasi-optimal solution partitions the network into areas with similar sizes, we designed an artificial potential-based quasi-optimal (APQO) algorithm that can also obtain a quasi-optimal positioning of k nodes but which requires significantly reduced run-time. Our simulation results validated that both algorithms can effectively derive the positions of k designated nodes which meet the requirement of privacy at the minimum routing energy cost.

Acknowledgments

The authors thank Dr. Jianjun Hu for his feedback on the genetic algorithms. This work is partially supported by the National Science Foundation Grant CNS-0845671.

References

- [1] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao, "Towards event source unobservability with minimum network traffic in sensor networks," in *Proceedings of the 1st ACM Conference on Wireless Network Security (WiSec '08)*, pp. 77–88, ACM, 2008.
- [2] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source location privacy in sensor network routing," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS '05)*, pp. 599–608, IEEE Computer Society, 2005.
- [3] Y. Jian, S. Chen, Z. Zhang, and L. Zhang, "Protecting receiver-location privacy in wireless sensor networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM'07)*, pp. 1955–1963, 2007.
- [4] K. Mehta, D. Liu, and M. Wright, "Icnp'07: location privacy in sensor networks against a global eavesdropper," in *Proceedings of the IEEE International Conference on Network Protocols*, pp. 314–323, 2007.
- [5] J. Deng, R. Han, and S. Mishra, "Countermeasures against traffic analysis attacks in wireless sensor networks," in *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM '05)*, pp. 113–126, IEEE Computer Society, 2005.
- [6] Q. Gu, X. Chen, Z. Jiang, and J. Wu, "Sink-anonymity mobility control in wireless sensor network," in *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pp. 36–41, 2009.
- [7] Z. Liu and W. Xu, "Zeroing-in on network metric minima for sink location determination," in *Proceedings of the 3rd ACM conference on Wireless network security (WiSec '10)*, pp. 99–104, ACM, 2010.
- [8] E. C.-H. Ngai, "On providing sink anonymity for sensor networks," in *Proceedings of the International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, pp. 269–273, ACM, 2009.
- [9] J. Deng, R. Han, and S. Mishra, "Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks," in *Proceedings of the International Conference on Dependable Systems and Networks (DSN '04)*, p. 637, IEEE Computer Society, 2004.
- [10] A. A. Nezhad, A. Miri, and D. Makrakis, "Location privacy and anonymity preserving routing for wireless sensor networks," *Computer Networks*, vol. 52, no. 18, pp. 3433–3452, 2008.
- [11] Samarati P. and Sweeney L., "Protecting privacy when disclosing information: k -anonymity and its enforcement through generalization and suppression," Tech. Rep., 1998.
- [12] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "The bikenet mobile sensing system for cyclist experience mapping," in *Proceedings of the 5th international conference on Embedded networked Sensor Systems (SenSys '07)*, pp. 87–101, ACM, New York, NY, USA, 2007.
- [13] L. Krishnamurthy, R. Adler, P. Buonadonna et al., "Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05)*, pp. 64–75, ACM, New York, NY, USA, 2005.
- [14] L. Selavo, A. Wood, Q. Cao et al., "Luster: wireless sensor network for environmental research," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys '07)*, pp. 103–116, ACM, New York, NY, USA, 2007.
- [15] V. Singhvi, A. Krause, C. Guestrin, J. Garrett, and S. Matthews, "Intelligent light control using sensor networks," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05)*, pp. ACM–218, New York, NY, USA, 2005.
- [16] N. Xu, S. Rangwala, K. K. Chintalapudi et al., "A wireless sensor network for structural monitoring," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys'04)*, pp. 13–24, New York, NY, USA, November 2004.
- [17] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *IEEE Symposium on Security and Privacy (SP '03)*, pp. 197–213, IEEE Computer Society, May 2003.
- [18] W. Trappe and L. Washington, *Introduction to Cryptography with Coding Theory*, Prentice Hall, 2002.
- [19] C. L. Hwang, J. L. Williams, and L. T. Fan, *Introduction to the Generalized Reduced Gradient Method*, Institute for Systems Design and Optimization, 1972.
- [20] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [21] W. Ding, G. Yan, and Z. Lin, "Self-deployment and coverage of mobile sensors within a bounded region," in *Proceedings of*

- the Chinese Control and Decision Conference*, pp. 3683–3688, 2009.
- [22] Rouche N., Habets P., and Laloy M., *Stability Theory by Lyapunov's Direct Methods*, Springer, 1977.
- [23] D. Dimarogonas and K. Kyriakopoulos, "An inverse agreement control strategy with application to swarm dispersion," in *Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 6148–6153, 2007.
- [24] Mixmaster Remailer, <http://mixmaster.sourceforge.net/>.
- [25] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services (MobiSys '03)*, pp. 31–42, ACM, 2003.
- [26] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," in *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM '05)*, pp. 194–205, IEEE Computer Society, 2005.
- [27] Ouyang Y., Le Z., Liu D., Ford J., and Makedon F., "Source location privacy against laptop-class attacks in sensor networks," in *Proceedings of the 4th international conference on Security and Privacy in Communication Networks (SecureComm '08)*, pp. 1–10, ACM, 2008.
- [28] M. Shao, Y. Yang, S. Zhu, and G. Cao, "Towards statistically strong source anonymity for sensor networks," in *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM'08)*, pp. 51–55, 2008.
- [29] W. Conner, T. Abdelzaher, and K. Nahrstedt, "Using data aggregation to prevent traffic analysis in wireless sensor networks," in *Proceedings of the International Conference on Distributed Computing in Sensor Networks (DCOSS '06)*, pp. 202–217, 2006.
- [30] A. Howard, M. Mataric, and G. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem," *Distributed Autonomous Robotic Systems*, vol. 5, pp. 299–308, 2002.
- [31] T. Balch and M. Hybinette, "Behavior-based coordination for large-scale robot formations," in *Proceedings of the 4th International Conference on Multiagent Systems*, pp. 363–364, 2000.

Research Article

Performance Analysis of Flow-Based Traffic Splitting Strategy on Cluster-Mesh Sensor Networks

Huimin She,¹ Zhonghai Lu,¹ Axel Jantsch,¹ Dian Zhou,² and Li-Rong Zheng¹

¹ School of Information and Communications Technology, KTH Royal Institute of Technology, 16440 Stockholm, Sweden

² School of Microelectronics, Fudan University, Shanghai 201203, China

Correspondence should be addressed to Huimin She, huimin@kth.se

Received 13 May 2011; Accepted 3 December 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Huimin She et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Performance analysis is crucial for designing predictable and cost-efficient sensor networks. Based on the network calculus theory, we propose a flow-based traffic splitting strategy and its analytical method for worst-case performance analysis on cluster-mesh sensor networks. The traffic splitting strategy can be used to alleviate the problem of uneven network traffic load. The analytical method is able to derive close-form formulas for the worst-case performance in terms of the end-to-end least upper delay bounds for individual flows, the least upper backlog bounds, and power consumptions for individual nodes. Numerical results and simulations are conducted to show benefits of the splitting strategy as well as validate the analytical method. The numerical results show that the splitting strategy enables much better balance on network traffic load and power consumption. Moreover, the simulation results verify that the theoretic bounds are fairly tight.

1. Introduction

With the advances of wireless communications and microelectronics technologies, wireless sensor networks (WSNs) have received more and more attention over the last decades due to their potential in a variety of application domains [1–4], such as environment monitoring, human activity tracking, healthcare, and military assistance, and so forth.

A typical sensor network consists of a larger number of sensor nodes that are capable of sensing the environment and forwarding their observation values to a fusion center (sink) through multihop wireless links. Thus, the traffic pattern in sensor networks is usually in a many-to-one manner. The nodes near the sink may need to forward more data and thus require more energy consumptions and bigger buffers than the nodes far away. Consequently, the distributions of energy and buffer requirements in WSNs are extremely uneven. However, energy supplies and buffers are limited and expensive resources in typical WSNs, since sensor nodes are usually made as tiny devices with limited buffers and equipped with batteries that may not be convenient or economical for replacement. One way to address these challenges is applying traffic splitting strategies which have been adopted

by many researchers for load balancing in communication networks [5, 6]. With traffic splitting, a main flow is divided into several subflows and forwarded to the destination through different routing paths. By distributing traffic over the network, the overall network load balance can be improved. It is shown in [7] that the spare capacity can be reduced and thus the overall performance of the system can be improved by splitting traffic across multiple disjoint paths.

One popular application of WSNs is real-time monitoring and tracking, such as logistic chain tracking [4] and healthcare application [8]. In such kind of applications, it is crucial to ensure sensor data delivered to the sink within time constraints so that appropriate actions can be made. In order to design a WSN with predictable delay, backlog, and energy consumptions, formal performance analysis is desired for analyzing a sensor network before its actual deployment. While simulation-based methods can offer high accuracy, it can be very time-consuming and tedious to find the worst-case performance. Each simulation run may take considerable time and evaluates only a single network configuration, traffic pattern, and load point. Hence, formal methods are desired to dimension sensor networks in an analytical way rather than case-by-case simulations. Starting with the initial

work by Cruz [9, 10], *network calculus* has been developed as a useful tool for the performance analysis of networked system [11]. In contrast to queueing theory, network calculus deals with performance bounds, such as worst-case delay and backlog bounds, rather than average values. It has been applied to sensor networks by many researchers recently [12–15].

In this paper, we propose a flow-based traffic splitting strategy and its analytical method for worst-case performance analysis on cluster-mesh sensor networks based on the network calculus theory. We introduce the flow-based traffic splitting strategy, which is useful in balancing not only network load but also power consumption. Aiming to evaluate the worst-case performance in terms of end-to-end least upper delay bound, least upper backlog bound, and power consumption, a splitting model is built for a single-node analysis and an analytical method is proposed for the network analysis. Through an example, we show that the performance analysis method is able to derive closed-form formulas of these bounds. The numerical results indicate that the backlog and power consumption can be balanced by applying the traffic splitting strategy. In addition, simulations are performed to validate the performance bounds of our analytical method. The results show that their tightnesses are satisfactory.

The rest of this paper is organized as follows. Section 2 introduces related work. Section 3 contains preliminaries including the cluster-mesh network topology and power consumption model, and basics of network calculus. Section 4 includes performance analysis of the flow-based traffic splitting strategy. An analysis example is given in Section 5. Numerical results and simulations are presented in Section 6. Finally, conclusions and directions for future work are given in Section 7.

2. Related Work

In general packet switching networks, network calculus provides methods to deterministically reason about timing properties and resource requirements. Based on the powerful abstraction of *arrival curve* for traffic flows and *service curve* for network elements, it allows computing the worst-case delay and backlog bounds. Systematic accounts of network calculus can be found in books [11, 16].

Network calculus has been extremely successful for ensuring performance bounds when applied to ATM, Internet, and other networks. It is recently extended and applied for performance analysis and resource dimensioning of WSNs by several researchers. In [12], Schmitt and Roedig firstly applied network calculus to sensor network and proposed a generic framework for performance analysis of WSNs with various traffic patterns. They further extended the general framework to incorporate computational resources besides the communication aspects of WSNs [14]. In [13], Kouba et al. proposed a methodology for the modeling and worst-case dimensioning of cluster-tree sensor networks. They derived plug-and-play expressions for the end-to-end delay bounds, buffering, and bandwidth requirements as a function of the WSN cluster-tree and traffic characteristics. Lenzini et al.

[17] proposed a method for deriving tight end-to-end *least upper delay bounds* in sink-tree networks. The least upper delay bound is defined as the minimum value of the upper delay bound. In [15], the authors presented a method for computing the worst-case delays, buffering, and bandwidth requirements while assuming that the sink node can be mobile.

Traffic splitting strategies have several common features with multipath routing protocols. There have been plenty of research works on multipath routing and traffic splitting for sensor networks [18–23]. The authors in [18] proposed a multipath routing scheme that finds several disjoint paths. In this scheme, the source node or an intermediate node chooses one path from the available paths to deliver the data to sink based on the performance requirements such as delay and throughput. An energy efficient multipath routing protocol for WSNs with multiple sinks is presented in [19]. The path construction is implemented by the source node sending route messages to its neighbors. Traffic is distributed over the multiple paths according to a load balancing algorithm. The results show that the proposed scheme results in a higher energy efficiency. In [20], authors proposed an N-to-1 multipath routing protocol, in which nodes are arranged in a spanning tree. Multipaths are constructed by traversing the tree. The multipath scheme is a combination of end-to-end multipath traffic dispersion and per-hop alternate path salvaging. Zou et al. [21] studied the interplay between data aggregation and flow splitting in WSNs and proposed a flow-based scheme. The flows are preserved until the aggregation point. The aggregated data is splitted into multiple flows on the rest of the path to the destination. The results show that the scheme can balance energy consumption and therefore prolong the lifetime of WSNs. In [22], the authors investigated a joint coding/routing optimization of network costs and capacity in WSNs. By combining the link rate allocation and network coding-based multipath routing, the total energy consumption of encoding power, transmission power and reception power can be reduced. A backpressure collection protocol (BCP) for sensor networks is presented in [23]. In this protocol, routing and forwarding decisions are made based on a per-packet basis. By using ETX optimization and floating LIFO queues, BCP is capable of improving throughput and delivery performance under static and dynamic settings, respectively.

This work applies the network calculus theory for analytical performance evaluation of a traffic splitting strategy on cluster-mesh sensor networks. Our work differs from others' works and contributes to state of the art in the following aspects. First, we address the particular problem of deriving performance bounds and resource requirements for a traffic splitting strategy on the cluster-mesh topology, which we believe are of great interest for time-sensitive WSN applications. Second, we introduce a flow-based traffic splitting strategy that can be used to balance traffic load in the network. We define a splitting model and set up an equivalent packet delivery model for the original network. Based on these models, the end-to-end least upper delay bounds and backlog bounds are derived. The results show that the variance of backlogs can be greatly reduced by applying the traffic

splitting strategy, which indicates better load balancing. Third, we conduct power consumption analysis, which is crucial for most applications of WSNs. The results indicate that the traffic splitting strategy also enables better balance on power consumptions. Since most applications of WSNs involve sensors with unreplaceable power supplies, better power balance would lead to longer lifetime of the whole network. On the other hand, even if the batteries of sensors can be recharged or replaced, better power balance would bring about less labor for recharging or replacing and thus can reduce the overall deployment costs. In addition, although our work focuses on performance analysis of a traffic splitting strategy on a particular network topology, we believe the intrinsic idea of the method is also very useful for analyzing sensor networks with other topologies and traffic planning policies.

We have described the traffic splitting scheme in our previous work [24, 25], from which we borrow many notations used in this paper. In this work, we have significantly extended and enhanced the previous work in the following aspects. First, in previous work, we only presented the analysis method without simulations. The analytical method in this work is validated through simulations which also prove the tightness of the delay and backlog bounds. Second, the end-to-end delay bound in [24, 25] is calculated by summing up the per-hop delay together. While, in this paper, the end-to-end delay bound is computed using the end-to-end equivalent service curve, the later method can get tighter bounds. A comparison between these two methods is shown in Figure 20. Third, we integrate power consumption analysis in this work, which we believe is of great importance for WSNs.

3. Preliminaries

This section presents system models, including the cluster-mesh network topology and power consumption model.

3.1. The Cluster-Mesh Topology. A wireless sensor network may consist of a large number of sensors that are densely deployed either inside the phenomenon of interest or close to it. These sensors can be organized in various topologies, such as mesh- and cluster-based topologies. The mesh networking has advantages like supporting path diversity which enables better balance on traffic load and energy consumption [26]. The cluster-based topologies are also quite suitable for WSNs with demanding requirements in terms of Quality of Service (QoS) support and real-time communications [13]. Considering these aspects, we adopt the *cluster-mesh* topology that merges advantages of mesh and cluster [3, 27]. It is a two-layered architecture with the mesh defining a backbone that consists of a set of *cluster heads* (CHs). A cluster is formed by grouping a number of sensors within a geographic neighborhood. We define the network composed by cluster heads and the sink as the layer-1 network and the network inside a cluster as the layer-2 network.

In summary, the cluster-mesh network contains three types of nodes: *sink*, *cluster head*, and *sensor*. Like in most

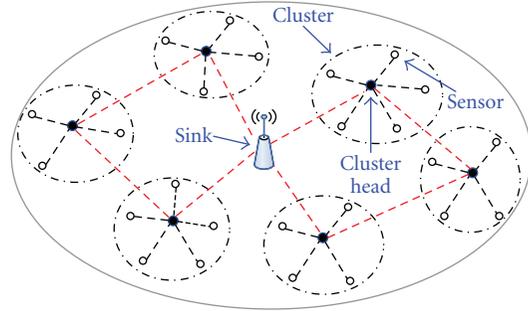


FIGURE 1: A cluster-mesh sensor network.

sensor networks, the sink is responsible for controlling the network and collecting data from all the other nodes. A cluster head and multiple sensors form a cluster. In order to reduce the cost and complexity, sensors do not communicate with each other and data generated by them is collected by their cluster head and delivered to the sink through neighbor cluster heads. For simplicity and conciseness, we consider cluster heads are static and they do not sense the environment and generate input data. However, this assumption can be easily relaxed, and the subsequent analysis is straightforward. In the mesh network composed by cluster heads and the sink, links are considered bidirectional.

Figure 1 shows an example of the cluster-mesh topology. A cluster-mesh network is a mesh network where each cluster head and its connected sensors form their own logical cluster. The layer-1 network can be modeled as a direct graph $G(\mathcal{N}, \mathcal{L})$, where \mathcal{N} is the set of all sensor nodes and the sink and \mathcal{L} is the set of all direct links in the network. In this paper, our work concentrates on analyzing the layer-1 network.

3.2. Power Consumption Model. In most types of sensor nodes, the energy consumption is mainly contributed by the transmitter, receiver, and computation module [28]. We consider the application scenario of sensor networks for fresh food monitoring in warehouses. In this scenario, sensors may perform tasks and send packets periodically. Consequently, the power consumption of the computation module can be considered as nearly constant denoted by p^c . Let ϵ^r denote the energy consumption of the receiver electronics for receiving one bit data. In practical applications, the power consumption of the receiving is usually stable [28]. So ϵ^r can be considered as a constant. According to the results in [29], the energy required to transmit a given amount of data is a convex and monotonically increasing function of the transmission rate, that is, the energy per bit can be expressed as (Figure 2)

$$\epsilon^t = \frac{N_0}{RG} (2^{R/\eta W} - 1), \quad (1)$$

where R is the transmission rate, W is the channel bandwidth, G is the channel gain, N_0 is the noise power, and $\eta \in (0, 1)$ is the probability that the information can be reliably transmitted at a given transmission rate. (In an optimal

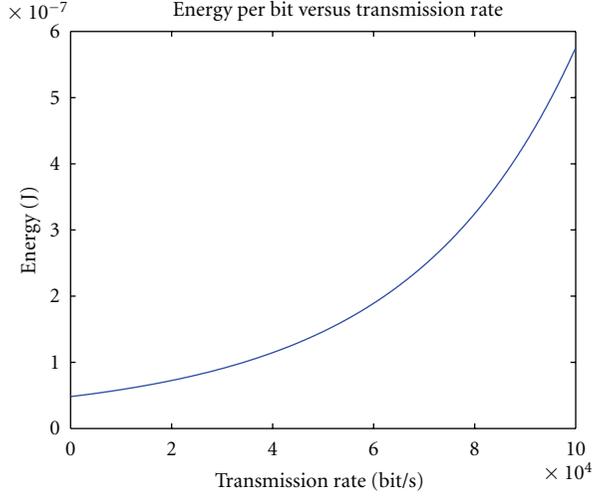


FIGURE 2: Energy per bit versus transmission rate [29]: $W = 20$ kHz, $N_0 = -100$ dB, $G_0 = -50$ dB, $d = 5$ m, $\theta = 3$.

channel coding scheme, a transmission rate $R = \eta C$ can be guaranteed for any $0 < \eta < 1$, where $C = W \log_2(1 + GP/N_0)$ is the Shannon capacity [29].) We adopt a simplified model for channel gain, that is, $G = G_0(d/d_0)$ (Section 2.6 of [30]), where θ is the path loss exponent ($2 \leq \theta \leq 4$), d is the distance between the transmitter and receiver, d_0 is a reference distance where the reference channel gain G_0 is measured.

Therefore, the total power consumption P_i of a node i can be expressed as

$$\begin{aligned} P_i &= \epsilon^r \sum_{k \in N_{\text{in}}(i)} \rho_{ki} + \sum_{j \in N_{\text{out}}(i)} \epsilon_{ij}^t \rho_{ij} + p^c \\ &= \epsilon^r \sum_{k \in N_{\text{in}}(i)} \rho_{ki} + \sum_{j \in N_{\text{out}}(i)} \frac{\rho_{ij} N_0}{R_{ij} G_{ij}} (2^{R_{ij}/\eta W} - 1) + p^c, \end{aligned} \quad (2)$$

where ρ_{ki} and ρ_{ij} denote the data rates on link ki and ij , respectively, R_{ij} denotes the transmission rate of node i sending data to node j , $G_{ij} = G_0(d_{ij}/d_0)^{-\theta}$ denotes the channel gain between node i and j , $N_{\text{in}}(i)$ denotes the set of nodes which are the direct sources of incoming data flows of node i , and $N_{\text{out}}(i)$ denotes the set of nodes which are the direct destinations of output data flows of node i .

3.3. Traffic Model and Service Model. As stated in the previous section, sensor nodes inside a cluster generate input data and then send them to their cluster head. A traffic flow is defined as an infinite stream of data from a source to a destination. Following network calculus, we model the input flow at a cluster head using its cumulative traffic $F(t)$, defined as the number of bits coming from the flow in time interval $[0, t]$. Furthermore, we use a wide-sense increasing function $\alpha(t)$ to constrain this cumulative traffic flow $F(t)$, defined as

$$F(t) - F(s) \leq \alpha(t - s), \quad \forall t \geq 0, t \geq s, \quad (3)$$

where $\alpha(t)$ is called the arrival curve of the input flow $F(t)$ [11]. *Affine arrival curve* is one of the most commonly used

arrival curves, which has been adopted in many works [12–15]. The application scenario of this work is real-time monitoring, and the sensor nodes sense the environment and send packets periodically to the cluster heads. Therefore, the affine arrival curve model can be used to abstract the input traffic of cluster heads, defined as $\alpha(t) = \gamma_{\sigma, \rho} = \rho \cdot t + \sigma$, where σ and ρ represent the burst tolerance (in bits) and the average data rate (in bits/s), respectively. Figure 3(a) shows examples of a periodic cumulative flow $F(t)$ and an affine arrival curve $\alpha(t)$.

Service curve is an abstraction to model the processing capability of a node, depending on link layer characteristics, such as transmission rate, channel characteristics, and packet scheduling. The node and the channel together are modeled as a network element which provides a service curve β^s to the input flows. If the node forwards packets with the rate R (bits/s) and delays packets for T (s) at maximum due to scheduling and queuing, it can be modeled by a *rate-latency* service curve [11] that consists of two components: a *rate service curve* and a *latency service curve*. The rate-latency service curve can be formally defined as $\beta(t) = \chi_{R, T} = R[t - T]^+$, where notation $[x]^+$ denotes $\max\{0, x\}$.

In wireless networks, data transmission over wireless channels is usually unreliable due to their inherent uncertainties. The actual transmission rate and success probability are influenced by the transmission power, path loss, noise power, and interference. In spite of these uncertainties, deterministic network calculus can still be useful in modeling wireless networks by making reasonable assumptions and abstractions. First, the uncertainties in some applications of WSNs are low. An example scenario for which our framework suits well is the process monitoring and tracking in logistics systems [4]. Second, the link unreliability and data loss rate can be mitigated by applying high transmission powers, especially for the cases with small distances between a transmitter and a receiver. Third, the interference between adjacent nodes can be alleviated by using appropriate MAC layer protocols. There are plenty of research works on designing TDMA-based link protocols which can create collision-free slot schedules [31, 32].

Based on these assumptions about link reliability and interference, we can abstract and approximate the service capability of a node by a deterministic service curve with the idea of effective transmission rate. From information theory, the Shannon capacity of a wireless channel can be expressed as $C = W \log_2(1 + GP/N_0)$, where W, G, N_0 are the same as those in (1). The service rate is defined as the rate that the information can be reliably transmitted. With an optimal or suboptimal channel coding scheme, a service rate of $R = \eta \cdot C$ can be guaranteed for any $0 < \eta < 1$ [29]. R defines a lower bound on the transmission rate. Therefore, the rate-latency service curve $\beta(t) = \chi_{R, T} = R[t - T]^+$ can be applied to model the service capability of a wireless channel, where T denotes the maximum possible processing/queueing delay.

3.4. Delay, Backlog, and Output Bounds. Given the arrival curve and service curve of a node, the least upper delay bound, least upper backlog bound, and output bounds can be derived according to network calculus [11]. The least

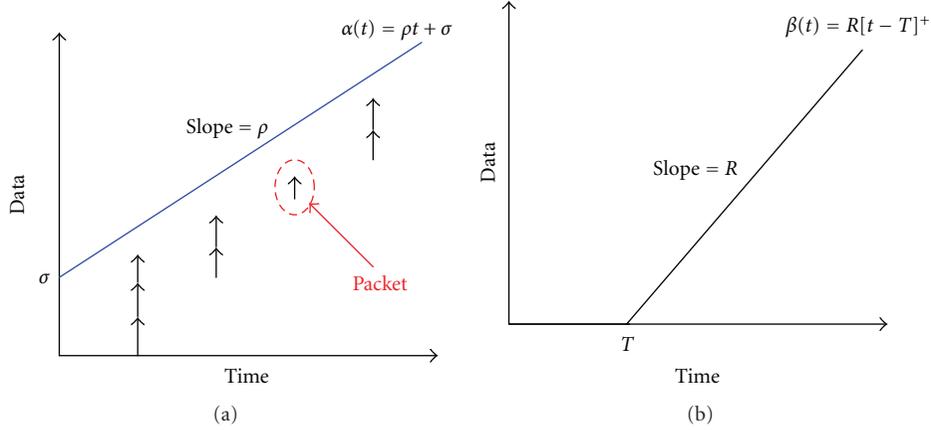


FIGURE 3: (a) An affine arrival curve: the arrows show the packet generation process. (b) A rate-latency service curve.

upper backlog bound is defined as the minimum value of the upper backlog bound. Consider a node i provides a service curve β_i^s to the input flow which is constrained by an arrival curve α_i . According to network calculus, the least upper delay bound of the flow can be computed by

$$D_i = h(\alpha_i, \beta_i^s) = \sup_{t \geq 0} \left\{ \inf_{\tau \geq 0} \{ \alpha_i(t) \leq \beta_i^s(t + \tau) \} \right\}. \quad (4)$$

Moreover, the least upper backlog bound of node i can be calculated by

$$B_i = v(\alpha_i, \beta_i^s) = \sup_{t \geq 0} \{ \alpha_i(t) - \beta_i^s(t) \}. \quad (5)$$

Additionally, the arrival curve of the departure flow can be derived by

$$\tilde{\alpha}_i = \alpha_i \circ \beta_i^s = \sup_{\tau \geq 0} \{ \alpha_i(t + \tau) - \beta_i^s(\tau) \}. \quad (6)$$

4. Analysis of the Flow-Based Traffic Splitting Strategy

In this section, we first introduce the splitting and multiplexing models. Then, the formal performance analysis procedure is presented. After that, we discuss the scope and assumptions of our analysis approach.

4.1. The Splitting Model. To analyze the splitting strategy, we build a splitting model that identifies the relations of input, output, delay, and backlog for a single node. Without losing generality, we consider a main flow is split into two subflows. The node f_1 traverse is abstracted as the combination of a buffer plus a *splitter* depicted in Figure 4(a).

We consider that the node performs a weighted proportional splitting scheme, in which the main flow is split according to the configured weights, ϕ_i for subflow i . In each round, the splitter will try to forward ϕ_i packets to output link i before moving to the next one. The values of ϕ_i can be set either according to a predefined rule or randomly. Increasing the value of ϕ_i can result in increased packets to

output link i . By adjusting ϕ_i , the amount of traffic over each link can be controlled. If the service rate is R bits/s, the maximum length of a round is consequently equal to $\sum_i \phi_i l / R$ seconds and the time for packets of subflow i to be forwarded within a round is bounded by $\phi_i l / R$ seconds, where l is the packet length. In the weighted proportional splitting scheme, the worst case appears when the packets of a subflow just misses its turn in the current round. Consequently, it will have to wait for its turn at the next round. In the worst case, packets of the subflow i have to wait up to $\sum_{i \neq j} \phi_j l / R$ seconds to be served.

Consider a main flow f_1 that is upper constrained by arrival curve $\alpha_1 = \gamma_{\sigma_1, \rho_1}$, be split into two subflows $f_{1.1}$ and $f_{1.2}$ according to weights ϕ_1 and ϕ_2 , where σ_1 and ρ_1 denote the burstiness and average data rate of f_1 , respectively. Burstiness is defined as the amount of data inputted/outputted to/from a system or a node at one time. Consequently, it should be equal or bigger than the packet size l . Let $\alpha_{1.1} = \gamma_{\sigma_{1.1}, \rho_{1.1}}$ and $\alpha_{1.2} = \gamma_{\sigma_{1.2}, \rho_{1.2}}$ denote the arrival curves of $f_{1.1}$ and $f_{1.2}$, respectively. Then, we have

$$\begin{aligned} \rho_{1.1} &= \frac{\phi_1}{\phi_1 + \phi_2} \rho_1, & \sigma_{1.1} &= \max \left(\left\lceil \frac{\phi_1 \sigma_1}{(\phi_1 + \phi_2) l} \right\rceil \cdot l, l \right), \\ \rho_{1.2} &= \frac{\phi_2}{\phi_1 + \phi_2} \rho_1, & \sigma_{1.2} &= \max \left(\left\lceil \frac{\phi_2 \sigma_1}{(\phi_1 + \phi_2) l} \right\rceil \cdot l, l \right), \end{aligned} \quad (7)$$

where $\lceil \cdot \rceil$ denotes the minimum integer equal to or bigger than the number inside.

Let the splitter provide a service curve $\beta^s = \chi_{R, T}$. Since the splitter serves a subflow at one time, the service rate for each subflow also equals R . Therefore, the equivalent service curve for subflow $f_{1.1}$ can be derived by

$$\hat{\beta}_1^s = \beta^s \otimes \delta_{\phi_2 l / R} = \chi_{R, T + \phi_2 l / R}. \quad (8)$$

Analogously, the equivalent service curve for $f_{2.2}$ is

$$\hat{\beta}_2^s = \beta^s \otimes \delta_{\phi_1 l / R} = \chi_{R, T + \phi_1 l / R}. \quad (9)$$

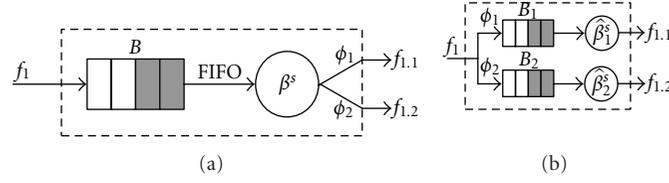


FIGURE 4: (a) The main flow f_1 is split into two subflows $f_{1,1}$ and $f_{1,2}$. (b) The equivalent model.

Furthermore, the equivalent bounds on backlogs can be calculated by

$$\begin{aligned} B_1 &= \sigma_{1,1} + \frac{\phi_1 \rho_1}{\phi_1 + \phi_2} \left(T_1 + \frac{\phi_2 l}{R} \right), \\ B_2 &= \sigma_{1,2} + \frac{\phi_2 \rho_1}{\phi_1 + \phi_2} \left(T_1 + \frac{\phi_1 l}{R} \right). \end{aligned} \quad (10)$$

Therefore, the least upper bound of the total backlog is computed by

$$B = B_1 + B_2 = \sigma_1 + \rho_1 \left[T + \frac{2\phi_1 \phi_2 l}{R(\phi_1 + \phi_2)} \right]. \quad (11)$$

The least upper delay bounds consist of three parts: the processing time, the time to serve input burstiness, and the scheduling delay. Let $D_{1,1}$ and $D_{1,2}$ denote the delay bounds of subflow $f_{1,1}$ and $f_{1,2}$, respectively. They can be computed by

$$D_{1,1} = T + \frac{\sigma_{1,1}}{R} + \frac{\phi_2 l}{R}, \quad D_{1,2} = T + \frac{\sigma_{1,2}}{R} + \frac{\phi_1 l}{R}. \quad (12)$$

Furthermore, the departure arrival curves of $f_{1,1}$ and $f_{1,2}$ can be derived by

$$\begin{aligned} \tilde{\alpha}_{1,1} &= \frac{\phi_1 \rho_1}{\phi_1 + \phi_2} t + \sigma_{1,1} + \frac{\phi_1}{\phi_1 + \phi_2} \left(\rho_1 T + \frac{\rho_1 \phi_2 l}{R} \right), \\ \tilde{\alpha}_{1,2} &= \frac{\phi_2 \rho_1}{\phi_1 + \phi_2} t + \sigma_{1,2} + \frac{\phi_2}{\phi_1 + \phi_2} \left(\rho_1 T + \frac{\rho_1 \phi_1 l}{R} \right). \end{aligned} \quad (13)$$

4.2. The Multiplexing Model. In order to analyze resource sharing when multiple input flows share the bandwidth of a link at a node, we propose a multiplexing model. We shall use this model for analyzing a network with various traffic flowing scenarios.

Without loss of generality, let us consider a node serve two flows f_1 and f_2 in the FIFO order as shown by Figure 5(a). And its equivalent model is drawn in Figure 5(b). Let the node provide a service curve β^s to the aggregating flows, and f_1 and f_2 have α_1 and α_2 as arrive curves, respectively. We define $\hat{\beta}_1^s = \kappa(\beta^s, \alpha_2)$ as the *equivalent service curve* [17] provided to flow f_1 , where $\kappa(\cdot, \cdot)$ is an operator to compute the equivalent service and τ is an intermediate argument. Thus, the departure arrival curve of f_1 can be derived by $\tilde{\alpha}_1 = \alpha_1 \circ \kappa(\beta^s, \alpha_2)$, and its least upper delay bound is computed by $h(\alpha_1, \kappa(\beta^s, \alpha_2))$, and the least upper backlog bound of the node is $v(\alpha_{\{1,2\}}, \beta^s)$,

where $\alpha_{\{1,2\}}$ denotes the arrival curve of the aggregating flow $f_{\{1,2\}}$. Similarly, the equivalent service curve provides to flow f_2 can be derived by $\hat{\beta}_2^s = \kappa(\beta^s, \alpha_1)$, and its delay and backlog bounds, and departure arrival curve can be derived accordingly.

We give an example to show how to compute $\kappa(\cdot, \cdot)$. Let $\beta^s(t) = \chi_{R,T} = R[t - T]^+$ and $\alpha_2(t) = \gamma_{\sigma_2, \rho_2}(t) = \rho_2 t + \sigma_2$, then, applying Corollary 4.5 in [17], the equivalent service curve for f_1 can be calculated by

$$\hat{\beta}_1^s = \kappa(\beta^s, \alpha_2) = \gamma_{R-\tau, R-\rho_2} \otimes \delta_{T+\sigma_2/R+\tau} \quad (\tau \geq 0), \quad (14)$$

where τ is an intermediate argument for calculating the least upper delay bound, and $\delta_T(t) = +\infty$ for $t > T$, and 0 otherwise.

The least upper delay bound of f_1 is calculated by

$$D_1 = h(\alpha_1, \hat{\beta}_1^s) = \inf_{\tau \geq 0} \left\{ T + \frac{\sigma_2}{R} + \frac{\sigma_1 - R\tau}{R - \rho_2} + \tau \right\}. \quad (15)$$

Furthermore, the least upper backlog bound of the node can be derived by

$$B = \sigma_1 + \sigma_2 + (\rho_1 + \rho_2)T. \quad (16)$$

Additionally, the arrival curve of the departure flow of f_1 is computed by

$$\tilde{\alpha}_1 = \alpha_1 \circ \hat{\beta}_1^s = \rho_1 t + \sigma_1 + \rho_1 \left(T + \frac{\sigma_2}{R} + \tau \right), \quad (17)$$

where τ is the same as the value obtained in (15).

4.3. The Splitting-Based Performance Analysis Procedure.

There have been several research works on the traffic splitting strategy in packet networks [6, 7], due to its efficiency in load balancing. In the flow-based splitting strategy, a traffic flow is split into multiple subflows at its source node, and these subflows are forwarded to the sink through different routing paths. The source node decides how the subflows are split. Given the traffic patterns, service models, the routing protocols, and the splitting strategy, we then detail the general performance analysis procedure as follows.

Step 1. Based on the traffic pattern, routing protocols, and the traffic splitting strategy, construct a performance analysis model that converts the original network into an equivalent network.

Step 2. Derive the input and departure arrival curves of all nodes in the network based on network calculus.

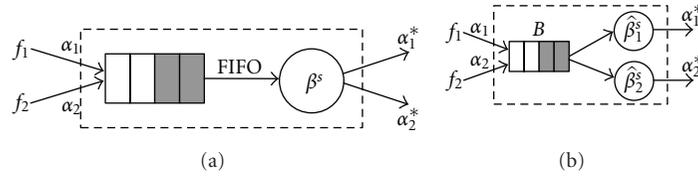


FIGURE 5: (a) A node serves two input flows. (b) The equivalent model.

Step 3. Derive the end-to-end equivalent service curves for the subflows; then compute the end-to-end least upper delay bound for the main flow using $D = h(\alpha, \hat{\beta})$, where α denotes the input arrival curve and $\hat{\beta}$ denotes the end-to-end equivalent service curve.

Step 4. Using the results in Step 2, compute the least upper backlog bound of each node $B_s = v(\sum_{i \in \mathcal{I}(s)} \alpha_i^s, \beta^s)$, where $\mathcal{I}(s)$ represents the set of input flows of node s .

Step 5. Compute the power consumption of a node s by $P_s = \epsilon^r \sum_{i \in N_{in}(s)} \rho_{is} + \sum_{j \in N_{out}(s)} \epsilon_{sj}^t \rho_{sj} + p_s^c$, where $N_{in}(i)$ and $N_{out}(i)$ denote the set of nodes which are the direct sources of incoming data flows and the direct destinations of output data flows of node i , respectively.

4.4. Discussions

4.4.1. Flow-Based Splitting versus Multipath Routing. Basically, a traffic splitting process consists of two stages: establishing multiple routing paths and allocating traffic on each path according to the splitting strategy. Multipath routing is a technique exploiting routing diversity by using multiple source-destination pairs. It has been receiving plenty of research attentions [33, 34]. There are plenty of works in the literature on how to set up multiple routing paths in ad hoc networks [18, 34, 35]. Our work focuses on analyzing the performance of the splitting strategy rather than finding multipath routes. We assume that multiple paths have already been established between source nodes and the sink.

There are several common features between multipath routing and flow-based traffic splitting: first, both of them use multipaths to explore routing diversities; second, both of them aim for improving load balance. Apart from these common features, there exist significant differences between them. In multipath routing, routing decisions are made on a per-packet basis, this is, each packet chooses its routing path and is forwarded to the destination individually. Multipath routing is mainly used for improving network performance in terms of reliability and robustness [33, 34]. While in flow-based splitting strategy, the routing and forwarding is made on a per-flow basis. So it is capable of realizing a controlled splitting and providing quality of service. For example, if there are two paths between a source and a destination, a flow may be split half to one path and half to the other. So the delay guarantees can be reasoned about.

4.4.2. Flow-Based Splitting versus Node-Based Splitting. According to the way that a traffic flow split, the traffic splitting strategy can be classified into two categories: flow-based splitting and node-based splitting.

In flow-based splitting, the source node decides how the subflows are scheduled and split. The subflows can be identified after splitting. As shown in Figure 6(a), the traffic flow f_i is split into two subflows ($f_{i,1}$ and $f_{i,2}$) only at its source node s_1 and forwarded to the sink (s_6) through two routing paths: $\{s_1, s_2, s_4\}$ and $\{s_1, s_3, s_5\}$. In the node-based traffic splitting strategy (Figure 6(b)), the traffic flow can be split at the intermediate nodes that have multiple output links (such as nodes s_1, s_2, s_3 in Figure 6(b)), and these nodes decide how their input flows are allocated to their output links. An example of node-based routing strategy is the backpressure-based routing protocol (BCP) [23]. Implementing a flow-based traffic splitting strategy is more complex than a node-based traffic splitting strategy in practice, but a flow-based splitting strategy also has its own advantages. In the flow-based splitting strategy, the routing and forwarding decisions are made on a per-flow basis. So it is capable of realizing a controlled splitting and satisfying quality of service requirements.

4.4.3. How to Set Splitting Parameters? Our work mainly provides a framework for quality of service analysis of the flow-based splitting strategy. Another research issue is on splitting parameter exploration, that is, how to set splitting factors? One way is to utilize static network state information, such as link capacity and buffer length, to set splitting parameters. For example, in Figure 6(a), source node S_1 can select appropriate splitting factors based on the link capacity information of its downstream links. A larger amount of traffic can be allocated to the links with higher capacity. In [36], authors presented an explicit rate-based flow control scheme, in which each route ran a proportional max-min fair bandwidth sharing algorithm to divide the measure bandwidth among the passed flows. Alternatively, the splitting decision can be made based on dynamical network state information. Each node records its current or historic buffer lengths, and the information is sent to the source node, so that the source node can select appropriate splitting parameters. For example, authors in [37] proposed a congestion-aware routing scheme which could redirect a certain amount of traffic to other paths under heavy traffic load. In this scheme, the congestion status information at each route is detected depending on the average MAC layer utilization and queue length. If congestion happens, traffic is split into other paths according to its services type.

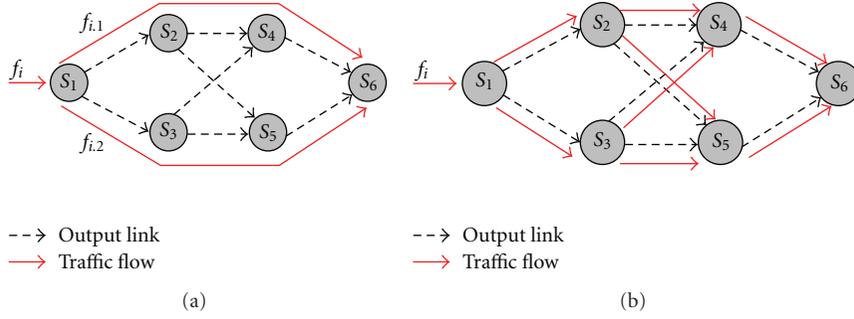


FIGURE 6: An example of splitting strategies. s_1 is the source node of the traffic flow, and s_6 is the sink. (a) Flow-based traffic splitting strategy; (b) node-based traffic splitting strategy.

5. An Analysis Example

In this section, we exemplify the general performance analysis and derive close-form formulas for of delay bounds, backlog bounds, and power consumptions under the conditions of affine arrival curve and rate-latency service curve. Consider a network consists of four nodes as shown in Figure 7. We define the *tagged main flow* as the main flow for which we shall derive the end-to-end delay bound. In this example, we choose f_1 as the tagged main flow. Let f_1 be constrained by the arrival curve $\alpha_1 = \gamma_{\sigma_1, \rho_1}$, and split into two subflows $f_{1,1}$ and $f_{1,2}$ and, respectively, traverse two different routing paths $\mathcal{R}(f_{1,1}) = \{s_1, s_2, s_4\}$ and $\mathcal{R}(f_{1,2}) = \{s_1, s_3, s_4\}$ from the source node s_1 to the sink. ϕ_1 and ϕ_2 are the splitting weights of $f_{1,1}$ and $f_{1,2}$, respectively. f_2 is the contention flow that is modeled by the arrival curve $\alpha_2 = \gamma_{\sigma_2, \rho_2}$. Let $\alpha_j^{s_i}$ and $\tilde{\alpha}_j^{s_i}$, respectively, denote the input and departure arrival curves of flow j at node s_i . Next, we need to derive the end-to-end least upper delay bound for flow f_1 and the least upper backlog bound and power consumption for each node.

In order to compute the end-to-end least upper delay and the least upper backlog bounds, we first need to derive the input and departure arrival curves of each node.

5.1. Arrival Curves of Input and Output. According to the results of the splitting model, the arrival curves of departure flows at node s_1 can be derived as

$$\begin{aligned}\tilde{\alpha}_{1,1}^{s_1} &= \frac{\phi_1 \rho_1}{\phi_1 + \phi_2} t + \sigma_{1,1} + \frac{\phi_1}{\phi_1 + \phi_2} \left(\rho_1 T_1 + \frac{\rho_1 \phi_2 l}{R_1} \right), \\ \tilde{\alpha}_{1,2}^{s_1} &= \frac{\phi_2 \rho_1}{\phi_1 + \phi_2} t + \sigma_{1,2} + \frac{\phi_2}{\phi_2} \left(\sigma_1 + \rho_1 T_1 + \frac{\rho_1 \phi_1 l}{R_1} \right).\end{aligned}\quad (18)$$

Node s_2 has two input flows, with arrival curves α_2 and $\alpha_{1,1}^{s_2} = \tilde{\alpha}_{1,1}^{s_1}$. According to the multiplexing analysis results (Section 4.2), we can derive the arrival curves of the departure flows of node s_2 as following

$$\begin{aligned}\tilde{\alpha}_2^{s_2} &= \rho_2 t + \sigma_2 + \rho_2 \left(T_2 + \frac{\sigma_{1,1}^{s_2}}{R_2} + \tau_1 \right), \\ \tilde{\alpha}_{1,1}^{s_2} &= \tilde{\alpha}_{1,1}^{s_1} + \frac{\phi_1 \rho_1}{\phi_1 + \phi_2} \left(T_2 + \frac{\sigma_2}{R_2} + \tau_2 \right),\end{aligned}\quad (19)$$

where $\sigma_{1,1}^{s_2} = \sigma_{1,1} + (\phi_1 / (\phi_1 + \phi_2)) (\rho_1 T_1 + \rho_1 \phi_2 l / R_1)$, τ_1 and τ_2 are defined by

$$\begin{aligned}\arg \min_{\tau_1} \zeta_1(x) &= \left\{ \tau_1 \geq 0 : \frac{\sigma_2 - R_2 \tau_1}{R_2 - \phi_1 \rho_1 / (\phi_1 + \phi_2)} + \tau_1 \right\}, \\ \arg \min_{\tau_2} \zeta_2(x) &= \left\{ \tau_2 \geq 0 : \frac{\sigma_{1,1}^{s_2} - R_2 \tau_2}{R_2 - \rho_2} + \tau_2 \right\}.\end{aligned}\quad (20)$$

For node s_3 , the arrival curve of its input flow is $\alpha_{1,2}^{s_3} = \tilde{\alpha}_{1,2}^{s_1}$, and it provides a service curve $\beta^{s_3} = \chi_{R_3, T_3}$. Consequently, the arrival curve of its departure flow is

$$\tilde{\alpha}_{1,2}^{s_3} = \tilde{\alpha}_{1,2}^{s_1} + \frac{\phi_2 \rho_1}{\phi_1 + \phi_2} T_3.\quad (21)$$

According to the connection relations, we can get the arrival curves of three input flows at node s_4 , which are $\alpha_2^{s_4} = \tilde{\alpha}_2^{s_2}$, $\alpha_{1,1}^{s_4} = \tilde{\alpha}_{1,1}^{s_2}$, and $\alpha_{1,2}^{s_4} = \tilde{\alpha}_{1,2}^{s_3}$. Since it is not necessary to compute the arrival curves of the departure flows at node s_4 , we omit the derivation here.

5.2. The End-to-End Delay Bound. In order to compute the end-to-end delay bound, we first need to derive the service curve provided by individual nodes. Let $\hat{\beta}_k^{s_i}$ represent the equivalent service curve provided by node s_i to its k th input flow. As shown in Figure 7, node s_2 serves two flow $f_{1,1}$ and f_2 . According to the multiplexing analysis in Section 4.2, the equivalent service curve for $f_{1,1}$ at s_2 is $\hat{\beta}_2^{s_2} = \kappa(\beta^{s_2}, \alpha_2)$. Node s_4 serves three flows, and the equivalent service curve for $f_{1,1}$ is $\hat{\beta}_1^{s_4} = \kappa(\kappa(\beta^{s_4}, \alpha_2^{s_4}), \alpha_{1,1}^{s_4})$. Thus, the end-to-end equivalent service curve for $f_{1,1}$ can be derived by

$$\begin{aligned}\beta_{1,1}^{e2e} &= \hat{\beta}_1^{s_1} \otimes \hat{\beta}_2^{s_2} \otimes \hat{\beta}_1^{s_4} \\ &= \beta^{s_1} \otimes \delta_{\phi_2 l / R_1} \otimes \kappa(\beta^{s_2}, \alpha_2) \otimes \kappa(\kappa(\beta^{s_4}, \alpha_2^{s_4}), \alpha_{1,1}^{s_4}) \\ &= \chi_{R_1, T_1} \otimes \delta_{\phi_2 l / R_1} \otimes \gamma_{R_2 \tau_2, R_2 - \rho_2} \otimes \delta_{T_2 + \sigma_2 / R_2 + \tau_2} \otimes \gamma_{R_4 \tau_3, R_4 - \rho_4'} \\ &\quad \otimes \delta_{T_4 + \sigma_4' / R_4 + \tau_3} \\ &= \chi_{R_1, T_1} \otimes \delta_{\phi_2 l / R_1 + T_2 + \sigma_2 / R_2 + \tau_2 + T_4 + \sigma_4' / R_4 + \tau_3} \otimes \gamma_{R_2 \tau_2, R_2 - \rho_2} \\ &\quad \otimes \gamma_{R_4 \tau_3, R_4 - \rho_4'}\end{aligned}\quad (22)$$

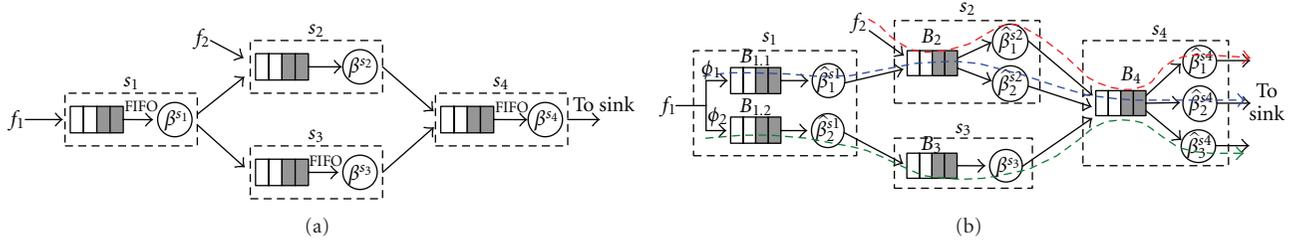


FIGURE 7: (a) A network analysis example: the main flow f_1 is split into two subflows $f_{1.1}$ and $f_{1.2}$. f_2 is the contention flow. (b) The equivalent analysis network: the blue, green, and red dashed lines show the routing path of flow $f_{1.1}$, $f_{1.2}$, and f_2 , respectively.

where τ_1 and τ_2 are the same as those in (19), τ_3 is calculated by $\arg \min_{\tau_3 \geq 0} \{ (\sigma_{1.1}^{s_4} - R_4 \tau_3) / (R_4 - \rho'_4) + \tau_3 \}$, $\rho'_4 = \rho_2 + \phi_2 \rho_1 / (\phi_1 + \phi_2)$ and

$$\begin{aligned} \sigma'_4 &= \rho_2 \left(T_2 + \frac{\sigma_{1.1}^{s_2}}{R_2} + \tau_1 \right) + \sigma_{1.2} \\ &+ \frac{\phi_2 [\rho_1 (T_1 + T_3) + \rho_1 \phi_1 l / R_1]}{\phi_1 + \phi_2} + \sigma_2. \end{aligned} \quad (23)$$

Analogously, the end-to-end equivalent service curve for $f_{1.2}$ can be derived by

$$\begin{aligned} \beta_{1.2}^{e2e} &= \hat{\beta}_2^{s_1} \otimes \beta^{s_3} \otimes \hat{\beta}_3^{s_4} \\ &= \beta^{s_1} \otimes \delta_{\phi_1 l / R_1} \otimes \beta^{s_3} \otimes \kappa(\kappa(\beta^{s_4}, \alpha_2^{s_4}), \alpha_{1.1}^{s_4}) \\ &= \chi_{R_1, T_1} \otimes \delta_{\phi_1 l / R_1} \otimes \chi_{R_3, T_3} \otimes \gamma_{R_4 \tau_4, R_4 - \rho'_4} \otimes \delta_{T_4 + \sigma'_4 / R_4 + \tau_4}, \end{aligned} \quad (24)$$

where τ_4 is defined by $\arg \min_{\tau_4 \geq 0} \{ (\sigma_{1.1}^{s_4} - R_4 \tau_4) / (R_4 - \rho'_4) + \tau_4 \}$, $\rho'_4 = (\rho_2 + \phi_1 \rho_1 / (\phi_1 + \phi_2))$ and

$$\begin{aligned} \sigma'_4 &= \frac{\phi_1 \rho_1}{\phi_1 + \phi_2} \left(T_1 + T_2 + \frac{\sigma_2}{R_2} + \tau_2 \right) + \frac{\rho_1 \phi_1 \phi_2 l}{R_1 (\phi_1 + \phi_2)} \\ &+ \rho_2 \left(T_2 + \frac{\sigma_{1.1}^{s_2}}{R_2} + \tau_1 \right) + \sigma_{1.1} + \sigma_2, \end{aligned} \quad (25)$$

where τ_1 and τ_2 are the same as those in (19).

After we get the end-to-end service curves, the least upper delay bounds of $f_{1.1}$ and $f_{1.2}$ can be, respectively, computed by $h(\alpha_{1.1}, \beta_{1.1}^{e2e})$ and $h(\alpha_{1.2}, \beta_{1.2}^{e2e})$,

$$\begin{aligned} h(\alpha_{1.1}, \beta_{1.1}^{e2e}) &= T_1 + T_2 + T_4 + \frac{\phi_2 l}{R_1} + \frac{\sigma_2}{R_2} + \frac{\sigma'_4}{R_4} \\ &+ \inf_{\substack{\tau_2 \geq 0 \\ \tau_3 \geq 0}} \left\{ \tau_2 + \tau_3 + \left[\frac{\sigma_{1.1}}{R_1} \vee \frac{\sigma_{1.1} - R_2 \tau_2}{R_2 - \rho_2} \vee \frac{\sigma_{1.1} - R_4 \tau_3}{R_4 - \rho'_4} \right] \right\}, \end{aligned}$$

$$\begin{aligned} h(\alpha_{1.2}, \beta_{1.2}^{e2e}) &= T_1 + T_3 + T_4 + \frac{\phi_1 l}{R_1} + \frac{\sigma'_4}{R_4} \\ &+ \inf_{\tau_4 \geq 0} \left\{ \tau_4 + \left[\frac{\sigma_{1.2}}{R_1} \vee \frac{\sigma_{1.2}}{R_3} \vee \frac{\sigma_{1.2} - R_4 \tau_4}{R_4 - \rho'_4} \right] \right\}. \end{aligned} \quad (26)$$

Hence, the end-to-end least upper delay bound for the flow f_1 equals the maximum of the delays of two subflows, namely,

$$D_{f_1} = \max \{ h(\alpha_{1.1}, \beta_{1.1}^{e2e}), h(\alpha_{1.2}, \beta_{1.2}^{e2e}) \}. \quad (27)$$

5.3. The Backlog Bound. Let B_{s_i} denote the backlog bound of node s_i ($i = 1, \dots, 4$). As we have already derived the arrival curves of input and output flows at each node, its least upper backlog bound can be calculated very easily. According to the result in (11), we have

$$B_{s_1} = B_{1.1} + B_{1.2} = \sigma_1 + \rho_1 \left[T_1 + \frac{2\phi_1 \phi_2 l}{R_1 (\phi_1 + \phi_2)} \right]. \quad (28)$$

Node s_2 has two input flows α_2 and $\alpha_{1.1}^{s_2}$, so its least upper backlog bound is computed by

$$B_{s_2} = \sigma_{1.1} + \sigma_2 + \frac{\phi_1 (\rho_1 T_1 + \rho_1 \phi_2 l / R_1)}{\phi_1 + \phi_2} + \left(\frac{\phi_1 \rho_1}{\phi_1 + \phi_2} + \rho_2 \right) T_2. \quad (29)$$

Analogously, the least upper backlog bounds of node s_3 and s_4 can be derived by

$$\begin{aligned} B_{s_3} &= \sigma_{1.2} + \frac{\phi_2}{\phi_1 + \phi_2} \left[\rho_1 (T_1 + T_3) + \frac{\rho_1 \phi_1 l}{R_1} \right], \\ B_{s_4} &= \rho_1 (T_1 + T_4) + \sigma_1 + \rho_2 \left(T_2 + T_4 + \frac{\sigma_{1.1}^{s_2}}{R_2} + \tau_1 \right) + \sigma_2 \\ &+ \frac{2\rho_1 l \phi_1 \phi_2}{R_1 (\phi_1 + \phi_2)} + \frac{\phi_2 \rho_1 T_3}{\phi_1 + \phi_2} + \frac{\phi_1 \rho_1}{\phi_1 + \phi_2} \left(T_2 + \frac{\sigma_2}{R_2} + \tau_2 \right). \end{aligned} \quad (30)$$

5.4. Power Consumption. According to the power model (Section 3.2), the total power consumption of a node is

contributed by the radio transmitter, radio receiver, and computation electronics. Thus, the power consumptions of all the nodes can be computed by

$$\begin{aligned}
P_{s_1} &= \epsilon^r \rho_1 + \epsilon_{s_1 s_2}^t \rho_{1.1} + \epsilon_{s_1 s_3}^t \rho_{1.2} + p^c \\
&= \rho_1 \left[\epsilon^r + \frac{\phi_1 N_0 (2^{R_{s_1 s_2} / \eta W} - 1)}{R_{s_1 s_2} G_{s_1 s_2} (\phi_1 + \phi_2)} + \frac{\phi_2 N_0 (2^{R_{s_1 s_3} / \eta W} - 1)}{R_{s_1 s_3} G_{s_1 s_3} (\phi_1 + \phi_2)} \right] \\
&\quad + p^c, \\
P_{s_2} &= \epsilon^r (\rho_{1.1} + \rho_2) + \epsilon_{s_2 s_4}^t (\rho_{1.1}^{s_4} + \rho_2^{s_4}) + p^c \\
&= \left[\epsilon^r + \frac{N_0 (2^{R_{s_2 s_4} / \eta W} - 1)}{R_{s_2 s_4} G_{s_2 s_4}} \right] \left(\frac{\phi_1 \rho_1}{\phi_1 + \phi_2} + \rho_2 \right) + p^c, \\
P_{s_3} &= \left[\epsilon^r + \frac{N_0 (2^{R_{s_3 s_4} / \eta W} - 1)}{R_{s_3 s_4} G_{s_3 s_4}} \right] \frac{\phi_2 \rho_1}{\phi_1 + \phi_2} + p^c, \\
P_{s_4} &= \left[\epsilon^r + \frac{N_0 (2^{R_{s_4 s_0} / \eta W} - 1)}{R_{s_4 s_0} G_{s_4 s_0}} \right] (\rho_1 + \rho_2) + p^c,
\end{aligned} \tag{31}$$

where $d_{s_4 s_0}$ denotes the distance between node s_4 and the sink node s_0 . Here, we assume that the link capacity can meet the requirements of the traffic bandwidth, that is, the service rate is bigger than the sum of input data rates.

6. Performance Evaluation

To show benefits of the traffic splitting strategy and validate the network calculus-based performance analysis method, we provide numerical results and simulations under the scenario of a fresh food monitoring application. In the numerical results, the end-to-end least upper delay bounds, the least upper backlog bounds, and power consumptions are compared under two scenarios: *general routing with no traffic splitting (NOS)* and *flow-based splitting strategy (FBS)*. In the simulations, we compare the results obtained by the analytical method with the simulation results also under these two scenarios.

The numerical results are based on an application example of a real-time fresh food monitoring system deployed in a warehouse [3, 4]. As shown in Figure 8, one sink and 9 cluster heads are uniformly distributed in a 20 m × 10 m warehouse. Each cluster head connects with 5 sensor nodes. The coordinates of cluster heads and sink (s_0) are $s_0(0, 0)$, $s_1(17.2, 1.7)$, $s_2(14.1, 5.5)$, $s_3(11, 0.5)$, $s_4(14.8, -3.6)$, $s_5(8.3, 4)$, $s_6(9.1, -4.4)$, $s_7(2.5, 4.7)$, $s_8(4.2, 0.8)$, $s_9(3.3, -3.6)$. We consider the application scenario of real-time monitoring, where sensor nodes periodically generate packets when there is a request.

6.1. Numerical Results. Assume sensor nodes in cluster $\mathcal{C}_1, \mathcal{C}_2$, and \mathcal{C}_3 are requested to send packets to the sink via cluster head s_1, s_2 , and s_3 , respectively. Consequently, there are three traffic flows in the network: f_1, f_2 , and f_3 . They are characterized by arrival curves $\alpha_1 = \sum_{n_j \in \mathcal{C}_1} \alpha_1^j$, $\alpha_2 = \sum_{n_j \in \mathcal{C}_2} \alpha_2^j$, $\alpha_3 = \sum_{n_j \in \mathcal{C}_3} \alpha_3^j$, where $\alpha_i^j (i = 1, 2, 3)$ denotes

TABLE 1: Parameters.

Parameter	Notation	Value	Unit
Packet length	l	400	bits
Computation power	p^c	10	uJ/s
Path loss factor	θ	3	—
Service delay	$T_i (i = 1, \dots, 9)$	0.1	s
Data rate of f_2	ρ_2	1.2	kbps
Data rate of f_3	ρ_3	1.6	kbps
Burstiness	$\sigma_1, \sigma_2, \sigma_3$	400	bits
Channel bandwidth	W	20	kHz

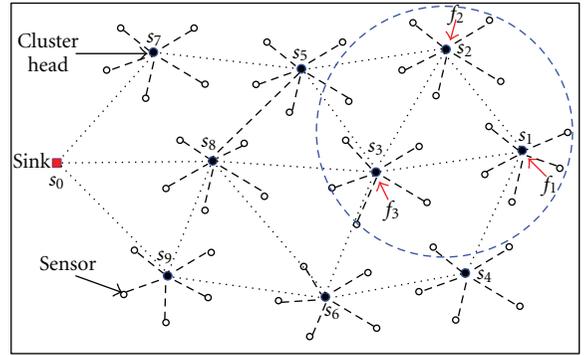


FIGURE 8: A cluster-mesh sensor network: s_0 is the sink. An event happens in the blue circle, and three traffic flows f_1, f_2, f_3 are generated.

the arrival curve of traffic generate by sensor node j in cluster \mathcal{C}_i . For periodic packets generation applications, the arrival process can be characterized by the affine arrival curve [12], that is, $\alpha_1 = \gamma_{\sigma_1, \rho_1}$, $\alpha_2 = \gamma_{\sigma_2, \rho_2}$, and $\alpha_3 = \gamma_{\sigma_3, \rho_3}$. Moreover, assume cluster head s_i provides a rate-latency service curve $\beta^{s_i} = \chi_{R_i, T_i}$. f_1 is the tagged main flow, and f_2, f_3 are the contention flows. Other parameters are listed in Table 1. We shall derive the end-to-end least upper delay bound for f_1 , the least upper backlog bounds, and power consumptions of all nodes in two scenarios: NOS and FBS (Figure 9).

From the power model in (1) and Figure 2, the energy per bit is a monotonically increasing function of the transmission rate [29] if other parameters are fixed. Thus, it is better to use low transmission power for the sake of energy efficiency. On the other hand, the delay bound is a monotonically decreasing function of the service rate. (e.g., given an arrival curve $\alpha(t) = \rho t + \sigma$ and service curve $\beta(t) = R[t - T]^+$, the delay bound is $T + \sigma/R$ when $R \geq \rho$.) Therefore, there is a tradeoff between energy consumption and delay. In order to study this tradeoff, we implement numerical experiments in two scenarios: (1) uniform service rate: the service rate of all cluster heads are the same and fixed; (2) heterogeneous service rate: in order to guarantee a limited delay and backlog, the service rate should be equal to or bigger than the data arrival rate. So we set the service rate R equal to the arrival rate ρ .

We compare the end-to-end least upper delay of f_1 , the backlog bounds, and power consumptions of all nodes in two

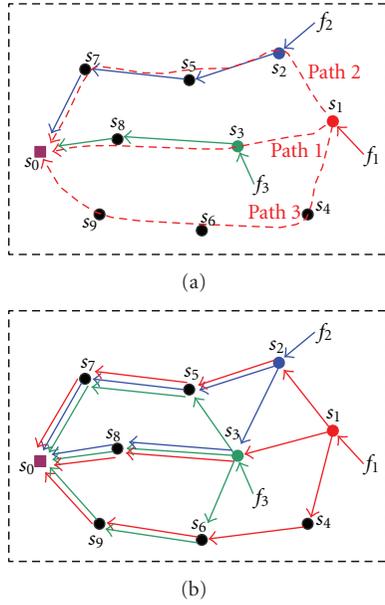


FIGURE 9: (a) General routing with no splitting (NOS): the tagged main flow f_1 chooses one of path 1, 2, or 3, and the routing paths of f_2 and f_3 are shown by the blue and green line, respectively. (b) Flow-based splitting (FBS): all three flows are split as shown by the red, blue, and green lines, respectively.

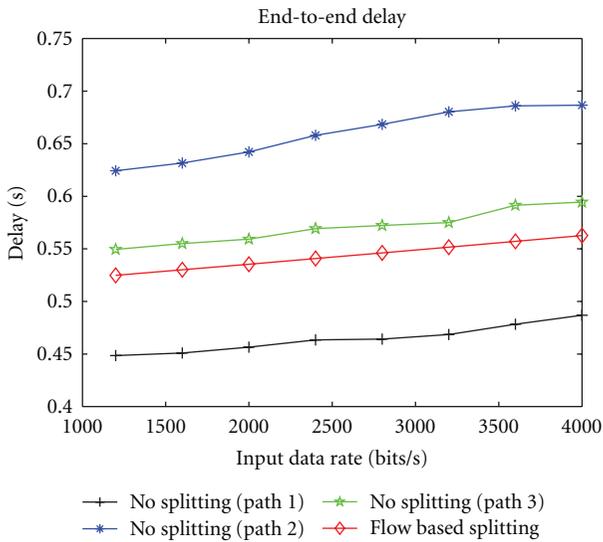


FIGURE 10: End-to-end least upper delay bound.

scenarios: NOS and FBS. In NOS, f_1 chooses one of the three paths: path 1— $\{s_1, s_3, s_8\}$, path 2— $\{s_1, s_4, s_6, s_9\}$, and path 3— $\{s_1, s_2, s_5, s_7\}$. In FBS, f_1 is evenly split into three subflows, and they are allocated on these three paths.

6.1.1. Uniform Service Rate. In the first numerical example, we choose a fixed service rate $R_i = 9.6$ kbps ($i = 1, \dots, 9$).

Figure 10 shows the comparison of the end-to-end least upper delay bounds of the tagged main flow f_1 in NOS and

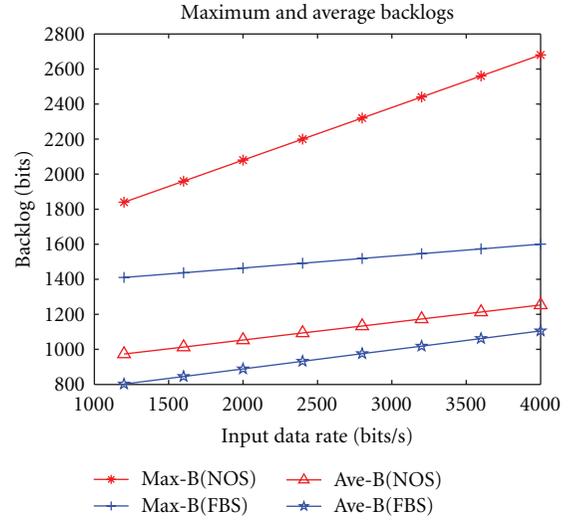


FIGURE 11: Least upper backlog bounds (in NOS: flow 1 chooses path 3).

FBS scenarios. In the NOS scenario, we compute the end-to-end delays of f_1 going through three different routing paths (Figure 9(a)). The input data rate of f_1 varies from 1.2 kbps to 4 kbps. From this figure, we can see that the end-to-end delays in all scenarios increase with the input data rates. Furthermore, on average, the delays in FBS are 23.6% and 9.4% less than those of path 2 and path 3 in NOS, respectively. And the delay in FBS is 12.4% bigger than those of path 1 in NOS. This is because path 1 is shorter than other two paths.

Figure 11 shows the least upper and average backlog bounds in the FBS and NOS scenarios with input data rates vary, where “Max-B(NOS)” means the maximum backlog in NOS which denotes the maximum value of backlogs among all nodes and “Ave-B(FBS)” means the average backlog in FBS which denotes the average value of backlogs over all nodes. From this figure, we can find that both the maximum and average backlogs in FBS are less than those in NOS. It indicates that the traffic splitting strategy can reduce backlogs. Moreover, the differences between the maximum backlogs of the two scenarios are much bigger than those of average backlogs. The average backlogs in FBS are 14.5% less than those in NOS on the average. While the maximum backlog in FBS is 23.4% less than that in NOS when the input data rate is 1.2 kbps, and the value increases to 40% when the input data rate is 4 kbps. We can also observe similar reduction in the variance of maximum backlogs (as shown in Figure 12), where the variance of backlogs in NOS is much bigger than that in FBS. It means that in NOS some nodes have very small backlogs, but some nodes have very large backlogs. Since the buffer size of a sensor node is basically determined by the value of maximum backlog, larger backlog would bring higher hardware cost. Therefore, applying the flow-based splitting strategy can bring better load balance and thus reduce overall cost.

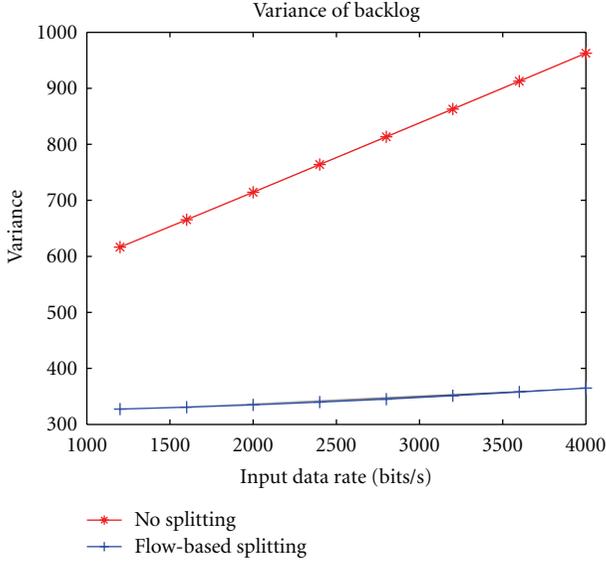


FIGURE 12: Variance of least upper backlog bounds (in NOS: flow 1 chooses path 3).

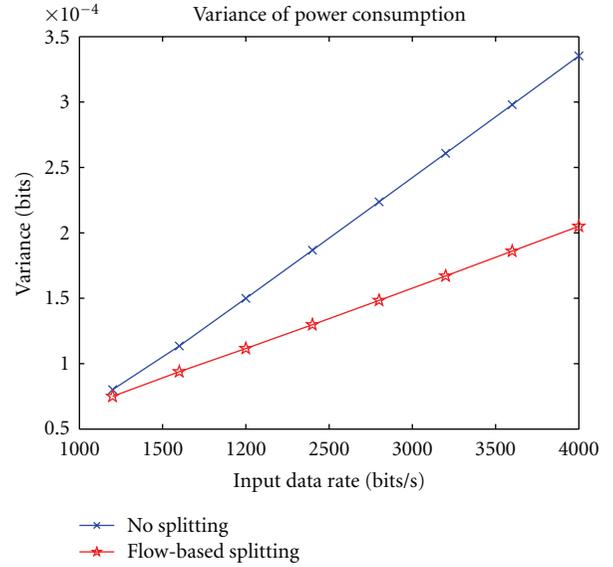


FIGURE 14: Variance of power consumption.

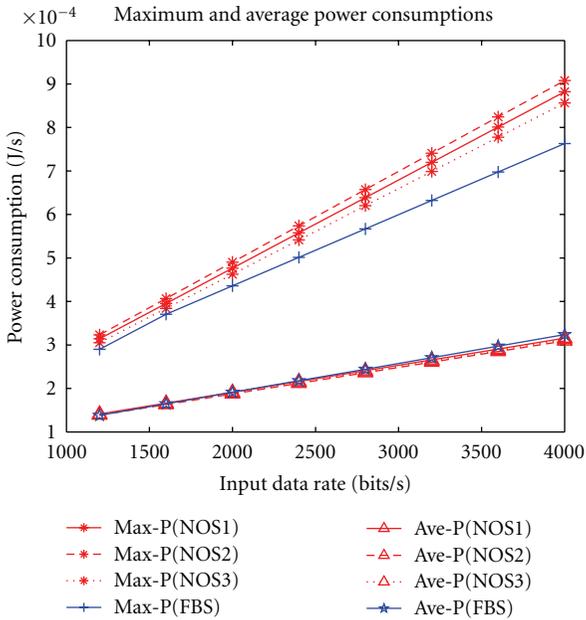


FIGURE 13: Power consumption (In NOS i : flow 1 chooses path i , where $i = 1, 2, 3$).

The maximum and average power consumptions of all nodes in NOS and FBS are shown in Figure 13, where “Max-P(NOS)” and “Ave-B(NOS),” respectively, denote the maximum and average power consumptions in the NOS scenario. First, we see from the figure that all the power consumptions increase with the input data rates. Furthermore, when the data rate increases, the average power consumptions in the NOS and FBS are almost the same. However, the maximum power consumption in NOS increases much faster than that in FBS, with the maximum differences between FBS and

NOS increasing from 0.8% to 12%. It indicates that the power consumptions of nodes are uneven in NOS. We can also see this from Figure 14 showing the variance of power consumption of all nodes. From this figure, we can find the variance in NOS increases much faster than that in FBS. Usually, the lifetime of a WSN is determined by the first node exhausting its energy. Hence, the flow-based splitting scheme can be used for balancing power consumption and consequently increasing the lifetime of the network.

6.1.2. Heterogeneous Service Rate. The data rate of f_1 varies from 1.2 to 4 kbps. The data rate of f_2 and f_3 is given in Table 1. The service rate of each node is equal to its arrival rate.

Being different from Figure 10, the end-to-end delays of FBS are basically bigger than those of NOS in this case (Figure 15). Moreover, when the input data rate increases, the end-to-end delay decreases. The reason is that the service rate increases with the input data rate. While the input burstiness is the same, so the delay would decrease.

Figure 16 shows the comparison of backlog bounds in the FBS and NOS scenarios. From this figure, we can find that the average backlogs in FBS and NOS are almost the same. While the maximum backlog in FBS is -2.4% less than that in NOS when the input data rate is 1.2 kbps, the value gradually increases to 12.8% when the input data rate gradually increases to 4 kbps. Similar to Figure 17, from Figure 17, we can see that the variance of backlogs in NOS is bigger than that in FBS. It also means that in NOS some nodes have very small backlogs, but some nodes have very large backlogs.

In the case of heterogeneous service rates, we see from the figure (Figure 18) that all the power consumptions increase with the input data rates. Furthermore, the average power consumptions in the NOS are approximately 11.4% bigger

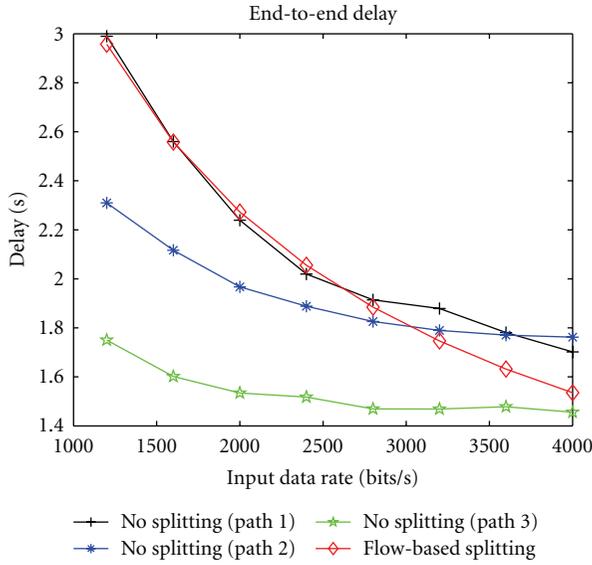


FIGURE 15: End-to-end delay.

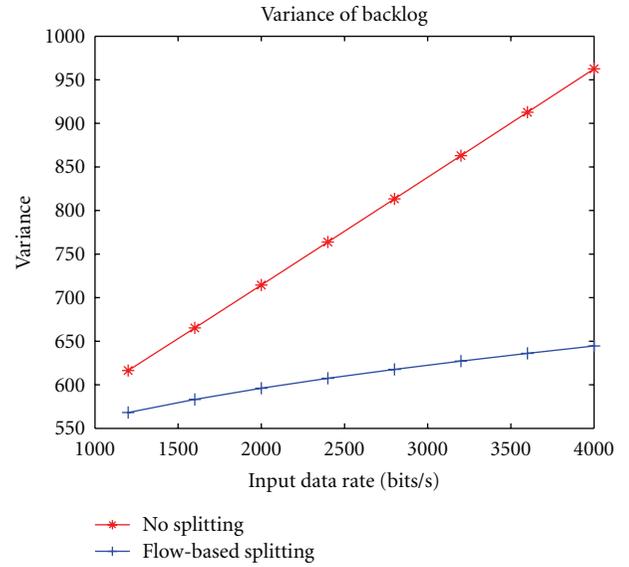


FIGURE 17: Variance of least upper backlog bounds (in NOS: flow 1 chooses path 3).

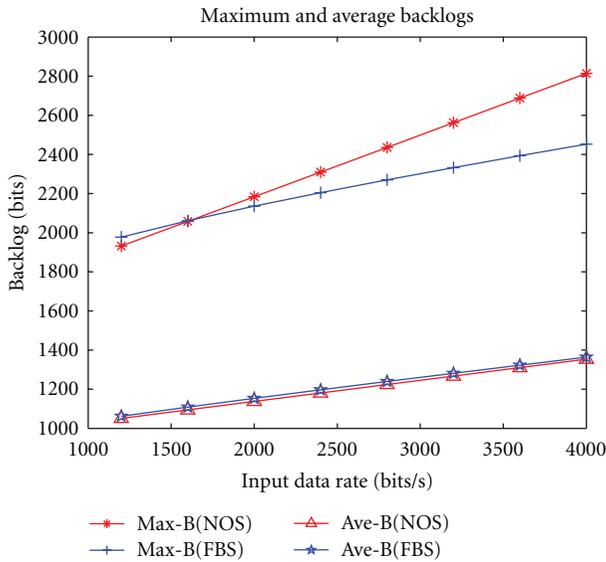


FIGURE 16: Least upper backlog bounds (in NOS: flow 1 chooses path 3).

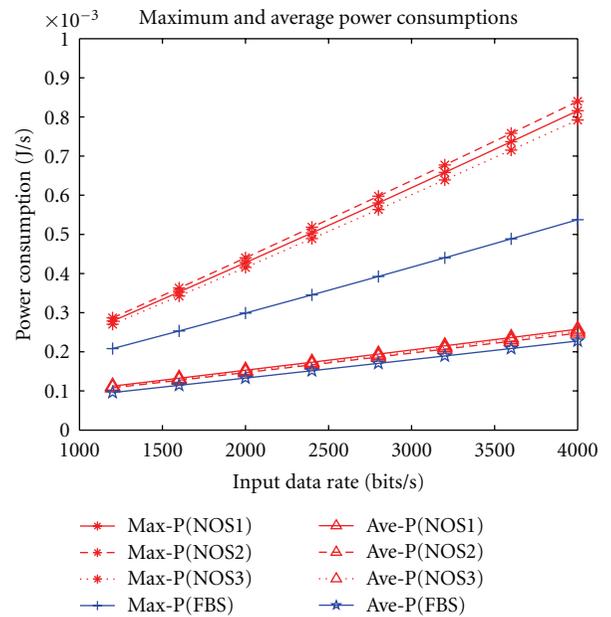


FIGURE 18: Power consumption (in NOS: flow 1 chooses path i , where $i = 1, 2, 3$).

than those in FBS. However, the maximum power consumption in NOS increases much faster than that in FBS, with the maximum differences between FBS and NOS increasing from 23.9% to 33%. The differences of maximum power consumption in this case are much bigger than those in uniform service rate case Figure 13. We can also see this from Figure 19 showing the variance of power consumption. From this figure, we can find the variance in NOS increases much faster than that in FBS.

From all those results and comparison, we can have the following conclusions: first, applying FBS strategy can balance traffic load and power consumption, so as to reduce overall system cost and increase the network lifetime. Second, there is a tradeoff between power consumption and system performance. Under uniform service rate, the end-to-end

delays of FBS are less than those of NOS in most cases, and the power consumptions of FBS are slightly less than those of NOS. While under heterogeneous service rates, the end-to-end delays of FBS are generally bigger than those of NOS, but the power consumptions of FBS are much less than those of NOS. It means that the decreasing of power consumption is obtained at the cost of increasing delay.

6.1.3. Comparison of End-to-End and Hop-by-Hop Methods. As stated in [24], there are two ways to compute the end-to-end delay bound. The first method is summing up the per-hop delay together. The main idea of the other method is

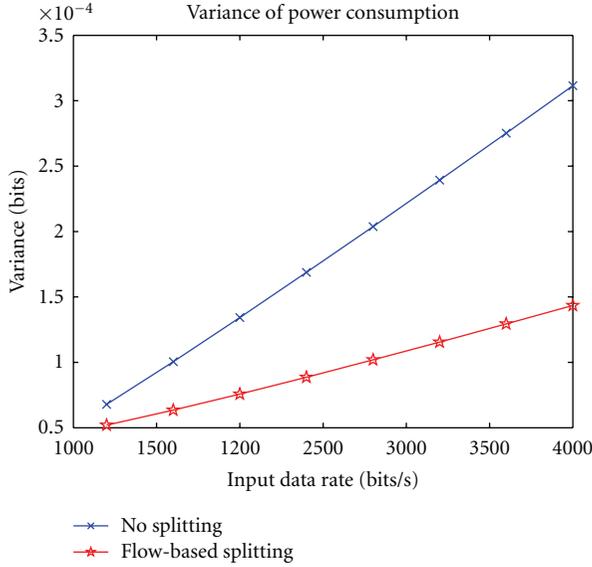


FIGURE 19: Variance of power consumption.

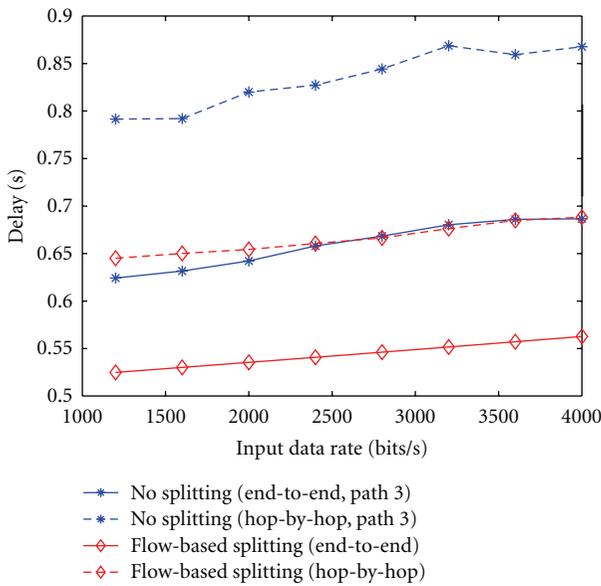


FIGURE 20: Compare the end-to-end delay computed by two methods.

to derive an equivalent service curve for a given traffic flow. And then the end-to-end delay bound is calculated using the equivalent service curve. In [24], we use the first method (hop-by-hop). While in this paper, we adopt the second method (end-to-end). Figure 20 illustrates the comparison of these two methods in the scenario of FBS and NOS. In average, the hop-by-hop delay in NOS is 26.4% bigger than that of the end-to-end delay. And, in FBS, the hop-by-hop delay is 22.5% bigger than that of the end-to-end delay. Therefore, the end-to-end method can get tighter bound than the hop-by-hop method.

6.2. Simulation Results. Since a simulation environment allows us to create a realistic sensor network behavior while

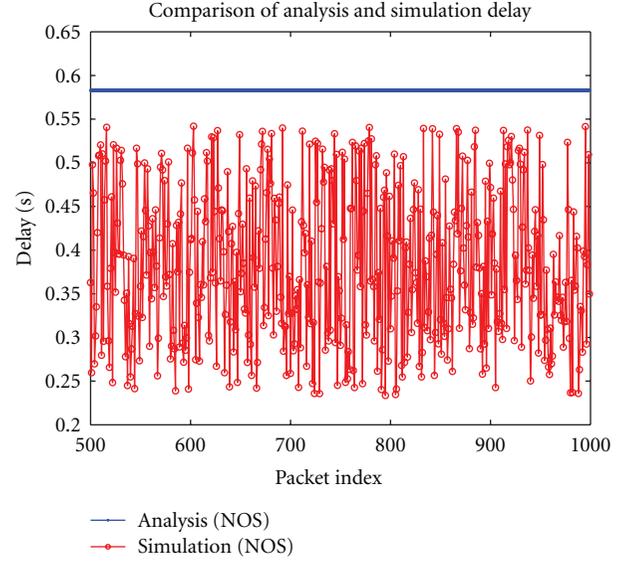


FIGURE 21: End-to-end delays in NOS.

still controllable, we conduct experiments in a simulation environment based on OMNeT++ 3.3 rather than in a field trial. We define *tightness* as the ratio of maximum simulation value divided by the analytical value.

In the simulation, we use a most common log-normal path loss model [38]. This model can provide more accurate multipath channel models than Rayleigh and Nakagami models for indoor environments [39]. The simulation is also based on the application scenario shown in Figure 8. Parameters used in simulations are the same as those in Table 1. Other parameters used are $d_0 = 1$ m, reference channel gain $G_0 = 10^{-5}$ (-50 dB), noise power $N_0 = 10^{-10}$ (-100 dB), and the channel noise is subject to a Gaussian random variable with deviation 4. We conduct 50 simulation runs. In each run, the total simulation period is 25000 cycles and the source generates one packet every cycle. The packet generation rate is based on the predefined data rate. For example, if the predefined data rate is 2.8 kbps, 7 packets are generated in every second, and the length of a cycle is 1/7 s. In order to bypass the initial nonstationary stage, the data of first 5000 cycles are omitted. In each run, the delay of every packet is recorded and the value of backlog is recorded in every cycle. Since we want to compare the analytical results with the worst-case simulation results, we select the results of runs leading to maximum delay and maximum backlog as the simulation results.

Figures 21 and 22 show the comparison of simulation results and analytical results of end-to-end delays of flow f_1 in the NOS and FBS scenarios, respectively. In NOS, the arrival rate of flow 1 is 2.8 kbps, and it selects path 3 as its routing path. In order to make the figure easy to read, we do not illustrate the delays of all packets but extract 500 values from them. From these two figures, we observe that all the simulation values are bounded by the analytical results. And the tightness in NOS and FBS are 91% and 93.2%, respectively. This indicates our analysis performs well on bounding the end-to-end data delivery delay.

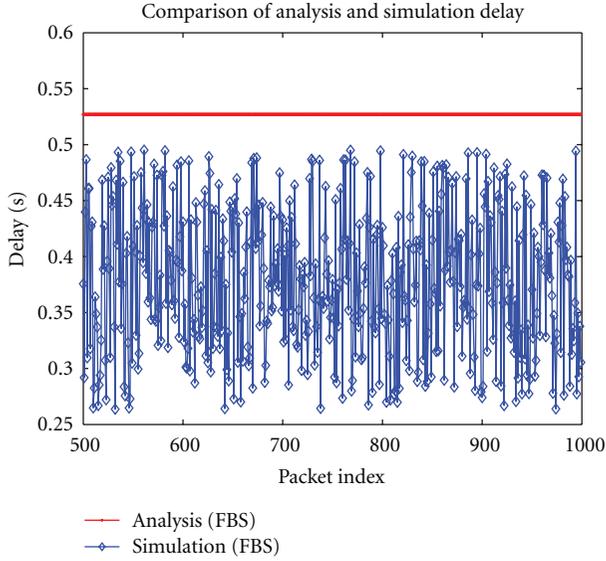


FIGURE 22: End-to-end delays in FBS.

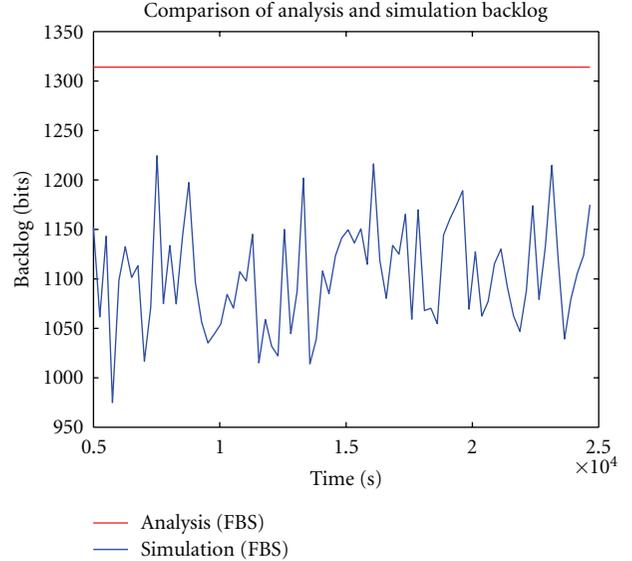


FIGURE 24: Nodes' backlogs in FBS.

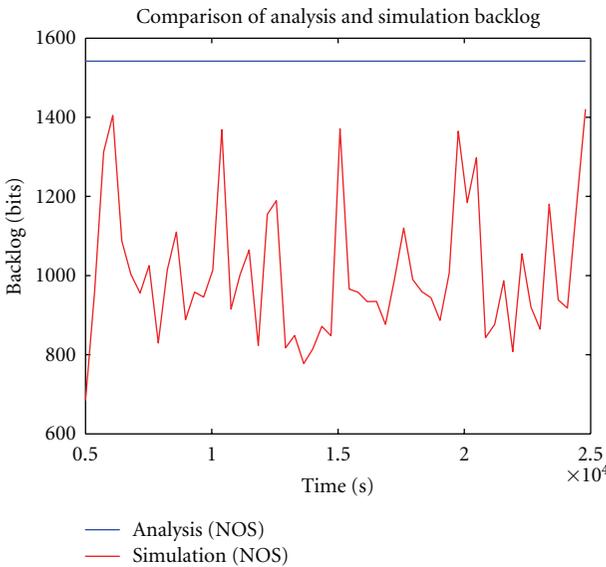


FIGURE 23: Nodes' backlogs in NOS.

For the backlog analysis, node s_8 is chosen as the observation node. Its backlogs in different time points are recorded and compared as shown in Figures 23 and 24. In NOS, the arrival rate of flow 1 is 2.8 kbps, and it selects path 3 as its routing path. We can see that all the simulation values of backlogs are within the scopes of the analytical values. Moreover, the backlogs in the FBS scenario are less than those in NOS, which also indicates that the flow-based splitting scheme can reduce the maximum backlogs by balancing traffic load over the network. Additionally, the tightness of the analytical results in NOS and FBS is 93.5% and 92.1%, respectively. In summary, the proposed analysis method is correct on deriving the backlog bound and the tightness is satisfactory.

7. Conclusions and Future Work

Dimensioning timing-critical sensor networks requires formal methods to ensure performance and cost in any conditions. In this work, we present a network-calculus-based analysis method to compute the worst-case end-to-end delay bounds for individual flows, backlog bounds, and power consumptions for individual nodes. Based on network calculus and the splitting model, we are able to compute per-flow equivalent service curve provided by the tandem of visited nodes and the input and departure arrival curves of each node. Consequently, we can derive the performance bounds for the network which applies the flow-based traffic splitting strategy. Under the assumptions of affine arrival curve and rate-latency service curves, closed-form formulas of these bounds are computed. The numerical results for the example scenario show that, by applying the splitting strategy, the end-to-end delay can be reduced in most cases, the maximum backlog can be reduced up to 40%, and the power consumption can be reduced up to 15%. Furthermore, the simulation results verify that the theoretical bounds of our analysis are valid and fairly tight.

As stated in Section 4.4, there are several directions for future work. First, we will study the problem of designing a splitting scheme, this is, how to select splitting parameters based on network state information. Another research issue is to explore the optimized design space with given buffer sizes, performance requirements, and energy constraints for specific applications.

References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [2] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *IEEE Computer*, vol. 37, no. 8, pp. 41–49, 2004.

- [3] Z. Pang, J. Chen, D. Sarmiento et al., "Mobile and wide area deployable sensor system for networked services," in *Proceedings of the 8th Annual IEEE Conference on Sensors*, Christchurch, New Zealand, 2009.
- [4] Z. Zhang, Q. Chen, T. Bergarp et al., "Wireless sensor networks for logistics and retail," in *Proceedings of the 6th International Conference on Networked Sensing Systems (INSS '09)*, Pittsburgh, Pa, USA, 2009.
- [5] S. Tai, R. Benkoczi, H. Hassanein, and S. Akl, "A performance study of splittable and unsplittable traffic allocation in wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, Istanbul, Turkey, July 2006.
- [6] A. Meddeb, "Benefits of multicast traffic split routing in packet switched networks," in *Proceedings of the IEEE International Conference on Communications (ICC '04)*, June 2004.
- [7] A. Zalesky, H. L. Vu, and M. Zukerman, "Reducing spare capacity through traffic splitting," *IEEE Communications Letters*, vol. 8, no. 9, pp. 594–596, 2004.
- [8] F. Hu, Y. Xiao, and Q. Hao, "Congestion-aware, loss-resilient bio-monitoring sensor networking for mobile health applications," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, Article ID 4909283, pp. 450–465, 2009.
- [9] R. L. Cruz, "A calculus for network delay, part I: network elements in isolation," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 114–121, 1991.
- [10] R. L. Cruz, "A calculus for network delay, part II: network analysis," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 132–141, 1991.
- [11] J. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, Springer, LNCS 2050, Berlin, Germany.
- [12] J. B. Schmitt and U. Roedig, "Sensor network calculus—a framework for worst case analysis," in *Proceedings of the IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS '05)*, vol. 3560, 2005.
- [13] A. Koubaa, M. Alves, and E. Tovar, "Modeling and worst-case dimensioning of cluster-tree wireless sensor networks," in *Proceedings of the 27th IEEE International Real-Time Systems Symposium (RTSS '06)*, Rio de Janeiro, Brazil, December 2006.
- [14] J. B. Schmitt, F. A. Zdarsky, and L. Thiele, "A comprehensive worst-case calculus for wireless sensor networks with in-network processing," in *Proceedings of the 28th IEEE International Real-Time Systems Symposium (RTSS '07)*, Tucson, Ariz, USA, December 2007.
- [15] P. Jurcik, R. Severino, A. Koubaa, M. Alves, and E. Tovar, "Real-time communications over cluster-tree sensor networks with mobile sink behaviour," in *Proceedings of the 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA '08)*, 2008.
- [16] C. S. Chang, *Performance Guarantees in Communication Networks*, Springer-Verlag, Berlin, Germany, 2000.
- [17] L. Lenzini, L. Martorini, E. Mingozzi, and G. Stea, "Tight end-to-end per-flow delay bounds in FIFO multiplexing sink-tree networks," *Performance Evaluation*, vol. 63, no. 9–10, pp. 956–987, 2006.
- [18] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energyefficient multipath routing in wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 11–25, 2001.
- [19] Y. M. Lu and V. Wong, "An energy-efficient multipath routing protocol for wireless sensor networks," in *Proceedings of the 64th IEEE Vehicular Technology Conference (VTC-06-Fall)*, Montreal, QU, Canada, September 2006.
- [20] W. Lou, "An efficient N-to-1 multipath routing protocol in wireless sensor networks," in *Proceedings of the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS '05)*, Washington, DC, USA, November 2005.
- [21] S. Zou, I. Nikolaidis, and J. Harms, "Aggregation vs. load balancing in WSNS," in *Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '07)*, Athens, Greece, September 2007.
- [22] C. Li, J. Zou, H. Xiong, and Y. Zhang, "Joint coding/routing optimization for correlated sources in wireless visual sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '09)*, Honolulu, Hawaii, USA, December 2009.
- [23] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing without routes: the backpressure collection protocol," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '10)*, April 2010.
- [24] H. She, Z. Lu, A. Jantsch, L.-R. Zheng, and D. Zhou, "Deterministic worst-case performance analysis for wireless sensor networks," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '08)*, Crete Island, Greece, August 2008.
- [25] H. She, Z. Lu, A. Jantsch, L.-R. Zheng, and D. Zhou, "Analysis of traffic splitting mechanisms for 2D mesh sensor networks," *International Journal of Software Engineering and Its Applications*, vol. 2, no. 3, pp. 25–37, 2008.
- [26] N. Baker, "Real world wireless mesh sensor network solutions," in *Proceedings of the IEE Seminar on Industrial Networking and Wireless Communications for Control*, 2006.
- [27] S. I. Lee and J. S. Lim, "Hybrid cluster mesh scheme for energy efficient wireless sensor networks," *IEICE Transactions on Communications*, vol. E91-B, no. 8, pp. 2610–2617, 2008.
- [28] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.
- [29] B. Prabhakar, E. U. Biyikoglu, and A. El Gamal, "Energy-efficient transmission over a wireless link via lazy packet scheduling," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, pp. 386–394, 2001.
- [30] A. Goldsmith and A. Nin, *Wireless Communications*, Cambridge University Press, New York, NY, USA, 2005.
- [31] S. C. Ergen and P. Varaiya, "Pedomacs: power efficient and delay aware medium access protocol for sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 7, Article ID 1637439, pp. 920–930, 2006.
- [32] A. Rowe, R. Mangharam, and R. Rajkumar, "Rt-link: A time-synchronized link protocol for energyconstrained multi-hop wireless networks," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON '06)*, vol. 2, pp. 402–411, Reston, Va, USA, 2006.
- [33] Y. Charfi, N. Wakamiya, and M. Murata, "Adaptive and reliable multi-path transmission in wireless sensor networks using forward error correction and feedback," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 3684–3689, Kowloon, Hong Kong, 2007.
- [34] P. Djukic and S. Valaee, "Reliable packet transmissions in multipath routed wireless networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 5, pp. 548–559, 2006.

- [35] S. De and C. Qiao, "On throughput and load balancing of multipath routing in wireless networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '04)*, vol. 3, pp. 1551–1556, 2004.
- [36] K. Chen, K. Nahrstedt, and N. Vaidya, "The utility of explicit rate-based flow control in mobile ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '04)*, vol. 3, pp. 1921–1926, 2004.
- [37] H. Jiao and F. Y. Li, "A service-oriented routing scheme with load balancing in wireless mesh networks," in *Proceedings of the IEEE International Symposium on Wireless Communication Systems (ISWCS '08)*, pp. 658–662, Reykjavik, Iseland, October 2008.
- [38] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, New York, NY, USA, 2002.
- [39] M. Z. Zamalloa and B. Krishnamachari, "An analysis of unreliability and asymmetry in low-power wireless links," *ACM Transactions on Sensor Networks*, vol. 3, no. 2, 2007.

Research Article

An Integrated Approach to the Design of Wireless Sensor Networks for Structural Health Monitoring

**Fabio Federici,¹ Fabio Graziosi,¹ Marco Faccio,¹ Andrea Colarieti,¹
Vincenzo Gattulli,² Marco Lepidi,² and Francesco Potenza²**

¹Center of Excellence for Research DEWS, University of L'Aquila, 67100 L'Aquila, Italy

²DISAT-CERFIS, University of L'Aquila, 67100 L'Aquila, Italy

Correspondence should be addressed to Fabio Federici, fabio.federici@univaq.it

Received 15 June 2011; Revised 10 December 2011; Accepted 18 December 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Fabio Federici et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless Sensor Networks are a promising technology for the implementation of Structural Health Monitoring systems, since they allow to increase the diffusion of measurements in the structure and to reduce the sensor deployment effort and the overall costs. In this paper, possible benefits and critical issues related with the use of Wireless Sensor Networks for structural monitoring are analysed, specifically addressing network design strategies oriented to the damage detection problem. A global cost function is defined and used for the definition of possible design methodologies. Among the various approach, the use of an integrated strategy, able to take advantage of a preliminary structural analysis is considered. Moreover, the implementation of a distributed processing is an explored strategy for an overall improvement of system performances. Benefits of this methodology are finally demonstrated through the analysis of a representative case study, the IASC-ASCE benchmark problem.

1. Introduction

Traditional Structural Health Monitoring systems usually consisted in a grid of sensors deployed along a structure, each one communicating through a wired connection with a central processing unit. Data collected by the various sensors were stored in the central unit memory and then postprocessed in order to determine structure's condition and assess a safety level. Progressive developments in sensor integration technology (e.g., the development and spread of MEMS sensors), design of low power circuits, and wireless communications have gradually allowed a wide proliferation of efficient, compact, and cheap wireless devices. This process has encouraged the adoption of wireless sensor networks, which have gradually supplanted wired systems in various application fields [1]. Recently, the use of wireless sensor networks has shown its advantages also in the field of structural health monitoring. In fact, wireless systems are usually less expensive than their wired counterparts and their installation is much simpler (e.g., just think about all the difficulties related to the setup of a wired monitoring

system in a monumental building). Nevertheless, there are many challenges related to the practical implementation of a Wireless Sensor Network for Structural Health Monitoring: wireless communication is often less reliable than the wired connection and allowed transmission distances are relatively short; on the other hand, monitored structures could be wide and present a large number of communication obstacles. Finally, sensor nodes are usually battery-powered, so there are very critical constraints in terms of energy availability. This may be particularly critical, because the effectiveness of the analysis is directly linked to the ability in performing measurements over a long period of time. The fundamental problem is then to determine appropriate strategies for network design, usually with the goal of energy consumption minimization. Moreover, there are other issues affecting network global performances and deeply related to the specific application, thus requesting a careful consideration in the design phase. Finally it is appropriate to wonder what are all possible benefits deriving from the availability of a very large number of spatially distributed, processing-capable nodes. Due to the potential enhancements obtained by the

use of Wireless Sensor Network in the development and spread of structural health monitoring systems, the presented work is focusing the attention on two main aspect:

- (i) analysis of potential benefits arising from an integrated approach to the design of wireless sensor networks for structural health monitoring, for example, an approach exploiting all the knowledge arising from a preliminary analysis of the structure;
- (ii) outlining of possible design strategies, with particular emphasis on the use of distributed processing techniques.

It will be shown how prior knowledge about structure's behaviour and expected structural response could be exploited in the design of a monitoring network. Moreover, the analysis will highlight how a distributed implementation of structural identification techniques could bring advantages in terms of network performances (e.g., by improving energy efficiency).

After a short review of the main concepts related to structural monitoring, the main advantages and disadvantages associated with the use of wireless sensor networks in Structural Health Monitoring will be detailed. Particular emphasis will be put in the analysis of possible processing techniques, specifically reviewing some of the attempts to define distributed processing schemes. An integrated approach to the design will be then analysed and motivated by the definition of a global network cost function and the definition of a possible integrated design strategy. Finally, the usefulness of this approach will be highlighted addressing a well-known case study, the IASC-ASCE benchmark problem.

2. Structural Health Monitoring

Structural Health Monitoring is defined as the process of implementing strategies for damage detection for the infrastructures of mechanical, civil, and aerospace engineering [2]. In this context, structural damage is defined as any general change in the geometry or in material's characteristics of the infrastructure under examination. The key feature of the monitoring action must be the characterization of the system in its normal service condition (i.e., without interrupting normal system's functionality), possibly over a long-time interval.

Restricting the field of observation, the attention is here focused on civil infrastructures monitoring (i.e., monitoring of standard and monumental buildings, bridges, lifelines, etc.). In their normal service condition, these structures will always be subject to environmental action (e.g., wind action) and to human-activity-related forces. As mentioned, monitoring process involves the observation of a structure over an extended period of time, periodically acquiring measures. Data processing allows the synthesis of structural indicators that are somehow representative of structural damage and the successive diagnostic judgement usually derives from a statistical analysis of the obtained parameters.

It is usual to consider the following monitoring systems classification (Rytter [3]):

- (i) level I system, that is, a system able to detect structural damage when it actually occurs;
- (ii) level II system, that is, a system able to detect damage and locate its position within the structure;
- (iii) level III system, that is, a system able to detect the damage and give an estimate of its location and intensity;
- (iv) level IV system, that is, a system able to estimate the location and extent of the damage and to use this data to determine the state of the overall structure, and therefore its security level.

At the highest levels of Rytter scale corresponds a major detail in structure's condition assessment and, usually, an increase in processing complexity. Processing techniques should therefore be chosen according to the specific requirements of the particular application. In facts, the ultimate goal is not necessarily a complete characterization of the state of the structure and for certain applications a lower level of diagnosis detail may be sufficient. The level of detail required can also differ according to the possible changes in the operational scenario.

Anyhow, regardless of the specific objective of the monitoring action, we can say that the key point is to determine appropriate techniques to extract damage sensitive parameters from measured data. These damage indicators, or features, should

- (i) significantly vary when there is evidence of structural damage for a level I system;
- (ii) present a significant variation related to the location of the damage for a level II system;
- (iii) depend on the extent of damage under a specific law for a level III system.

Finally, for a level IV system, it must exist an appropriate method of statistical analysis that allows the extraction from the assessed indicators of specific information concerning the building, its security level, and the uncertainty of these estimates.

We will now briefly review the main techniques for the determination of the features. As mentioned above, our analysis will focus on a clearly defined scope: building's characterization and monitoring. Furthermore we will consider principally the vibrational analysis approach. It is assumed that the sensors placed along the structure are accelerometers and the measured response is expressed exclusively in terms of acceleration.

2.1. The Traditional Approach: Modal Analysis. The traditional approach for structural health monitoring using measurements of structural response is based on modal identification techniques. The modal parameters, namely the natural frequencies, mode shapes, and damping ratios, are the damage-sensitive features.

Besides the wide use of this technique in the field of civil engineering, the main reason for its wide adoption lies basically in recent years spreading of the so-called output-only analysis. Traditionally, the modal identification requires the measurement of structural response (output) to a known and controlled system of forces (input). Consequently, it was necessary to use appropriate equipment capable of applying such forces on buildings. This approach had relevant implications in terms of system cost, footprint (given the large size of the equipment), and difficulty of running measurements over an extended period of time. Instead, in the case of output-only analysis, we usually try to determine the modal parameters analysing the response to environmental actions (such as wind or traffic), which remain unmeasured as they can be well approximated as white Gaussian noise. This procedure has two obvious advantages: there is no need to use any controlled solicitation equipment and it is possible to characterize the structure under its normal operating conditions (thus fully implementing the basic monitoring principle). For this reason, we often refer to this technique as “operational modal analysis.”

Over the years, various techniques for modal structural identification have been developed and recently compared on experimental data [4]. These include

- (i) time domain techniques, such as ITD (Ibrahim Time Domain [5]), ERA (Eigensystem Realization Algorithm [6]), Next (Natural Excitation Technique), and SSI (Stochastic Subspace Identification) [7];
- (ii) frequency domain techniques, such as BSF (Basic Frequency Domain, also known as the Peak Picking [8]) and FDD (Frequency Domain Decomposition [9], subsequently improved as EFDD, Enhanced Frequency Domain Decomposition [10]);
- (iii) time-frequency methods, such as those based on analysis of wavelet transforms [11], Cohen’s class transforms [12], and recently EMD (Empirical Mode Decomposition) [13].

The use of modal parameters allows to extract information about structural damage, in different manner as

- (i) by analysing changes of natural frequencies in order to detect the occurrence of damage [14];
- (ii) by analysing changes in mode shapes in order to locate the position of the damage [15];
- (iii) by evaluating changes in flexibility or stiffness or using statistical analysis techniques. Yan et al. [16] analysed all these various methods in details.

As mentioned (and illustrated in [16]) damage detection techniques based on modal parameters represent the “traditional” approach to the problem. This approach could appear convenient when a complete characterization of the structure is strictly required, but, in general, it may present some critical issues. For example, the modal-based identification procedures are almost never completely automatic and often not universal, but rather specifically related to the facility under examination. Moreover these techniques are not

efficient in tracing the microdamages that could early occur in the process of damage. Finally, as clearly illustrated by Peeters and Roeck [17], modal parameters variations can be caused both by the presence of damage, whether as a result of changes in environmental conditions.

2.2. Innovative Approaches. The mentioned disadvantages are at the base of recent years increasing interest for all the possible damage detection techniques that are not based on modal analysis. Among the most common approaches can be cited the so called “modern approaches” [16]:

- (i) wavelet analysis: it can be shown [18] that the spectrum obtained from wavelet is somewhat directly representative of damage. One of the major advantages of these techniques is that they allow the analysis of nonstationary signals;
- (ii) the use of genetic algorithms: it has been demonstrated [19] that the use of genetic algorithms allows detection of structural damage;
- (iii) the use of neural networks: this technique has been used several times in the literature, especially for the classification of other features extracted from measurements [20].

Transmissibility analysis [21] also appears particularly interesting, as it has been shown how the extracted feature can provide good results regardless of the type of applied loads.

Moreover, the transmissibility analysis [22–24] or wavelet analysis [25] can also be used for the determination of modal parameters. The adoption of direct feature extraction should not therefore be seen as a limitation, since the extracted parameters are not only indicative of structural damage, but can also enable the global characterization of the structure.

3. Wireless Sensor Networks for Structural Health Monitoring

As stated in Section 2, the primary aim of structural health monitoring is detection, location, and quantification of structural damage according to different strategies: from the analysis of modal parameters to the direct extraction of damage-related features. Wireless sensor networks [26] emerged as a suitable solution for the implementation of monitoring strategies [27]. In fact, the deployment of a wireless system is almost always easier than the installation of its wired counterpart, even in the case of buildings subject to operational constraints and limitations (i.e., buildings of historical or artistic relevance). The use of economic wireless sensors usually allows high-density coverage of a structure with relatively low costs [28]. In addition to the characterization of damage, the system could also be designed with additional goals: for instance the ability to collect a significant amount of data from a very high number of points could allow the development of efficient model updating strategies. Finally, local processing could allow the

introduction of early warning strategies (fundamental in the case of structure at risk of experiencing sudden severe stress, as in the case of earthquakes).

3.1. Network Basic Requirements. The design of a structural monitoring oriented wireless sensor network must necessarily start from the analysis of application requirements. One of the the main problems is that the various analysis techniques usually require the collection of a significant volume of data from sensor nodes, and these data have to be combined in order to extract significant synthetic parameters. This modality heavily impacts on node's energy consumption. Moreover, other analysis requirements can directly influence the design of the network.

In what follows, these problems will be clarified and the main problems in the design of a structural health motoring oriented wireless sensor network will be detailed.

3.1.1. Coverage Requirements. Sensor node's placement is one of the key points to consider in the design of a wireless network [29]. In fact, besides ensuring proper operation and full coverage of the area of interest, the optimal location may have a significant impact on power consumption, propagation delay, and data throughput.

Deployment critical issues clearly emerge where application's proper functioning is highly dependent on node's position. In these cases, application's constraints may be much more important than usual and must be necessarily satisfied. In the case of structural monitoring this problem appears evident: the location of a specific sensor node should be chosen according to the significance of the response at that particular point in relation to structural analysis objectives. Often in the design process, usual WSN metrics must be taken into account only as secondary specifications, with their satisfaction subordinated to analysis needs. So, the design of a structural monitoring oriented wireless sensor networks must firstly consider application's demands [30, 31]. For example, as proposed by Guratzsch [32], an optimized design of sensor nodes deployment could be the one that maximizes the probability of detecting damage. Starting from a finite element model of the structure, appropriate simulations of the possible damage patterns will then make it possible to map an optimal distribution of the nodes along the structure. When the final goal is structural identification, it is possible to adopt a similar criterion, selecting the points that provide the most significant measurements in relation to the parameters to be determined (e.g., mode shapes in the case of modal analysis).

There are several types of algorithms for automatic determination of the optimal nodes position in relation to damage sensitivity. For example, Udwardia and Garba [33] or Lim [34] presented algorithms for optimal placement in relation to the identification and control of the structure. Hiramoto et al. [35] proposed the use of Riccati equation for optimal positioning in relation to vibration control. Tongpadungrod et al. [36] have instead used the principal component analysis to determine performance indicators.

3.1.2. Network Connectivity. Deployment algorithms oriented to structural analysis and damage detection often provide solutions which are nonoptimized, or even not feasible with regard to communication. For example, especially in the case of large structures, there is no guarantee that the positions determined by the algorithms mentioned above will also ensure proper communication between nodes. Although some methods able to take into account both structural analysis and communication requirements have been explored, these approaches are often not feasible. Other strategies, like the use of redundant nodes, must be considered. For example, it is possible to include sensing nodes that are not strictly necessary for the analysis, but arranged in such positions as to ensure proper coverage of the structure. A similar strategy is rather one that involves the use of relay nodes. A technique for relay nodes optimal placement in a wireless sensor network has been presented by Lloyd and Xue [37]. Finally, it is important to consider the case of clustered networks applied to structural monitoring because the partition of the network may be conditioned by the specific application (e.g., a cluster can be associated with a sub-structure). Again the best strategy is to ensure first the satisfaction of application requirements (significant deployment and sectioning in relation to damage analysis and detection) and then optimize other metrics, for example, with an efficient cluster head design and their optimal placement along the structure.

3.1.3. Energy Consumption and Network Lifetime. The reduction of energy consumption is one critical aspect in the design of wireless sensor networks. The sensor nodes are usually battery powered and the amount of available energy is extremely limited. In contrast, wireless sensor networks are often dedicated to tasks such as monitoring of physical phenomena and therefore require a life time as long as possible (it is not uncommon to have to deal with applications that require an operating time of months or even years).

The wireless nodes need often to be installed in hostile or hard-to-reach environments. So, it is not possible to provide an ordinary maintenance operation for battery replacement. Attempts to recover energy through, for example, the use of solar cells have proven to be in general critical. Consequently, one of design's main goals must be energy consumption's minimization.

Each sensor node has different energy consumption characteristics, but, as outlined by Anastasi et al. [38], we can assume as true the following conditions.

- (i) In their operative state, radio communication devices dissipate an amount of energy much higher with respect to processing devices. As shown by Pottie and Kaiser [39] transmitting one bit requires the same amount of energy needed to run hundreds of instructions. Consumption could be instead significantly lowered putting the radio device in a sleep state.
- (ii) The impact of the sensing blocks on energy consumption varies according to the specific application and it must be managed differently in each particular case.

In the context of structural health monitoring the problem of limited available energy is critical for various reasons. For example, the amount of data to be transmitted can be significant, especially when the monitoring action has to be performed on long-time intervals; moreover, sensor nodes are often placed in not easily accessible locations, so a battery replacement results impractical and expensive.

For monitoring applications, the power consumptions optimization strategies can follow essentially two traditional approaches [38]:

- (i) duty-cycling approaches where duty cycle is the time in which radio communication device is active. Since radio is responsible for major energy consumption, strategies to minimize the communication activity should be pursued, leaving the radio chip in the sleep state for most of the time. The strategies usually employed are two and complementary: the control of the topology, that is, the use of a limited number of nodes in a redundant topology and the power control used to obtain the state of sleep in the inactive nodes;
- (ii) data-driven approaches that consists essentially in data-reduction techniques chosen according to the specific application. The basic principle is still taking advantage of the fact that a local data processing carried out by single nodes would result in low consumption than that required to transmit the same data. The used techniques are usually in-network processing, or a data aggregation performed before transmission, data compression, or the use of techniques of data prediction.

3.2. Additional Possible Requirements. There are also other requirements, mostly aimed at ensuring network robustness, or particularly critical only for specific applications; in what follows we briefly review some of them.

3.2.1. Synchronization Protocols. Different nodes may have different trigger moments and therefore, the initial timestamp of acquired data acquired can be different for different sensors. Moreover, all the possible errors due to different clock misalignment and drifts must be considered [40].

Synchronization may constitute a critical issue for some of the reviewed structural monitoring techniques, while it can be less critical for others. For example, it has been proven how a synchronization lack could compromise the correct estimate of mode shapes, while it is less critical for natural frequency and damping ratios determination [41].

The problem of synchronization between nodes in a wireless sensor network has been object of extensive research over the years and various synchronization techniques have been developed. Basically, the majority of techniques try to use communication between neighbouring nodes to align differences between local clocks. For example, RBS (Reference Broadcast Synchronization), FTSP (Flooding Time Synchronization Protocol) and TPSN (Timing-Sync Protocol for Sensor Networks) are widely used synchronization techniques. The latter has proved particularly suitable in the context of structural health monitoring [42].

3.2.2. Fault Tolerance and Robustness. Usually, structural monitoring networks should provide good fault tolerance, at least in relation to the following possible fault causes:

- (i) running out of batteries, because the observation interval should be the longest possible;
- (ii) possible sensing units malfunction: if data is not properly detected it is possible to obtain a false positive damage detection;
- (iii) damage due to a violent stress, because events such as earthquakes are extremely significant in the life of a building, and it is therefore necessary to ensure proper functioning of the system so that we can study the behavior in those particular stress conditions.

Some of the classical strategies [43] to achieve good fault tolerance in wireless networks are not immediately applicable in the case of structural health monitoring: for example, a possible method of fault prevention is to design network topology in order to ensure maximum connectivity. As mentioned, however, in the monitoring networks, sensors location is generally conditioned by the constraints of application. In this case it is necessary to jointly consider application and fault tolerance requirements, or insert appropriate redundant nodes. Fault detection can help prevent possible false positives in damage detection applications. In this regard, Chan et al. [44] have detailed some preliminary studies on possible strategies to improve the reliability of a system for damage detection in relation to possible fault, putting in place appropriate detection strategies.

3.2.3. Real-Time Constraints. Actually, real-time processing is rarely considered as a requirement of current sensor networks for structural health monitoring, but is rather one of the most interesting research topics in this area. It bears mentioning it for two main reasons:

- (i) it is an essential requirement in Early Warning oriented applications;
- (ii) distributed processing architecture allows in some cases a local damage diagnosis. This fact, and the progressive advancements in processing hardware (e.g., the spread of application specific processor for wireless sensor networks) makes obtaining real-time damage detection a very next feature goal.

3.3. Distributed Architecture for Structural Health Monitoring. The problems highlighted clearly emphasize the need to determine an optimal strategy for the design of sensor networks oriented monitoring, that is, strategies able to take into account the principal highlighted issues. As mentioned, one of the problems lies in the conflicting demands between two aspects: analysis requirements and efficiency of the network. In facts, while a detailed analysis of the structure would require the collection of all data from various sensors, optimization of consumption would require to minimize the time intervals in which nodes operate and above all to minimize data communication.

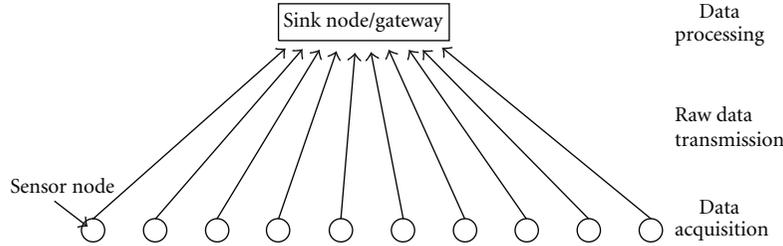


FIGURE 1: Traditional centralized architecture for data aggregation-based structural monitoring.

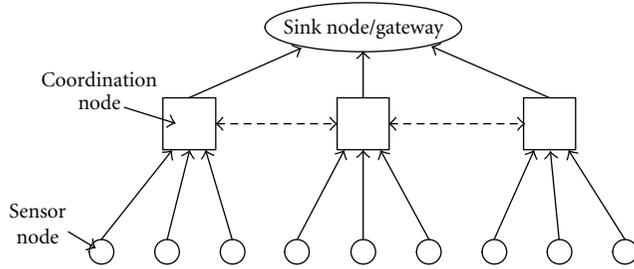


FIGURE 2: Distributed computing strategy for damage detection.

In recent years the attempt to overcome this problem has led to an increasing interest towards the use of distributed processing schemes. This approach can be critical for an efficient design, so it is appropriate to summarize the main available techniques.

Early structural health monitoring oriented wireless sensor networks suffered from this contradiction in a clear manner. In fact, the architectures used were centralized, with a single node to act as a sink for the various leaf nodes, instead devoted to the measurement of structural response in the significant points (Figure 1).

In this scenario, each sensor nodes must communicate with the sink, each time transferring the data acquired in a time window of interest. Since the size of the acquired data can be very large, this architecture is highly inefficient in terms of consumption, since it requires a massive use of radio communication unit. This architecture appears biased towards the needs of the application and takes little account instead of the optimization of communications. This is reflected in many of the first implementations in the context of wireless structural health monitoring and is basically tied to the fact that wireless systems were used as a simple cable replacement.

The need for a change in paradigm has emerged quickly, leading to a rethinking of the architecture used. As we saw in Section 2, the basic objective of monitoring is to characterize the structural damage, or more generally the state of the structure. It is therefore natural to consider the possibility of implementing an in-network processing or distributed processing within the network. For example, we might look for a convenient way to ensure that the single node may arrive to an estimate of the modal parameters before the transmission phase, thus processing only local acquired data.

This approach, which basically responds to the data-driven philosophy, appears efficient provided that

$$E_{\text{raw}}^p + E_{\text{info}}^{\text{tx}} \leq E_{\text{raw}}^{\text{tx}}, \quad (1)$$

where E_{raw}^p it is the energy needed for raw data processing, $E_{\text{info}}^{\text{tx}}$ it is the energy needed for local processing of raw data and $E_{\text{raw}}^{\text{tx}}$ it is the energy needed for the transmission of processed data. As mentioned, this condition can be assumed as true for common off-the-shelf sensor nodes. Network optimization could then pass via a review of the techniques mentioned in Section 2 in optic of a distributed processing across the network.

One of the first significant contributions in this direction is the one presented by Gao et al. with the introduction of Distributed Computing Strategy [45]. Assuming that the damage is inherently a local phenomenon and given the high density of sensors required to detect the damage, they have proposed to overcome the problem of low efficiency of centralized architectures by introducing the clustered architecture as shown in Figure 2.

The analytical method used is one of those classified as “traditional”: a combination of distributed modal analysis (performed using the NExT/ERA techniques) and use of the flexibility matrix to estimate the damage. The obtained results have already shown the potentiality of the distributed approach.

In the context of modal analysis it is interesting to examine the strategy of distributed computing developed by Zimmerman et al. [46] (Figure 3). The technique of analysis considered in that case was the simple peak picking: assuming that the structure is excited by a white Gaussian noise, the Fourier transform of the single node measured response corresponds to the frequency response of the structure for that node. Since response’s peaks are located in correspondence to the natural frequencies, local analysis of the calculated spectral profile can lead to a fairly accurate estimate of natural frequencies themselves. This estimate can be improved by analysing the global information obtained by all the nodes.

The technique is simple but it indeed allows the determination of natural frequencies and can also be used for damage detection. In fact, as mentioned in Section 2, changes in natural frequencies may indicate the occurrence of structural damage.

The same article also illustrates a distributed version of the Frequency Domain Decomposition. In that case it

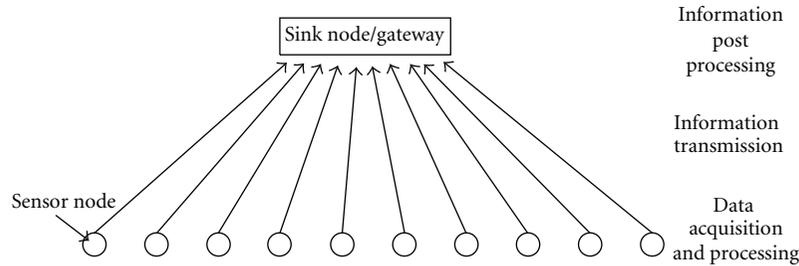


FIGURE 3: Decentralized architecture for peak picking-based modal analysis.

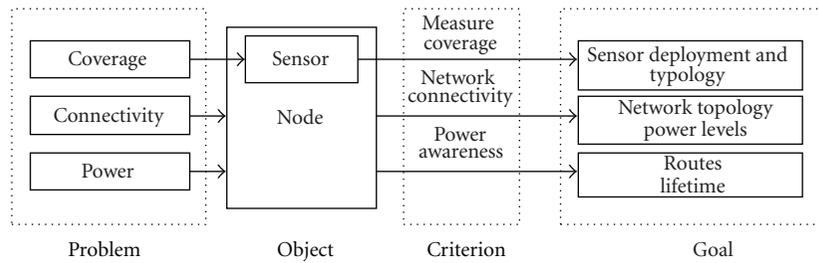


FIGURE 4: Different approaches in the design of a Wireless Sensor Network.

was shown that by using the measured data from a pair of sensors, it is possible to obtain an accurate estimate of the mode shapes. This approach, based on the calculation of “local” mode shapes, has been further developed by Sim [47] introducing the concept of overlapping clusters of sensors for the determination of the local mode shapes using the NExT/ERA technique (the method is still valid with techniques such as Frequency Domain Decomposition and Stochastic Subspace Identification).

This technique, however, do not reduce the amount of data to be transmitted, which may even increase compared to the previous case. Zimmermann and Lynch [48] have introduced a market-based approach to network topology formation. The goal was basically overcoming the previous case problems and obtaining of a more flexible system.

Modal analysis has however other several disadvantages, starting from the difficulties associated with full automation. It may therefore be convenient to use “modern” techniques, not based on modal analysis. For example, Gun et al. [49] presented a distributed approach for damage detection based on wavelet analysis. Worden et al. [50] have presented a technique based on the use of the transmissibility for the detection of structural damage in plate structures. Toivola and Hollménn [51], have instead presented a statistical technique for the selection of features. Canales et al. [52] have proposed an approach based on local transmissibility analysis for the detection of the damage.

The latter case is of particular interest because the analysis technique is local and each sensor node can independently analyse the result and make local decisions about the level of danger. In a distributed scenario, each node may be suitably programmed to behave differently depending on of the different position or react differently to an event.

4. An Integrated Approach to the Design of a Wireless Sensor Network for Structural Health Monitoring

The results obtained from the preliminary study of a structure and the eventually realized model can be exploited in the design of a wireless sensor network for structural health monitoring. This section will detail and justify this “integrated” approach.

The problem of wireless sensor networks design has been addressed in several studies. Depending on the problem’s formulation, three different approaches can be distinguished.

Coverage Problem. The main goal is to determine sensor deployment location, given some coverage quality constraint.

Connectivity Problem. The main goal is to determine a network topology and node’s transmission power level, given some network connectivity constraint.

Power Awareness Problem. The main goal is usually to determine transmission routes, given some network lifetime constraint.

In general, design choices will usually arise from compromises between these various needs; for example, as already mentioned, network lifetime depends on the energy stored in node’s batteries. The reduction of transmission power level can certainly increase lifetime, but at the cost of a lesser connectivity.

Coverage, connectivity, and lifetime are the main problems to be addressed in the design of a sensor network (Figure 4). A good coverage requires a suitable number of

sensors, sufficient to detect the response of a given set of targets. A good connectivity requires that each nodes must be able to communicate with its nearest sink node, given a certain transmission power level. Network lifetime should be the maximum possible, given node's power consumption and transmission power.

It should be noted that there are several possible definitions for the lifetime of a network. For example, a feasible indicator is the total number of operations that the network is able to complete since at least one node stops to operate. Alternatively, it is possible to consider the relationship between node's total amount of available energy and its average energy consumption per time unit.

Furthermore, since radio transmission predominantly affects node's power consumption, an alternative indicator is the volume of data that the node can transmit before battery exhaustion.

A good coverage is essential for a Structural Health Monitoring Sensor Network: the main goal of monitoring action is in fact to measure a sufficient pool of information, in order to capture the structural signature.

It will be also important to ensure a good network lifetime: as pointed out, sensor nodes could be often installed in remote places, so replacing batteries would be a difficult and expensive task.

Other possible requirements, like fault tolerance and measurement synchronization, are here considered as second-order specifications.

It could be useful to evaluate the quality of a wireless sensor network oriented to structural health monitoring defining the following cost function:

$$C = C_{cov} + C_{con} + C_{lt}, \quad (2)$$

with C_{cov} coverage cost, C_{con} connectivity cost, and C_{lt} lifetime cost.

Both coverage cost and connectivity cost depend on the number of nodes of the network: the first by the number of sensor nodes N_s , and the second by the total number of nodes N (supposing the presence of N_r relay nodes, it will be $N = N_s + N_r$). Sink nodes are excluded from the calculation, mainly because they usually have not stringent constraints in terms of power consumption.

Regarding the coverage problem, the sensor deployment can be driven by an expert interpretation of the dominant behaviour characterizing the structural response, often supported by numerical simulations. In particular, useful suggestions about the number, type, and placement of sensors can follow from geometric, static, and dynamic analysis, with major attention to assess a minimal sufficient number of sensors, still able to extract the information of interest. For structural health monitoring purposes, key considerations are specifically oriented to limit the points to be monitored. They regard, for instance

- (i) the nature and position of the external constraints, connecting some structural elements to the ground, which may completely or partially fix the degrees of freedom of the constrained nodes;

- (ii) the stiffness distribution, which may suggest reasonable assumption about the extension and bending flexibility of mono- and bidimensional elements;
- (iii) the mass distribution, which may enable an efficient reduction in the number of dynamically active degrees of freedom, based on the dominant inertial forces.

On the other hand, different considerations tend to augment the measurement points, in order to capture critical aspects of the structural response, worth to be monitored. Preliminary linear and nonlinear analyses may reveal

- (i) the presence of internal resonances between the natural frequencies, which may activate relevant phenomena of energy transfer between the resonant modes;
- (ii) the presence of nodes (fixed points) in the shape of the dominant natural modes;
- (iii) the localization and hybridization of modes, which may concentrate high dynamic accelerations and stresses in one or more structural regions;
- (iv) the development of high-amplitude oscillations in slender elements.

A general point to be considered is that a meaningful representation of the structural response is composed of both global (e.g., the natural frequencies) and local information (e.g., the components of natural modes). Global information is naturally redundant, since it can be usually extracted by several sensors. Local information may require instead a certain amount of redundancy to compensate the eventual failure of single node. Therefore, it is always convenient to plan a small increment of sensors with respect to the minimal sufficient set. In principle, these additional sensors should be considered separately in terms of costs, since, being redundant, they can be required to satisfy reduced performance levels.

The coverage cost C_{cov} can be expressed as follows:

$$C_{cov} = \gamma_n \sum_{i=1}^{N_n} c_{ni} + \gamma_m \sum_{i=1}^{N_m} c_{mi}, \quad (3)$$

where N_n is the minimum and sufficient number of sensor nodes needed for the specific structural analysis and N_m is the number of redundant sensor nodes needed to obtain a good coverage robustness (clearly, $N_s = N_n + N_m$). Two cost indicators, c_{ni} and c_{mi} , represent the cost per sensor node, respectively, for core sensor nodes and redundant sensor nodes. This indicators depend on sensor node typology and performances (and therefore also with the economic cost), as well as the installation costs. Finally, the γ coefficients, here and in the following, represent weighting factors ($\gamma \leq 1$).

Network connectivity will depend on two main aspects:

- (i) the ability of each node to communicate with at least its nearest sink node: this ability depends on network architecture, routing strategy, and transmission power levels;

- (ii) the possibility of satisfying connectivity requirements including a certain number of relay nodes.

A virtuous integration among the needs of the structural health monitoring process and the improvement of the wireless network performance can be based on the actual possibility to harmonize the hierarchical network organization with the hierarchical structural scheme of the monitored object. In this respect, a smart cluster design strategy of the network nodes can be based on the recognition of different substructures, characterized by a limited number of significant degrees of freedom. Typical exemplifying cases are represented by clusters of all the nodes placed on structural element groups affected by internal rigidity constrains (as the horizontal planes of pseudo three-dimensional frame models in concrete structures, or the macroelements in three-dimensional models of masonry structures).

The connectivity cost function c_{con} can be expressed in the following way:

$$c_{con} = \gamma_c \sum_{i=1}^{N_c} c_{ci} + \gamma_r \sum_{i=1}^{N_r} c_{ri}. \quad (4)$$

In the previous relation N_c represents the total number of links, while c_{ci} is the architectural cost (i.e., the cost related to the existence of a connection between two nodes, given a certain power transmission). As outlined, this cost strongly depends on chosen network topologies and routing strategies. Without loss of generality, this cost can be modelled as an increasing function of the distance between two sensor nodes d_i :

$$c_{ci} = \alpha_{ci} d_i^\beta. \quad (5)$$

The fixed β coefficient models possible radio channel attenuation factor. The global cost is an increasing function of the number of links: this is strictly true for traditional networks, not for innovative approaches based on the use of network coding techniques [53]. If architectural solutions do not allow the desired network connectivity a choice could be to increase transmission power level, for example, for the problematic (isolated) nodes. In general, this is not a particularly good choice, since it greatly impacts power consumption. As mentioned, an alternative is instead the insertion of N_r relay nodes, each at the cost of c_{ri} .

Network lifetime, as mentioned earlier, is primarily related to node's energy consumption. A feasible indicator is the following:

$$c_{lt} = \gamma_0 \sum_{i=1}^{N_n+N_m} c_{0i} + \gamma_v \sum_{i=1}^{N_n+N_m+N_r} c_{vi}. \quad (6)$$

The relation takes into account two main factors:

- (i) the fixed cost c_{0i} related to acquisition and data processing. If nodes can be put in a sleep state, it is possible to consider the following formulation:

$$c_{0i} = c_{0i}^{run} + c_{0i}^{sleep} \quad (7)$$

$$\text{with } c_{0i}^{sleep} \ll c_{0i}^{run}.$$

- (ii) a cost related to the volume of data to be transmitted:

$$c_{vi} = \alpha_i^{meas} \alpha_i^{proc} c_{ui}, \quad (8)$$

where c_{vi} represents the transmission cost per data volume unit. As mentioned, energy consumption is an increasing function of transmitted data volume. One possible choice in order to increase network lifetime could be the adequate selection of measured data. For example, a sampling frequency reduction or the selection of only one axis of a 3D accelerometer can reduce the total amount of data. The α_i^{meas} factor represents this reduction. Similarly, the α_i^{proc} factor represents the possible dimension reduction resulting from a distributed processing.

Data volume, and thus the c_{ui} cost, can be directly related to the distance d_i : in fact, considering, for example, the *first-order radio model*:

$$\begin{aligned} R_i &= \epsilon_{elec} v, \\ T_i &= \epsilon_{elec} v + \epsilon_{amp} d_i^2 v, \end{aligned} \quad (9)$$

where R_i and T_i are, respectively, transmission and receive power, while d_i is still the distance between two generic sensor nodes and v_i the volume of data to be transmitted. The factors ϵ_{elec} and ϵ_{amp} are, respectively, the energy needed for transmission or reception of a single bit and the energy consumption per transmitted bit of transmission amplifier.

Analogously to the connectivity-related cost c_{ci} , even this cost depends on link's distance, but while the former indicator defines the cost related to a certain level of connectivity (e.g., a certain network topology), the second defines the cost related to information transfer on the available network.

The definition of the C cost function allows to outline a possible design strategy specifically calibrated on the requirements of a structural health monitoring application. As outlined, the three components of the defined global cost are not independent. The weighting factors can thus be chosen so as to give an importance to one of the specific aspects, in fact orienting the design action. As mentioned, coverage is the most important requirement in the design of a structural health monitoring system. So, a possible strategy consists in assigning to γ_n the maximum value and then proceeds with the following steps:

- (1) definition of the minimum number of sensor and their positions along the structure by means of a preliminary structural analysis and modelling action;
- (2) definition of the number of redundant sensor N_m , starting again from structural considerations;
- (3) definition of the network architecture and routing strategy, given a certain transmission power level and coverage requirements;
- (4) connectivity verification and possible insertion of relay nodes;
- (5) data selection and processing distribution in order to satisfy network lifetime requirements.

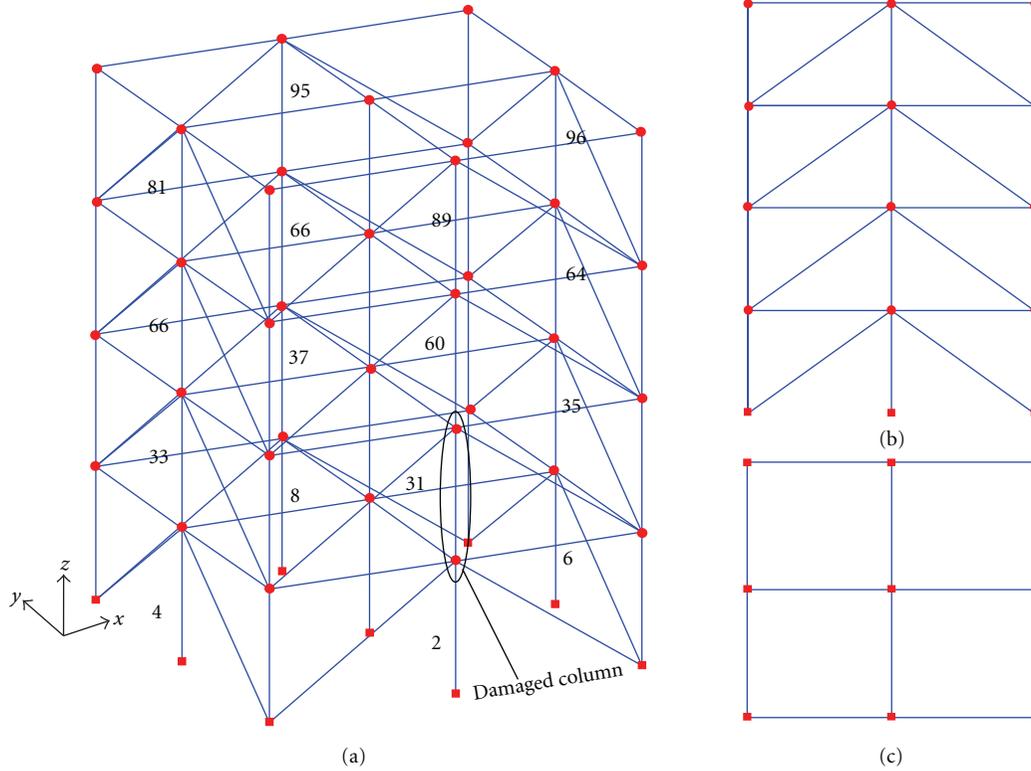


FIGURE 5: IASC-ASCE benchmark structure model.

As already mentioned, network lifetime is influenced not only by the amount of transmitted data but also by the transmitting power. For this reason, if the lifetime requirement is not satisfied by means of step 5, it will be necessary to reconsider the architectural choices. One possible strategy is the reduction of transmission power level, possibly adding additional relays to maintain connectivity.

To conclude, it must be remarked that the proposed strategy allows a certain flexibility, since it can be adapted to satisfy different purposes of the monitoring process, as well as to exalt the network potential. In fact, the design problem solution can be uniquely determined at different steps. In general, structural considerations (steps 1, 2) may leave the sensor deployment open to alternative solutions, different in their individual topology, but substantially equivalent in terms of measure coverage. Therefore, the connectivity and/or power awareness requirements (steps 4, 5) become determinant as discriminating criteria. This common situation is highly stressed in advanced networks, equipped with distributed processing capacities, oriented to peculiar structural purposes within the structural health monitoring field (experimental modal analysis, damage identification, model updating, and early warning). In this case, a proper tuning of the cost weights may transfer the strategy focus from the information measure (dominant coverage problem) to the information processing (dominant power problem). The exemplifying case study presented in the following section illustrates how the power cost ends up to discriminate between two network topologies with similar coverage costs,

when a three-dimensional frame structure is monitored for damage identification purposes.

5. Integrated Design Example for a 3D Frame Structure

The IASC-ASCE benchmark problem is here used to demonstrate the main features of the proposed methodology in a well-known case study.

The problem, formulated in 1999 under the umbrella of IASC (*International Association for Structural Control*) and ASCE (*American Society of Civil Engineering*) deals with a four-story, two-bay by two-bay steel frame (see Figure 5).

The numerical model, here used for demonstration purposes, has been constructed through the Finite Element Method (FEM) using 132 beam elements. Classical assumptions, such as the rigid m-plane behaviour of each floor is used to derive a reduced-order model, following a methodology clearly explained in [4, 54].

In the benchmark problem, displacements along the x - y axes as well as rotations with respect to the vertical axis in each floor were constrained to be dependent on the central mode. Rotations with respect to the x and y axes were allowed at all nodes. Consequently, the application of the reduction procedure gives the equation:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{M}\mathbf{R}\ddot{\mathbf{u}}_g, \quad (10)$$

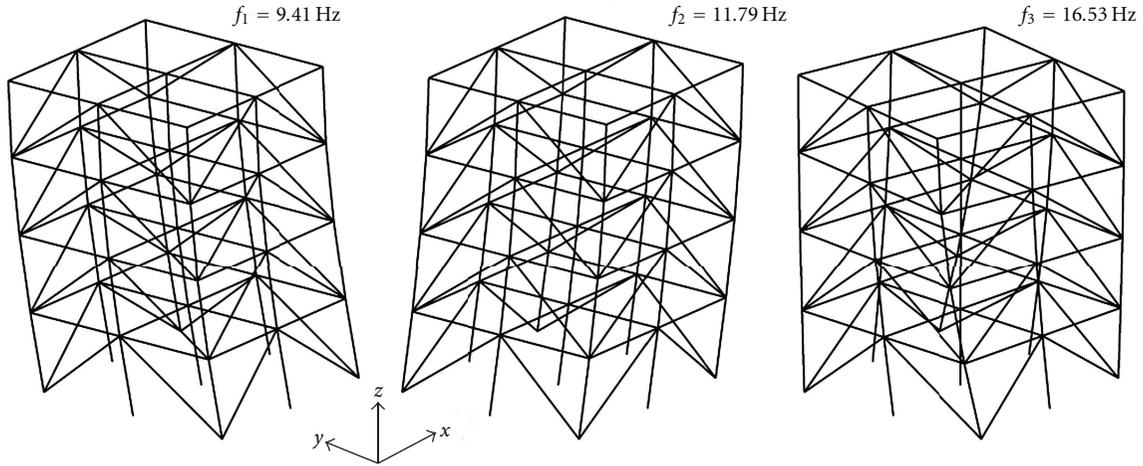


FIGURE 6: First 3 mode shapes of the frame structure.

where \mathbf{u} is the displacement vector describing the active 88 DOFs and \mathbf{M} and \mathbf{K} are the mass and stiffness matrices; \mathbf{R} is the rigid matrix which allows to simulate the ambient disturbance responses as generated by the $\ddot{\mathbf{u}}_g$ ground motion acceleration vector containing the two horizontal components in the x and y directions as well as a rotation with respect to the z axis. Finally, the damping matrix \mathbf{C} is obtained such that a damping ratio of 1% is introduced in the six lowest modes, while the three excitation components are three bandwidth limited, statistically independent, and normally distributed random inputs. Figure 6 shows the first 3 mode shapes of the frame structure.

The simulation of the structural response consists of 6 min inputs from which data are obtained at a frequency of 200 Hz. A damage pattern can be also simulated as stiffness reduction of a given element.

The plain design strategy for the wireless network tends to realize the necessary and sufficient coverage, maintaining a minimal measurement redundancy. According to the same structural considerations which justify the pseudo three-dimensional model of the frame, three independent components of motion (plus 1 redundant, dependent on the others) should be measured for each floor plane ($N_n = 12$, $N_m = 4$, considering monodimensional accelerometers for simplicity). It can be supposed that connectivity reasons, related to the transmitter features, the frame dimensions and the environmental conditions, require the addition of one relay node for each floor ($N_r = 4$). The consequent topology of the network is referred to as WSN_1 in the following and is illustrated in Figure 7(a).

Each floor is equipped with three mono-dimensional sensors (or equivalently a three-dimensional sensor with triple cost) in the central node, and two eccentric nodes (the redundant sensor and the redundant relay, with reduced functions and reduced cost). Since this topology is able to wholly characterize the global structural response of the frame, it is expected that the WSN_1 may well-perform for most of the structural monitoring purposes. For instance, all the modal components would be captured during

experimental modal analyses, typically finalized to modal identification or model updating.

Nonetheless, the benchmark problem refers to a particular structural monitoring purpose, relying on the damage identification in columns. Moreover, an advanced wireless network might feature a distributed processing potential to be exploited. The key question to be addressed is whether and how the proposed design strategy can evaluate the cost-based convenience of a different network topology, taking into account the possibilities of the process distribution to smart nodes.

To give general consistency on the network design, a reasonable assumption is that an efficient damage identification technique can be based on the comparison between the experimental response measured at the column upper and lower node (u_i and u_j , resp.). These considerations allow to detail most of the processing costs in the WSN_1 . Aiming to distribute the processing effort, two smart central nodes P_i (measuring x_i, y_i, θ_i) and P_j (measuring x_j, y_j, θ_j), placed in the i th and j th adjacent floors, respectively, are requested to locally perform the following information processing:

- (i) P_i : reconstruct, the experimental response u_i from x_i, y_i, θ_i (operation O_i),
- (ii) P_j : reconstruct, the experimental response u_j from x_j, y_j, θ_j (operation O_j),
- (iii) P_i or P_j : compare the experimental response u_i and u_j (operation O_{ij}), that is, each pair of smart nodes is charged of three local processing operations for each column. Assigning conventionally the elementary costs to the related power expense (see Table 1), and following the algebra of the previous section, the final cost of the WSN_1 can be evaluated as $C_{WSN_1} = 5800$). It is worth noting that this evaluation follows from a conventional assignment of the individual cost weights, explicitly oriented to reward the minimal sufficient network topology, characterized by a small number of sensors (high γ_n values).

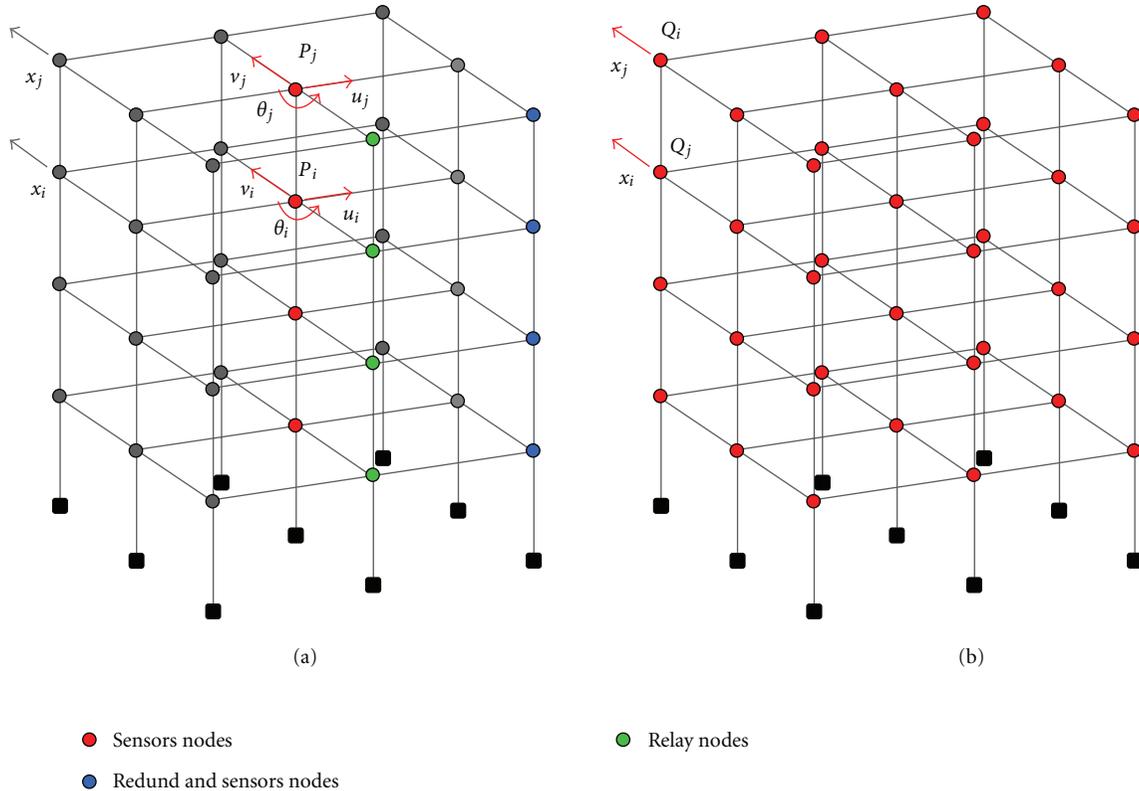


FIGURE 7: Wireless Sensor Networks for column damage identification in the IASC-ASCE benchmark frame: (a) WSN₁, (b) WSN₂.

TABLE 1: Values of the weighting factors, unitary costs, and coefficients in the cost function.

Weighting factors	WSN ₁		Weighting factors	WSN ₂	
	Unitary costs	Coefficients		Unitary costs	Coefficients
$\gamma_n = 1.0$	$c_{ni} = 100$	—	$\gamma_n = 0.5$	$c_{ni} = 100$	—
$\gamma_m = 1.0$	$c_{mi} = 100$	—	$\gamma_m = 0.5$	$c_{mi} = 100$	—
$\gamma_c = 0.7$	$c_{ci} = 40$	$\alpha_{ci} = 1.0$	$\gamma_c = 0.6$	$c_{ci} = 40$	$\alpha_{ci} = 1.0$
$\gamma_r = 0.7$	$c_{ri} = 70$	—	$\gamma_r = 0.7$	$c_{ri} = 70$	—
$\gamma_0 = 0.5$	$c_{0i} = 60$	—	$\gamma_0 = 0.5$	$c_{0i} = 60$	—
$\gamma_v = 0.9$	$c_{ui} = 80$	$\alpha_i^{\text{meas}} \alpha_i^{\text{proc}} = 1.0$	$\gamma_v = 0.9$	$c_{ui} = 80$	$\alpha_i^{\text{meas}} \alpha_i^{\text{proc}} = 1/3$

Combining both structural considerations and damage identification purposes, a second network WSN₂ can be considered (Figure 7(b)), adopting a diffuse node deployment, finalized to have a sensor in each beam-column joint ($N_n = 36$). Due to the high sensor density, additional sensors or relay nodes are supposed unnecessary ($N_n = N_r = 0$). Under the previous hypotheses about the processing distribution on smart nodes, each couple of eccentric nodes Q_i (measuring u_i) and Q_j (measuring u_j), is requested to locally perform a single information processing:

- (i) Q_i or Q_j : compares the experimental response u_i and u_j (operation O_{ij})

that is, each pair of smart nodes is charged of one local processing operation only. The sensor deployment and link scheme need to be accompanied by a proper routing strategy, which is supposed to support this particular behaviour.

As before, the final cost of the WSN₂ can be evaluated. With respect to WSN₁, the WSN₂ definitely consists of a larger number of nodes, each one performing a lower number of operations. The consequent major difference in terms of cost is not a significant reduction of the data volumes to be transmitted (which reduces of only one fourth), but the actual possibility of a more efficient routing strategy in the transmission. To quantify this advantage, an $\alpha_i^{\text{meas}} \alpha_i^{\text{proc}}$ coefficient less than unit has to be applied, inversely proportional to the square of the data volume per single transmission.

Therefore, it is easy to verify that WSN₂ is a better solution (i.e., is less expensive than WSN₁)

- (i) if the designer can somehow reduce the individual node cost, or equivalently wants to strongly penalize the minimal network topology (low γ_n values);

- (ii) if the designer must consider high processing costs, large amount of data volumes, or equivalently wants to award an efficient processing distribution (high γ_v values).

In the particular case study, according to first approach, the WSN₂ has been adopted for the damage identification purposes, as it becomes less expensive than the WSN₁ ($C_{\text{WSN}_2} = 3384$) when the sensor cost weight is reduced to one half.

To illustrate the WSN₂ effectiveness with respect to a particular damage identification procedure a possible response-based implementation is proposed. Given the sensors s_i and s_j , the transmissibility is defined as

$$T_{ij}(f) = \frac{u_i(f)}{u_j(f)}, \quad (11)$$

where u_i, u_j are Fourier transform of the displacement response at i and j nodes under the ground motion input due to ambient disturbances. Transmissibility magnitude among the selected nodes can be approximated by the following relation:

$$\left| T_{ij}(f) \right| = \sqrt{\frac{G_{ii}(f)}{G_{jj}(f)}}, \quad (12)$$

where G_{ii} and G_{jj} are the estimated power spectral density of the structural responses measured at nodes i and j . In the possible implementations, an estimate of power spectral density can be pursued using Welch's method [55], which is basically an improved version of the periodogram-based power spectral density estimation. In Welch's method, measured signal is divided in overlapping segments and each segment is then windowed. The average of the periodogram calculated from each segment is a good estimate of the needed power spectral density. The use of Welch's method can significantly reduce the contribution of noise and is therefore widely used in embedded applications.

Supposing that the goal is to detect columns damages, an intuitive choice may be to consider, for all the frame storeys, the transmissibility calculated between pairs of sensors positioned at both ends of each frontage middle columns. As reported, all the columns of the structure are oriented to have higher bending flexibility in the x direction. Without introducing normalization, we can then assume frontages as single reference substructures (i.e., single scenarios for parameter comparison and classification).

Assuming that the structure is not initially damaged, the previous algorithm would lead to the calculation of baseline transmissibility. Damage detection should intuitively be based on the analysis of variations with respect to the baseline. In the successive measurement cycles a node will have then to calculate an updated transmissibility, searching for evident variations.

The ASCE tool is here used for the generation of a $T = 800$ s sequence of acceleration at all the $N = 16$ sensor nodes deployed along the structure. Obtained structural data, we used the Welch method to estimate the power

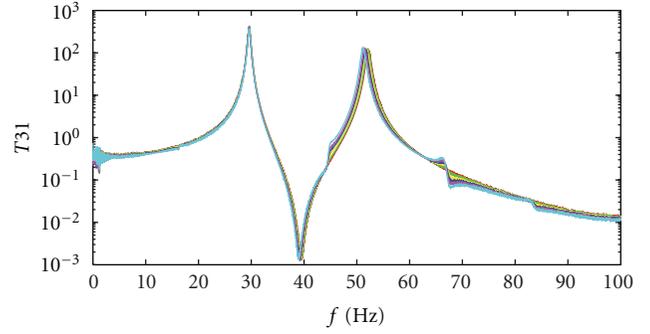


FIGURE 8: Transmissibility evaluated on column 31 for different damage intensity.

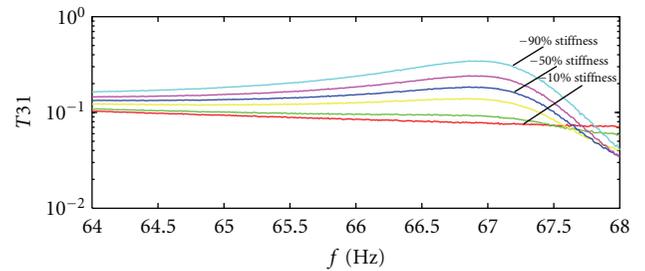


FIGURE 9: Detail of transmissibility T31 for different damage intensity.

spectral density of each response, using $N_w = 8$ segment and an Hamming window. We assumed the direct ratio between the power spectral density at two nodes as estimate of the relative transmissibility.

Figures 8 and 9 show the variations of the transmissibility calculated at different frequencies between nodes at the ends of element 31 in which the damage is concentrated and simulated as a loss of element's stiffness. The different transmissibility functions are drawn for different damage intensity, with the decrease in stiffness varying from 10% to 90% of the normal conditions.

The effects of damage on the transmissibility among other nodes have been investigated and as example the functions related to columns 37 and 60 are reported in Figures 10 and 11.

The use of local information through the evaluation of the transmissibility functions can be demonstrated still efficient with respect to a more global analysis if an efficient damage indicator is introduced. This indicator could in fact become a synthetic damage-sensitive feature. There are various ways to extract a synthetic feature. For example Johnson and Adams [56] used the following indicator:

$$DI = \frac{\left| \sum_f 1 - \left(T_{ij}^u(f) / T_{ij}^d(f) \right) \right|}{n}. \quad (13)$$

TABLE 2: DI versus EDI for $x - z$ response at various columns.

Stiffness Reduction	Columns						Columns					
	31	37	60	66	89	95	31	37	60	66	89	95
10%	0.011	0.013	0.012	0.009	0.003	0.008	5.633	5.744	5.160	5.382	5.192	4.922
30%	0.044	0.054	0.051	0.024	0.001	0.041	7.331	7.415	6.591	6.949	6.446	6.111
50%	0.075	0.092	0.087	0.029	0.008	0.076	7.895	7.919	7.167	7.603	6.906	6.543
70%	0.112	0.142	0.128	0.030	0.025	0.123	8.263	8.191	7.603	8.106	7.199	6.840
90%	0.182	0.236	0.196	0.026	0.065	0.212	8.425	8.311	8.060	8.659	7.431	7.081
Index	DI						EDI					

TABLE 3: DI versus EDI for $y - z$ response at various columns.

Stiffness Reduction	Columns						Columns					
	35	33	64	62	93	91	35	33	64	62	93	91
10%	0.691	0.066	0.028	0.001	0.006	0.003	4.792	4.696	4.018	4.368	3.997	4.125
30%	0.204	0.142	0.455	0.055	0.024	0.006	6.236	6.017	4.898	5.507	4.901	5.124
50%	0.232	0.129	5.810	0.116	0.042	0.005	6.901	6.601	5.254	6.017	5.272	5.564
70%	0.253	0.104	0.269	0.186	0.063	0.001	7.443	7.054	5.516	6.442	5.568	5.934
90%	0.271	0.074	0.066	0.292	0.097	0.013	8.073	7.556	5.809	6.927	5.921	6.365
Index	DI						EDI					

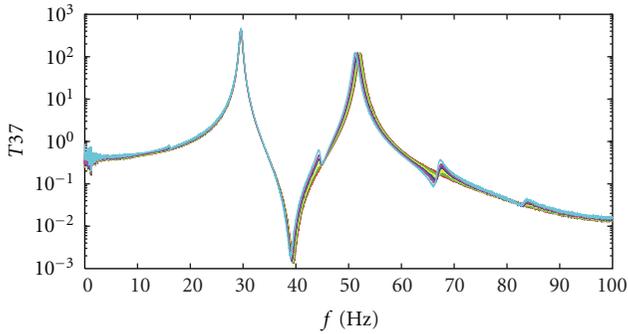


FIGURE 10: Transmissibility evaluated on column 37 for different damage intensity.

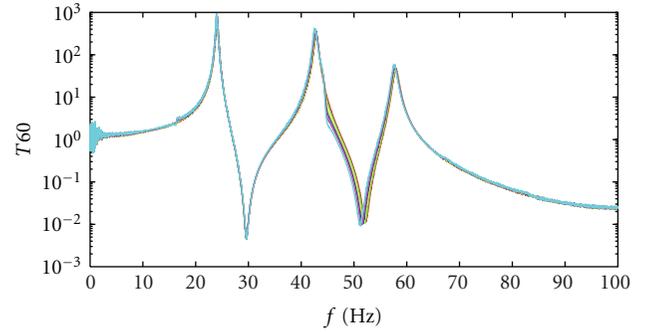


FIGURE 11: Transmissibility evaluated on column 60 for different damage intensity.

Here, a possible improvement of the technique is proposed introducing the damage feature index DF as

$$DF(f) = \left| \log \left(1 - \frac{T_{ij}^u(f)}{T_{ij}^d(f)} \right) \right|. \quad (14)$$

Then, the enhanced damage index (EDI)

$$EDI = \frac{10}{\frac{1}{n} \sum_f DF(f)} \quad (15)$$

is proposed as a synthetic indicator, easily implementable in wireless sensor networks.

In Tables 2 and 3 we have reported all the values assumed by the EDI indicator, evaluated for all the selected node couples, when a damage is introduced again on the element

31 and progressively increased (T_i indicates transmissibility for column i). As mentioned, it is convenient to separately evaluate different sides.

It should be noted that the EDI indicator is effectively sensitive to damage, provided that we consider different columns characteristics. In fact, considering the single frontages, we have a significant variation of the indicator when damage is introduced. Moreover, the highest value is relative to sensor couple positioned at the extremities of the damaged element, for all the examined cases and the indicator value increases with the the damage intensity augmentation.

We have obtained similar results applying damages in other positions, concluding that in the analysis of singular frontages the proposed procedure can effectively locally diagnose the occurrence of a damage.

6. Conclusion and Future Developments

In this paper many of the critical aspects related with structural health monitoring oriented wireless sensor networks design have been reviewed. The analyses have allowed the definition of a cost function useful for the assessment of a deterministic criterion to compare different network solutions.

The cost function can be adapted to alternately reward or penalize the network coverage, connectivity, and power expense, depending either on expert designer choices or particular project constrains. According to cost-saving purposes, it has been evidenced how an original, dedicated algorithm for the network design can actually take advantage of a number of preliminary structural characterizations of the object to be monitored, implementing a so-called integrated design strategy. It has been shown how an integrated design could be able to simultaneously satisfy different target balances among application, communication, and energy requirements and could represent an interesting starting point towards an overall efficiency and sustainability improvement.

A practical design example has shown how the proposed design methodology can be applied to a real monitoring problem. A damage detection strategy has been outlined and successfully applied to the exemplifying case of a benchmark frame structure, introducing among other things a novel damage indicator, the enhanced damage index. It has been shown how this indicator can be useful in columns damage detection.

Future developments will be oriented to further investigate the presented technique, implementing the transmissibility method in a real scenario and using a reliable statistical analysis tool to verify its validity. The comparison of theoretical results and real world data-derived results will allow to properly validate the method here presented.

References

- [1] C. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.
- [2] H. Sohn, C. R. Farrar, F. M. Hemez et al., *A Review of Structural Health Monitoring Literature: 1996–2001*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2004.
- [3] A. Rytter, *Vibration based inspection of civil engineering structures*, Ph.D.dissertation, Aalborg University, Department of Building Technology and Structural Engineering, 1993.
- [4] E. Antonacci, A. De Stefano, V. Gattulli, M. Lepidi, and E. Matta, "Comparative study of vibration-based parametric identification techniques for a three-dimensional frame structure," *Journal of Structural Control and Health Monitoring*. In press.
- [5] P. Mohanty and D. J. Rixen, "A modified Ibrahim time domain algorithm for operational modal analysis including harmonic excitation," *Journal of Sound and Vibration*, vol. 275, no. 1-2, pp. 375–390, 2004.
- [6] J. Juang and R. Pappa, "An eigensystem realization algorithm for modal parameter identification and model reduction," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 620–627, 1985.
- [7] E. Reynders and G. D. Roeck, "Reference-based combined deterministic-stochastic subspace identification for experimental and operational modal analysis," *Mechanical Systems and Signal Processing*, vol. 22, no. 3, pp. 617–637, 2008.
- [8] J. Bendat and A. Piersol, *Random Data; Analysis and Measurement Procedures*, Wiley-Interscience, New York, NY, USA, 1971.
- [9] R. Brincker, L. Zhang, and P. Andersen, "Modal identification of output-only systems using frequency domain decomposition," *Smart Materials and Structures*, vol. 10, no. 3, pp. 441–445, 2001.
- [10] R. Brincker, C. Ventura, and P. Andersen, "Damping estimation by frequency domain decomposition," in *Proceedings of the 19th International Modal Analysis Conference (IMAC '01)*, 2001.
- [11] S. Erlicher and P. Argoul, "Modal identification of linear non-proportionally damped systems by wavelet transform," *Mechanical Systems and Signal Processing*, vol. 21, no. 3, pp. 1386–1421, 2007.
- [12] A. Roshan-Ghias, M. Shamsollahi, M. Mobed, and M. Behzad, "Estimation of modal parameters using bilinear joint time-frequency distributions," *Mechanical Systems and Signal Processing*, vol. 21, no. 5, pp. 2125–2136, 2007.
- [13] J. Chen, Y. L. Xu, and R. C. Zhang, "Modal parameter identification of tsing Ma suspension bridge under typhoon victor: EMD-HT method," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 92, no. 10, pp. 805–827, 2004.
- [14] A. Pandey, M. Biswas, and M. Samman, "Damage detection from changes in curvature mode shapes," *Journal of Sound and Vibration*, vol. 145, no. 2, pp. 321–332, 1991.
- [15] J. Kim, Y. S. Ryu, H. M. Cho, and N. Stubbs, "Damage identification in beam-type structures: frequency-based method vs mode-shape-based method," *Engineering Structures*, vol. 25, no. 1, pp. 57–67, 2003.
- [16] Y. Yan, L. Cheng, Z. Wu, and L. Yam, "Development in vibration-based structural damage detection technique," *Mechanical Systems and Signal Processing*, vol. 21, no. 5, pp. 2198–2211, 2007.
- [17] B. Peeters and G. D. Roeck, "One-year monitoring of the z24-bridge: environmental effects versus damage events," *Earthquake Engineering and Structural Dynamics*, vol. 30, no. 2, pp. 149–171, 2001.
- [18] S. Law, X. Li, X. Zhu, and S. Chan, "Structural damage detection from wavelet packet sensitivity," *Engineering Structures*, vol. 27, no. 9, pp. 1339–1348, 2005.
- [19] J. Chou and J. Ghaboussi, "Genetic algorithm in structural damage detection," *Computers and Structures*, vol. 79, no. 14, pp. 1335–1353, 2001.
- [20] F. Qu, D. Zou, and X. Wang, "Substructural damage detection using neural networks and ica," in *Advances in Neural Networks*, F.-L. Yin, J. Wang, C. Guo et al., Eds., vol. 3173 of *Lecture Notes in Computer Science*, pp. 750–754, Springer, Berlin, Germany, 2004.
- [21] N. M. M. Maia, J. M. M. Silva, and A. M. R. Ribeiro, "Transmissibility concept in multi-degree-of-freedom systems," *Mechanical Systems and Signal Processing*, vol. 15, no. 1, pp. 129–137, 2001.
- [22] C. Devriendt and P. Guillaume, "Identification of modal parameters from transmissibility measurements," *Journal of Sound and Vibration*, vol. 314, no. 1-2, pp. 343–356, 2008.

- [23] C. Devriendt, G. D. Sitter, P. Guillaume et al., "An operational modal analysis approach based on parametrically identified multivariable transmissibilities," *Mechanical Systems and Signal Processing*, vol. 24, no. 5, pp. 1250–1259, 2010, special Issue: Operational Modal Analysis.
- [24] C. Devriendt, G. Steenackers, G. D. Sitter, and P. Guillaume, "From operating deflection shapes towards mode shapes using transmissibility measurements," *Mechanical Systems and Signal Processing*, vol. 24, no. 3, pp. 665–677, 2010.
- [25] J. Lardies, M. Ta, and M. Berthillier, "Modal parameter estimation based on the wavelet transform of output data," *Archive of Applied Mechanics*, vol. 73, no. 9–10, pp. 718–733, 2004.
- [26] 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.
- [27] J. P. Lynch and K. J. Loh, "A summary review of wireless sensors and sensor networks for structural health monitoring," *The Shock and Vibration Digest*, vol. 38, no. 2, pp. 91–128, 2006.
- [28] C. B. Yun and J. Min, "Smart sensing, monitoring, and damage detection for civil infrastructures," *KSCE Journal of Civil Engineering*, vol. 15, no. 1, pp. 1–14, 2011.
- [29] 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.
- [30] E. B. Flynn and M. D. Todd, "A Bayesian approach to optimal sensor placement for structural health monitoring with application to active sensing," *Mechanical Systems and Signal Processing*, vol. 24, no. 4, pp. 891–903, 2010.
- [31] M. Meo and G. Zumpano, "On the optimal sensor placement techniques for a bridge structure," *Engineering Structures*, vol. 27, no. 10, pp. 1488–1497, 2005.
- [32] R. Guratzsch, *Sensor placement optimization under uncertainty for structural health monitoring systems of hot aerospace structures*, Ph.D. dissertation, Vanderbilt University, 2007.
- [33] F. Udwadia and J. Garba, "Optimal sensor locations for structural identification," in *Proceedings of the JPL Workgroup on Identification and Control of Flexible Space Structures*, pp. 247–261, 1985.
- [34] T. Lim, "Sensor placement for on-orbit modal identification," in *Proceedings of the 32nd Conference AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials*, American Institute of Aeronautics and Astronautics, 1991.
- [35] K. Hiramoto, H. Doki, and G. Obinata, "Optimal sensor/actuator placement for active vibration control using explicit solution of algebraic Riccati equation," *Journal of Sound and Vibration*, vol. 229, no. 5, pp. 1057–1075, 2000.
- [36] P. Tongpadungrod, T. Rhys, and P. Brett, "An approach to optimise the critical sensor locations in one-dimensional novel distributive tactile surface to maximize performance," *Sensors & Actuators*, vol. 105, no. 1, pp. 47–54, 2003.
- [37] E. L. Lloyd and G. Xue, "Relay node placement in wireless sensor networks," *IEEE Transactions on Computers*, vol. 56, pp. 134–138, 2007.
- [38] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.
- [39] G. Pottie and W. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, pp. 51–58, 2000.
- [40] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, 2004.
- [41] Y. Lei, A. Kiremidjian, K. Nair, J. Lynch, and K. Law, "Algorithms for time synchronization of wireless structural monitoring sensors," *Earthquake Engineering and Structural Dynamics*, vol. 34, no. 6, pp. 555–573, 2005.
- [42] X. Jiang, Y. Tang, and Y. Lei, "Wireless sensor networks in structural health monitoring based on zigbee technology," in *Proceedings of the 3rd International Conference on Anti-Counterfeiting, Security, and Identification in Communication (ASID '09)*, pp. 449–452, IEEE Press, 2009.
- [43] A. Tanenbaum and M. Steen, *Distributed Systems: Principles and Paradigms*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001.
- [44] H. Chan, C. Zhang, P. Qing, T. Ooi, and S. Marotta, "Automatic sensor-fault detection system for comprehensive structural health monitoring system," in *Proceedings of the IMAC-XXIII: A Conference on Structural Dynamics, January*, Society for Experimental Mechanics, 2005.
- [45] Y. Gao, B. Spencer, and M. Ruiz-Sandoval, "Distributed computing strategy for structural health monitoring," *Structural Control and Health Monitoring*, vol. 13, no. 1, pp. 488–507, 2006.
- [46] A. T. Zimmerman, M. Shiraishi, R. A. Swartz, and J. P. Lynch, "Automated modal parameter estimation by parallel processing within wireless monitoring systems," *Journal of Infrastructure Systems*, vol. 14, no. 1, pp. 102–113, 2008.
- [47] S. H. Sim, B. Spencer, M. Zhang, and H. Xie, "Automated decentralized modal analysis using smart sensors," in *Proceedings of the SPIE*, vol. 7292, 2009.
- [48] A. Zimmerman and J. Lynch, "Market-based frequency domain decomposition for automated mode shape estimation in wireless sensor networks," *Structural Control and Health Monitoring*, vol. 17, no. 7, pp. 808–824, 2010.
- [49] J. Gun, L. Soon-Gie, J. Carletta, and T. Nagayama, "Decentralized damage identification using wavelet signal analysis embedded on wireless smart sensors," *Engineering Structures*, vol. 33, no. 7, pp. 2162–2172, 2011.
- [50] K. Worden, G. Manson, and D. Allman, "Experimental validation of a structural health monitoring methodology: part I. Novelty detection on a laboratory structure," *Journal of Sound and Vibration*, vol. 259, no. 2, pp. 323–343, 2003.
- [51] J. Toivola and J. Hollmén, "Feature extraction and selection from vibration measurements for structural health monitoring," in *Advances in Intelligent Data Analysis VIII*, vol. 5772 of *Lecture Notes in Computer Science*, pp. 213–224, Springer, Berlin, Germany, 2009.
- [52] G. Canales, L. Mevel, and M. Basseville, "Transmissibility based damage detection," in *Proceedings of the 27th International Modal Analysis Conference (IMAC-XXVII '09)*, Orlando, FL, USA, February 2009.
- [53] M. Di Renzo, M. Iezzi, and F. Graziosi, "Beyond routing via network coding: an overview of fundamental information-theoretic results," in *Proceedings IEEE 21st International Symposium on of the Personal Indoor and Mobile Radio Communications (PIMRC '10)*, pp. 2745–2750, September 2010.
- [54] E. Johnson, H. Lam, L. Katafygiotis, and J. Beck, "A benchmark problem for structural health monitoring and damage detection," in *Proceedings of the 14th Engineering Mechanics Conference*, pp. 317–324, 2000.
- [55] P. Welch, "The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.
- [56] T. J. Johnson and D. E. Adams, "Transmissibility as a differential indicator of structural damage," *Journal of Vibration and Acoustics*, vol. 124, no. 4, pp. 634–641, 2002.

Research Article

DI-GEP: A New Lifetime Extending Algorithm for Target Tracking in Wireless Sensor Networks

Shucheng Dai,¹ Chuan Li,¹ and Chun Chen²

¹ College of Computer Science, Sichuan University, Sichuan 61065, China

² Business School, Sichuan Normal University, Sichuan 61065, China

Correspondence should be addressed to Chuan Li, lcharles@scu.edu.cn

Received 25 May 2011; Accepted 4 January 2012

Academic Editor: Bahram Honary

Copyright © 2012 Shucheng Dai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless sensor networks (WSNs) are widely used in detecting, locating, and tracking moving objects. The cheap, low-powered, and energy-limited sensors that are set up in large areas may consume large portion of energy and disable the whole network. In this paper, a new energy-efficient method based on Distributed Incremental Gene Expression Programming (DI-GEP) is proposed to collaboratively mine moving patterns of moving targets in order to turn on/off some sensor nodes at certain time to save energy further. Meanwhile, an adjustable sliding window is designed to quickly train the latest collected location data in order to improve the efficiency of DI-GEP. The simulation results show that the proposed method effectively prolongs the network lifetime by around 25% compared with the EKF and ECPA.

1. Introduction

Recent advances in low-power micro-electro-mechanical system (MEMS) technology, wireless communications, and digital electronics have made it possible to design and develop highly integrated, yet low-cost, low-power, multifunctional microsensor nodes, with the capabilities of sensing, processing, and wireless communications. Once deployed in a certain region, the wireless sensor networks (WSN), composed of thousands of sensor nodes, can work for several years. Through cooperative processing of these sensor nodes, WSNs work in many areas, for example, civil, military, health, and so on. For example, WSNs can be deployed in a hospital to track and monitor patients to remotely collect the physiological data of a patient continuously. Unlike traditional networks, WSNs are self-organized, application specific, and data centric [1].

A wireless sensor node is typically battery operated. Thus, the most important constraint in WSNs is the low energy consumption requirement among their sensor nodes. Sensor nodes carry limited, generally irreplaceable, battery-power sources. So, WSNs must focus primarily on power conservation and provide inbuilt trade-off mechanisms that give end users the chance of prolonging network lifetime at

the cost of high quality of service (QoS). Sensor nodes may fail due to energy depletion and lead to network failure. So it is very important for WSN to operate energy efficiently. Raising research interest promotes us to develop energy-efficient protocols or algorithms for WSNs.

Target tracking is an important application in terms of WSNs [2]. Bayesian Network and Kalman filtering are two classical methods for achieving this task. One possible solution is as follows. The system state includes the position, direction, and velocity of the target. At each step, sensors near to the target form clusters and select a leader to perform the Kalman filtering, and the updated state is forwarded to cluster leaders chosen from the next step. The Kalman filtering implementation is straightforward in a centralized environment. But it is difficult in the extremely distributed environments such as WSNs due to the energy constraints and lower computation capability of sensor nodes.

The target tracking applications in WSNs are always limited by the inherent energy constraints of sensor nodes, aiming to improve the energy efficiency in target tracking applications in WSNs; the paper proposes a new scheme based on Gene Expression Programming (GEP) [3]. GEP is also adapted to fit for distributed environment. GEP works well in modeling the moving patterns of targets without

aprior knowledge. Based on the historical location information of the target, GEP automatically evolves a trajectory of a moving object. To handle the problem, this paper makes the following contributions.

- (1) A new algorithm named Distributed Incremental Gene Expression Programming (DI-GEP) is proposed to mine moving patterns of targets. The basic idea is that DI-GEP runs at multiple collaboratively working sensor nodes to mine the trajectory of a target.
- (2) An adjustable sliding window is adopted to ensure that distributed GEP can quickly train the latest collected location data. When new location data are received, old location data are discarded when prediction error exceeds a certain threshold, which is defined and can be calculated by (7). The policy ensures that succeeding evolutions can energy efficiently find latest moving patterns.
- (3) Extensive simulations are conducted on OMNet++, a discrete event simulator, to show that new algorithms effectively prolong the network lifetime by about 25% in average when compared to other algorithms, that is, EKF and ECPA.

The rest of the paper is organized as follows. Section 2 presents the related work on energy-saving algorithms. Section 3 gives the preliminaries including GEP-related and target tracking. Section 4 introduces the target diction model. Section 5 formally defines the problems. Section 6 proposes the main algorithms in our scheme. Section 7 gives the experimental analysis. Section 8 concludes this paper and gives the future research directions.

2. Related Work

There are many research efforts on target detection and tracking in terms of WSNs, which describes several aspects of collaborative signal processing [2, 4, 5] and real-time application for biologists to find the presence of individuals [6]. A set of approaches presented in [7–9] were proposed recently to solve the target localization and tracking problem with proximity binary sensors, which transmit only one bit information to indicate whether a target is present. The information transmitted among sensor nodes was greatly reduced, while the localization error was increased. Shrivastava et al. [8] proved that the accuracy in tracking a target is of the order of $\rho * R$, where R is the sensing radius and ρ is the sensor density, which articulates the common intuition that, for a fixed sensing radius, the accuracy improves linearly with an increasing sensor density, which shows that, for a fixed number of sensors, the accuracy improves linearly with an increase in the sensing radius. Dai et al. present a light weight target tracking method based on densely distributed sensor networks [10, 11] and also propose a new node deployment policy for target tracing applications in WSNs [12] to further improve target tracking performance and quality including accuracy, network lifetime, energy consumption level, trends analysis, and so forth.

The lifetime of a WSN depends greatly on power consumption from each sensor node. Energy-efficient algorithms, protocols, and node hardware and software designing technologies can help prolong the lifetime of the network. Several approaches have been proposed at hardware and software levels to design energy efficient CPU, OS, algorithms, and communication protocols [1]. Dynamic power management (DPM) schemes have been proposed in [13–15] to reduce the power consumption by selectively turning off idle components, such as radio frequency (RF) transmitter, RF receiver, sensing device, A/D converter, and the sensor node.

Target tracking applications are special and have their own characteristics. It is unnecessary to turn on all sensor nodes because an object only appears at certain time and place. It is feasible to turn off some idle nodes if we can predict the time and place where the object will appear. Classical target tracking algorithms such as Bayesian Network and Kalman filtering cannot be directly used to predict moving patterns of targets in WSNs due to resource limitations.

To track moving targets energy efficiently, Allegretti et al. [16] proposed a solution based on CA (Cellular Automata) to reduce long distance communications among nodes because of its locally data exchanging scheme, but a higher power consumption is introduced because it cannot turn off those nodes that are far away from the moving object. Qing et al. [17] proposed ECPA (Enhanced Closest Point Approach) to predict the location of targets during the phase of moving, but the velocity and direction calculation algorithms with regard to the targets are computation intensive for sensor nodes, which often have low power and computation capability.

3. Preliminary

3.1. Introduction of Gene Expression Programming. Gene Expression Programming (GEP) was proposed by Ferreira in 2006 [3]. As a new member of Evolutionary Algorithm (EA) family, GEP is widely applied in data mining areas, that is, function finding, classification, association rule mining, time series prediction, parameter optimization, and digital circuit design, and so forth. In GEP, Genotype (Chromosome) and Phenotype (Expression Tree (ET)) are separated. Without prior knowledge, GEP automatically evolves over training data and discovers knowledge as mathematical formula depicting movement patterns of moving objects in WSNs.

In GEP, an individual, that is, a solution corresponding to a problem, is represented as linear fixed-length string named chromosome. It contains one or more genes. Each gene is decoded into a nonlinear expression tree (ET). Decoded ETs are linked together by prespecified linking function symbols such as plus (+) and minus (-). One chromosome represents one formula that is, the solution to a specific problem. Genetic operations are applied on chromosomes, that is, genotype and selection operations are performed on ETs.

A gene in GEP consists of head and tail. The head contains symbols from either function symbol set (F) or terminal symbol set (T) and the tail only contains symbols

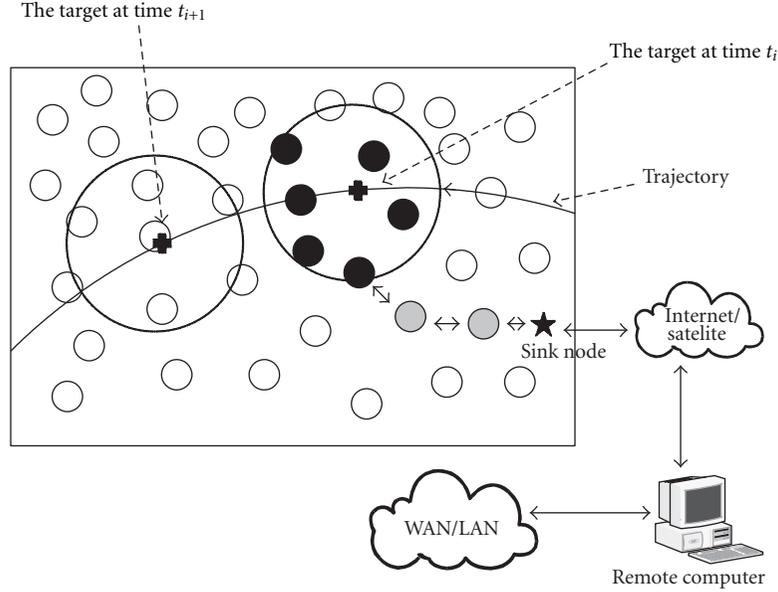


FIGURE 2: Target tracking model.

- (2) Node scheduling algorithm. It activates the necessary nodes and turn off unrelated sensors at certain time in future to save energy.
- (3) Sliding window strategy is used to improve performance of DI-GEP.

The experiments in Section 7 will demonstrate the effectiveness and efficiency of proposed methods.

4. Target Detection Model

Sensor nodes receive the physical signal and convert it to electrical signal. Based on the variation of electrical signal, sensors can detect the existence of target.

To describe the model formally, we make the following three assumptions.

- A_1 : There are N sensors $s_i \in S$ distributed randomly or manually across the monitored areas, where S is the set of all sensors. Each sensor detects targets by its reading x_i , where $i = 1, 2, \dots, N$.
- A_2 : There is a single target anytime.
Note that, by Assumption (A_2), a present target is depicted by (H_1), and an absent target is represented by (H_0). The criterion is based on the following formulations [18]:

$$\begin{aligned} H_0 : x_i &= n_i, \\ H_1 : x_i &= w_i + n_i, \end{aligned} \quad (2)$$

where w_i is the obtained signal by sensor s_i . Several physical signals such as sound and electromagnetic wave have signal strength decaying according to the power law, and the noise is represented by n_i .

- A_3 (borrowed from [19]): Let w_t is the power emitted by target

$$w_i = \begin{cases} w_t, & d_i < d_0, \\ \frac{w_t}{(d_i/d_0)^k}, & d_i \geq d_0, \end{cases} \quad (3)$$

where d_0 is determined by the target shape and size which is set to be small enough and satisfies $d_i > d_0$, where d_i is the distance between the target and sensor node x_i and k is the decaying factor which is set to 2–5 according to different physical signals and its environment.

This study adopts a practical target detection model shown in Figure 3, which satisfies

$$p_i = \begin{cases} 1, & d_i \leq r_l, \\ \left(\frac{d_i}{r_0}\right)^k, & r_l \leq d_i \leq r_u, \\ 0, & d_i > r_u, \end{cases} \quad (4)$$

where r_l is the lower bound (LB) of sensing range, p_i is the probability of a target detected by the sensor node s_i , and r_u is the upper bound (UB) sensing range.

5. Problem Formulation

A trajectory of a moving object is treated as a sequence of time-stamped locations that are collected by sensor nodes around the target. It is described as follows.

Definition 2 (Trajectory). A trajectory of a moving object is a time sequence with time interval Δt :

$$P(t) = [X(t), Y(t)] = \{(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)\}, \quad (5)$$

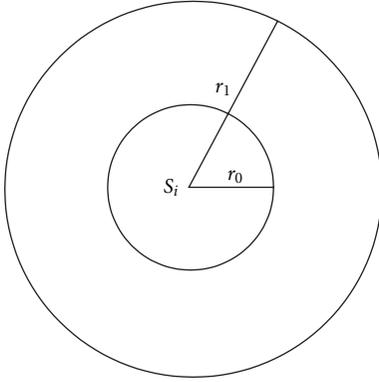


FIGURE 3: Target detection model.

where for all $i \in [0, n]$, $t_i < t_{i+1}$, $t_{i+1} = t_i + \Delta t$ (x_i, y_i) is 2D points that represent locations of the target appeared at time t_i and $x(t_i) = x_i$, $y(t_i) = y_i$, $P(i) = [x_i, y_i]$. Δt is used to sample the locations collected by sensors to improve our algorithms energy efficiency as well as performance. Because the location data may be of large scale, which will put great burden on sensors and exhausts a great of energy because of huge amount of computations and communications.

$P(t)$ describes a varying kinds of trajectories, that is, line segments, quadric curves, cubic curves, and splines. Once $P(t)$ is obtained, it is easy to achieve single-step or multiple-step location predictions.

$P(t)$ can be obtained by trajectory mining algorithm. In terms of target tracking applications, there are several unnecessary historical location data during evolving process in distributed GEP. To deal with this problem, we adopt a sliding window prediction method (SWP) to load the latest historical data to train trajectories. The basic idea of SWP is given below.

Given the historical location data $P(0), P(1), \dots, P(n)$ with length $n + 1$. The sliding window size is denoted as h ($h \leq n + 1$).

- (a) Find a formula $\hat{P}(t) = [f(t), g(t)]$ from h samples and predict the location at time instant m , ($m > n - 1$) by (6). Example 3 illustrates the phases of evolutions of trajectories.

$$\hat{P}(m) = [\hat{X}(m), \hat{Y}(m)] = [f(t_m), g(t_m)]. \quad (6)$$

- (b) During evolving process, the size of sliding window h determines how many historical location data are used. The smaller h leads to less energy consumption and faster convergence speed. h is adjusted based on the location prediction error ε , geometric distance between the prediction value and the real measurement. ε should be as small as possible and is calculated by

$$\varepsilon = \sqrt{(\hat{x}(m) - x(m))^2 + (\hat{y}(m) - y(m))^2}. \quad (7)$$

TABLE 1: Location data obtained.

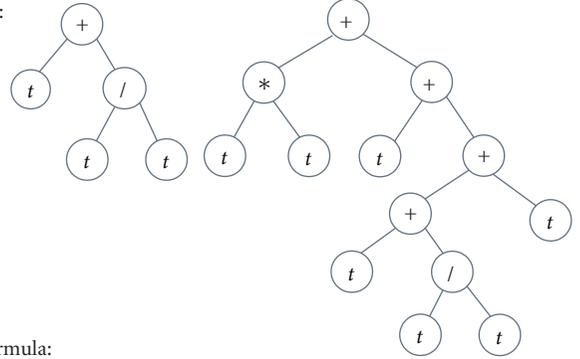
t_i	x	y
0	1	1
1	2	5
2	3	11
3	4	19

Chromosome:

+t/tttttttt

+*+ ttt ++ tt/tt

ET:



Formula:

$t + 1$

$t^2 + 3t + 1$

FIGURE 4: An example of a tracking trajectory.

In target-tracking applications, the trade-off between the energy consumption and the prediction accuracy is balanced by adjusting ε to satisfy different application requirements. In densely distributed sensor networks, tracking-tolerant environment or fast response time tracking areas, ε can be set to a bigger value to save energy. But, in some areas with higher tracking accuracy with slow moving targets environment, ε can be set to a smaller one. In sum, ε cannot be set to zero since the estimated trajectory would always deviate from the actual path targets passed.

Example 3. The historical locations are listed in Table 1.

Thus, f may be $P(t) = [t + 1, t^2 + 3t + 0.5]$, $P(t) = [t + 2, t^2 + 2t + 1]$ or even perfect approximation $P(t) = [t + 1, t^2 + 3t + 1]$ given in Figure 4. All these approximations are suitable in environments with different prediction accuracy requirements.

6. DI-GEP Scheme

6.1. Fitness Evaluation of Individual. In evolutionary computations, fitness functions and selection environments are the two very important faces of fitness and are, therefore, intricately connected. When we speak of the fitness of an individual, on the one hand, it is always relative to a particular environment and, on the other, it is also relative to the measure (the fitness function) we are using to evaluate

them. Consequently, the success of a problem not only depends on the way the fitness function is designed but also on the quality of the selection environment [3].

Combining the fitness evaluation and prediction error, DI-GEP calculates the fitness of each individual in distinct populations by

$$E_i = \frac{1}{h} \sum_{j=1}^h (\varepsilon_i^2), \quad (8)$$

where ε_i is the evaluation error of the i th location data and E_i is the fitness value of the i th individual.

6.2. *Trajectory Mining Algorithm.* The main steps of trajectory search algorithms are given below.

- (1) Sensor nodes are activated based on the node scheduling algorithm.
- (2) Communications occur among sensor nodes when one node succeeds in obtaining a trajectory and notifies other nodes.
- (3) Other nodes stop running their algorithms and obtain the trajectory to predict future location of the moving object.

Figure 5 details the flowchart of DI-GEP.

DI-GEP stops if one of these stopping criteria is satisfied.

- (1) The maximum number of generations is reached.
- (2) DI-GEP exceeds the specified runtime.
- (3) One or more other nodes send stopping signal to the node.
- (4) The node succeeds in obtaining a trajectory.

The implementation of DI-GEP is described in Algorithm 1. It mines a trajectory represented as individual in DI-GEP.

6.3. *Location Prediction and Node Scheduling.* Once a trajectory is found, the model uses it to predict the location where the target will appear as at time t_{i+j} by (6), where $0 < j \leq L$ and L is the prediction length that is used in single-step or multistep predictions.

To reduce computational cost, we do not use a circle but a square to select nodes around $\hat{P}(t_{i+1})$. If sensor node $s_k(x_k, y_k)$ satisfies (9), then it should be selected and activated at time t_{i+j} to detect the target

$$\begin{aligned} f(t_{i+j}) - r_0 \leq x_k \leq f(t_{i+j}) + r_0, \quad 0 < j \leq L, \\ g(t_{i+j}) - r_0 \leq y_k \leq g(t_{i+j}) + r_0, \quad 0 < j \leq L. \end{aligned} \quad (9)$$

The node scheduling algorithm is given in Algorithm 2.

6.4. *Incremental Evolution Strategy.* In real-world practice, a trajectory of a target is very complex and variable. Thus, to improve the performance of DI-GEP, the key steps of our policy are as follows.

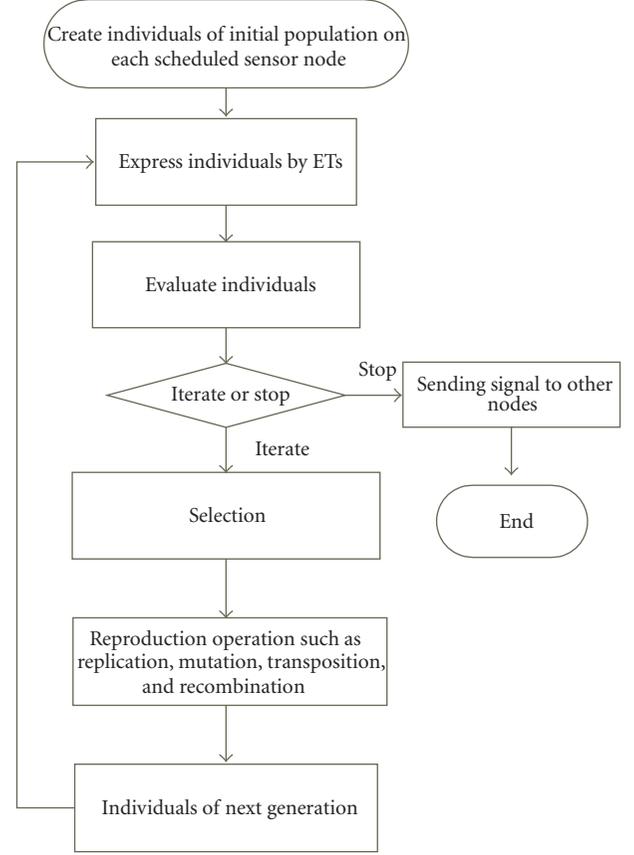


FIGURE 5: The workflow of distributed GEP.

Input: settings and historic location data
Output: one individual representing a trajectory or null if no trajectory is found

- (1) load historic location data of size h and initial configuration
- (2) randomly create an initial population
- (3) decode each chromosome into one ET
- (4) calculate each chromosome's fitness by (8).
- (5) while (stopping criteria are not satisfied) {
- (6) select individuals, generate next generation
- (7) apply genetic operations sequentially on the new generation
- (8) decode each chromosome into ET.
- (9) calculate each chromosome's fitness
- (10) }
- (11) return an individual with best fitness.

ALGORITHM 1: Distributed GEP trajectory mining.

- (1) Trajectory is described as a curve. Any complex curve can be spitted into multiple simpler curves that are described as line segments, quadratic curves, or cubic curves. This method can not only ensure the flexibility of modeling the trajectory but also guarantee less computation cost.
- (2) To capture the variation of a trajectory and accelerate the evolving process, sliding window policy is

Input: functions $\hat{P}(t)$ found in Algorithm 1 and prediction length L .

Output: activated sensor nodes

```

(1) for ( $k = 0; k < L; k++$ ) {
(2)   for each (sensor node  $s_j(x_j, y_j)$  in WSN) {
(3)     if ( $f(t_{i+k+1}) - R \leq x_j$  and  $x_j \leq f(t_{i+k+1}) + R$ 
(4)       and  $g(t_{i+k+1}) - R \leq y_j$  and  $y_j \leq g(t_{i+k+1}) + R$ )
(5)        $S_i$  is scheduled awake at  $t_{i+k+1}$ 
(6)     else
(7)        $S_i$  is scheduled asleep at  $t_{i+k+1}$ 
(8)   } }
```

ALGORITHM 2: Node scheduling.

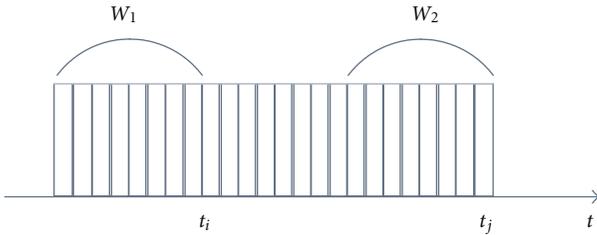


FIGURE 6: Sliding window.

proposed to keep a certain number of latest historical data during individual evolving.

- (3) When the obtained function cannot represent the current moving behaviors, that is, the prediction error is greater than a certain threshold; distributed GEP and the node scheduling algorithm should run again with location data in sliding window.

We assume that a trajectory function $P(t)$ is obtained through historical location data sampled at $t_i, t_{i-1}, \dots, t_{i-h-1}$. These data fall in the sliding window w_1 described in Figure 6.

$P(t)$ works well in predicting the future location during the time interval $[t_{i+1}, t_{j-1}]$, that is, $\varepsilon \leq \tau$. Suppose that at time t_j , $P(t)$ does not work well because $\varepsilon > \tau$, the trajectory should be recalculated as follows.

- (1) The sliding window moves to w_2 to include the latest location data. The process can be simplified by w_1 sliding right when prediction is performed to reduce memory usage.
- (2) DI-GEP and the node scheduling algorithm run again.

The performance of these algorithms is evaluated on OMNet ++ and Castalia.

In order to compare the performance with other target tracking algorithms, we evaluate DI-GEP, ECPA, and extended Kalman filtering (EKF). The target cannot be found until the network fails, and the tracking task stops simultaneously.

TABLE 2: Parameters setting in DI-GEP.

Parameters	values
Function set	{+, -, *, /}
Basic terminal set	{t}
Number of generations	1000
Population size	100
Head length	7
Mutation rate	0.044
Inversion rate	0.1
IS rate	0.1
RIS rate	0.1
Length of insertion sequence	{1, 2, 3}
One-point recombination rate	0.3
Two-point recombination rate	0.3
Gene recombination rate	0.1

7. Experiments

7.1. Sensor Networks Setting. Suppose that hundreds of sensor nodes are uniformly distributed in a square of 100×100 meters. A target can present at a random place in the WSNs and sends signal with the strength of w_t . The signal decays according to (3), with parameters $d_0 = 0.15$ and $w_t = 15d_0^{-k}$. The prediction accuracy τ is 2 meters and $h = 7$.

The sensing range r_o of sensor nodes is set to 7 and r_1 is set to 10. The sensing energy e_s is 100 uJ, and transmission (receiving and sending) energy for one packet is $e_t = e_r = 100$ uJ. The initial energy of each sensor node is 100 mJ. Parameters used in distributed GEP are listed in Table 2.

7.2. Network Lifetime. Suppose that different number of sensor nodes are uniformly distributed in the grid network, this experiment analyses the impact of the number of sensor nodes on the network lifetime. The results are given in Figure 7. It shows that the three algorithms often obtain longer network lifetime when the number of sensor nodes gets bigger. The network lifetime of DI-GEP is averagely 35% longer than EKF and ECPA.

7.3. Energy Consumption. In this experiment, four hundred sensor nodes are used to monitor the grid network. The sensor nodes are manually distributed at cross points in the grid network.

This experiment analyzes the energy consumption of three algorithms. The results are given in Figure 8. The results show that the total left energy decreases when the time passes. The total left energy cannot be zero because all these algorithms are invalid when the network fails. Note that, some sensor nodes consume their energy and cannot communicate with other nodes any more. Meanwhile, DI-GEP performs better than EKF and ECPA. This is because it consumes less energy than EKF and ECPA after running the same time.

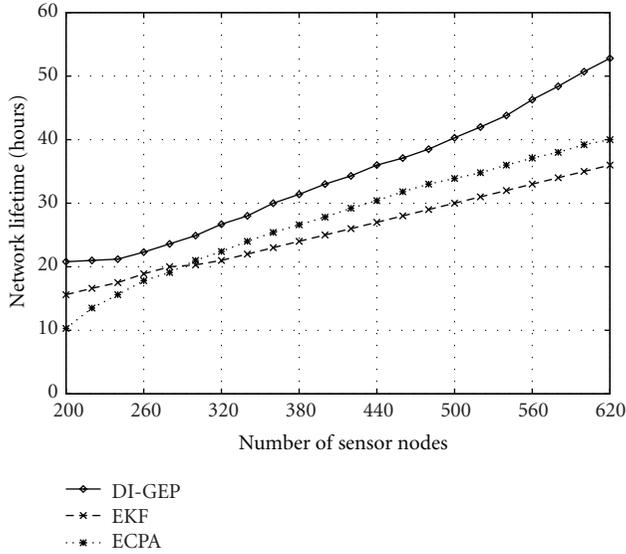


FIGURE 7: Network lifetime and number of sensor nodes.

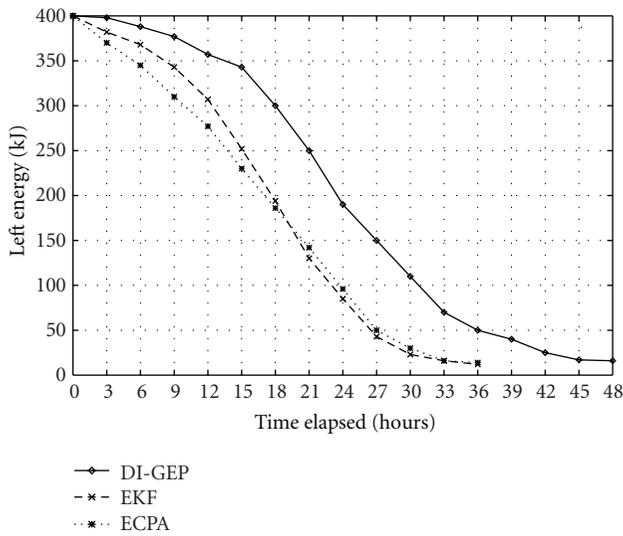


FIGURE 8: Energy consumption and time elapsed.

7.4. Active Nodes. This experiment uses the similar setting as given in the previous section and testifies the influence of the number of the active nodes as shown in Figure 9. DI-GEP outperforms EKF and ECPA in node scheduling because of its better trajectory prediction, so the number of active nodes in DI-GEP is about 25, 30% less than that in EKF and ECPA, separately.

7.5. Prediction Accuracy. This experiment uses the same setting as given in previous section and will testify that the prediction accuracy can heavily affect the network lifetime. The results are shown in Figure 10. Distributed GEP outperforms EKF and ECPA because of its better trajectory prediction, so at the same prediction accuracy, the network lifetime is averagely 28% longer than that in EKF and ECPA.

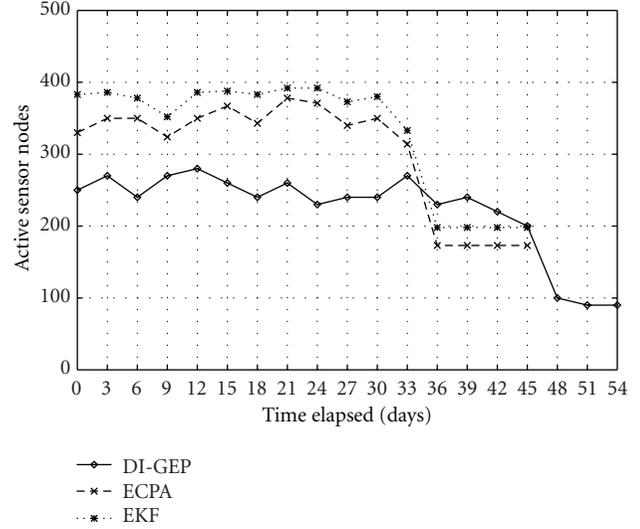


FIGURE 9: Active nodes and time elapsed.

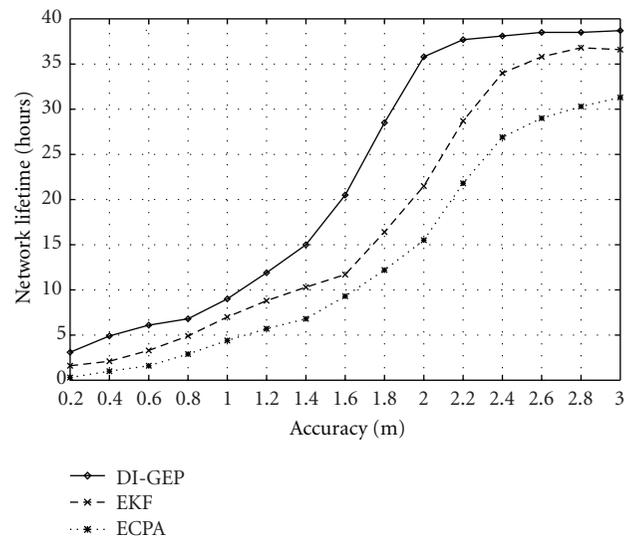


FIGURE 10: Network lifetime and prediction accuracy.

8. Conclusions

In order to track targets energy-efficiently in WSNs, we presented a distributed incremental algorithm based on GEP for target tracking applications in WSNs, proposed sliding window policy for distributed GEP to improve evolution process, proposed a new target tracking model, and give extensive experimental results to show the good performance of our method.

The future work includes (a) optimize DI-GEP to capture abrupt moving behaviors, (b) optimize DI-GEP to suit randomly distributed wireless sensor networks, and (c) consider border intrusion detection to save more energy in the initial state.

Acknowledgments

The work in this paper was partially supported by the National Science Foundation of China under Grant no. 61103043, and Youth Fund of Sichuan University (project No. 2030404134011 and 2010SCU11049). The authors would like to express their gratitude to these two entities for their financial and technical support.

References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.
- [2] D. Li, K. D. Wong, Y. H. Hu et al., "Detection, classification and tracking of targets in distributed sensor networks," *IEEE Signal Processing Magazine*, pp. 17–29, 2002.
- [3] C. Ferreira, *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*, Springer, Berlin, Germany, 2006.
- [4] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration for tracking applications," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, 2002.
- [5] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, "Distributed target classification and tracking in sensor networks," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1163–1171, 2003.
- [6] A. M. Ali, K. Yao, T. C. Collier, C. E. Taylor, D. T. Blumstein, and L. Girod, "An empirical study of collaborative acoustic source localization," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 41–50, New York, NY, USA, April 2007.
- [7] J. Singh, U. Madhow, R. Kumar, S. Suri, and R. Cagley, "Tracking multiple targets using binary proximity sensors," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 529–538, New York, NY, USA, April 2007.
- [8] N. Shrivastava, R. Mudumbai, U. Madhow, and S. Suri, "Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 251–264, New York, NY, USA, November 2006.
- [9] W. Kim, K. Mechtov, J. Y. Choi, and S. Ham, "On target tracking with binary proximity sensors," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 301–308, April 2005.
- [10] S. Dai, C. Chen, C. Tang et al., "Light-weight target tracking in dense wireless sensor networks," in *Proceedings of the 5th International Conference on Mobile Ad-hoc and Sensor Networks*, IEEE Computer Society, pp. 480–487, 2009.
- [11] S. Dai, C. Tang, S. Qiao, Y. Wang, H. Li, and C. Li, "An energy-efficient tracking algorithm based on gene expression programming in wireless sensor networks," in *Proceedings of the 1st International Conference on Information Science and Engineering (ICISE '09)*, IEEE Computer Society, pp. 774–777, Nanjing, China, December 2009.
- [12] S. Dai, C. Tang, S. Qiao, K. Xu, H. Li, and J. Zhu, "Optimal multiple sink nodes deployment in wireless sensor networks based on gene expression programming," in *Proceedings of the 2nd International Conference on Communication Software and Networks (ICCSN '10)*, IEEE Computer Society, pp. 355–359, Singapore, February 2010.
- [13] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Design and Test of Computers*, vol. 18, no. 2, pp. 62–74, 2001.
- [14] C. Lin, Y. X. He, and N. Xiong, "An energy-efficient dynamic power management in wireless sensor networks," in *Proceedings of the 5th International Symposium on Parallel and Distributed Computing (ISPDC '06)*, pp. 148–154, Timisoara, Romania, July 2006.
- [15] X. Wang, J. Ma, and S. Wang, "Collaborative deployment optimization and dynamic power management in wireless sensor networks," in *Proceedings of the 5th International Conference on Grid and Cooperative Computing (GCC '06)*, pp. 121–128, Hunan, China, October 2006.
- [16] D. G. Allegretti, G. T. Kenyon, and W. C. Priedhorsky, "Cellular automata for distributed sensor networks," *International Journal of High Performance Computing Applications*, vol. 22, no. 2, pp. 167–176, 2008.
- [17] Y. Qing, A. Lim, K. Casey, and R. K. Neelisetti, "Real-time target tracking with CPA algorithm in wireless sensor networks," in *Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '08)*, pp. 305–313, San Francisco, Calif, USA, June 2008.
- [18] W. Wang, V. Srinivasan, K. C. Chua, and B. Wang, "Energy-efficient coverage for target detection in wireless sensor networks," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 313–322, April 2007.
- [19] T. Clouqueur, K. K. Saluja, and P. Ramanathan, "Fault tolerance in collaborative sensor networks for target detection," *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 320–333, 2004.

Research Article

Enabling Collaborative Musical Activities through Wireless Sensor Networks

Santiago J. Barro, Tiago M. Fernández-Caramés, and Carlos J. Escudero

Departamento de Electrónica y Sistemas, Universidade da Coruña, 15071 A Coruña, Spain

Correspondence should be addressed to Carlos J. Escudero, escudero@udc.es

Received 14 July 2011; Revised 9 December 2011; Accepted 10 December 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Santiago J. Barro et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In professional audio production the setup of electronic musical devices is a time-consuming and error-prone process, as it involves manual operations like establishing local configurations and carrying cables for each device. Such is the case of MIDI (musical instrument digital interface), which is widely used in the context of musical applications. On the other hand, the capabilities of WSN (wireless sensor networks) allow developers to build up more complex applications, since nodes have the ability of autoidentifying, autoconfiguring, and establishing associations with other nodes, behaving in a smarter way than other networks. In this paper, we propose the use of an optimized WSN network for interconnecting MIDI devices. This network has been named collaborative musical wireless network (CMWN): it eases device configuration, enables musical collaboration, and allows artists to explore new ways of expression. The paper also presents the hardware and performance results of a prototype able to create CMWNs.

1. Introduction

The field of professional audio production refers to all those activities in some way related with the processing of sound using electronic means [1], that is, musical performances, composition and arrangement [2], artistic performances, and multimedia spectacles [3]. Technology plays an important role in the world of professional audio production. Several types of device are involved, such as microphones to record voices and musical instruments, audio processors to apply special effects (echo, delay, etc.), and multitrack mixers to synchronize audio and video in soundtracks for documentaries and cinema, amongst others. The quality of the artistic results depends on the selection of the device settings, which is usually done by a technician with special skills in music and technology, either by using cables or software.

Communication between musical devices is possible thanks to musical protocols, which can be classified into two categories: wave-oriented protocols and control-oriented protocols [4]. Wave-oriented protocols carry the sound data, either in analog or digital format, and contain all

the information needed to play the sound. In contrast, control-oriented protocols are used to intercommunicate musical devices internally, usually to cause an action to be performed in response to the occurrence of an event. This is the case of the MIDI protocol, in which the information transmitted needs to be interpreted by a synthesizer before an audible waveform can be obtained [2]. The combination of both wave-oriented and control-oriented protocols allows artists to create a complete multimedia performance through the interaction of various multimedia nodes, which form a multimedia network. Control-oriented protocols have the advantage of being much easier to process because the musical events are explicit. For instance, with wave-oriented protocols a signal processing analysis algorithm like the one needed to detect sound frequency [5] can work incorrectly due to the presence of different musicians playing simultaneously in live concerts, as the harmonics of their instruments are mixed with the noise. However, control-oriented protocols like MIDI detect the sound immediately, simply by reading the corresponding value of a MIDI message.

Wireless sensor networks (WSN) are mainly used to collect data from a set of sensors distributed over a wide area without the need for a physical structure [6]. By design, these sensors are inexpensive and low-powered and they are applied to fields like agriculture, health, industry, and so forth. Typically, the sensors form *ad hoc* communication networks able to self-organize and auto-configure their nodes. These features can be taken into account for professional audio production, where, as we have seen, it is necessary to interconnect and configure a large number of musical devices.

By using WSN to provide connection support for control-oriented protocols (like MIDI) we can obtain many benefits and new possibilities. For example, in the traditional MIDI connections devices are classified as masters or slaves, with masters being responsible for initiating a communication. However, in WSN any device can initiate communications, providing greater flexibility. Additionally, WSN offers the possibility of auto-identifying, auto-configuring, and associating devices, allowing designers to provide smarter musical applications. The main limitation of WSN could be the low bandwidth usually available, but this does not represent a real problem for this application, since control-oriented protocols are not highly demanding in terms of bandwidth.

Therefore, this article proposes the use of WSN technology for interconnecting musical MIDI devices. This kind of network, named musical collaborative wireless network (MCWN) allows nodes not only to communicate wirelessly, but also to auto-identify and associate with other nodes in order to perform collaborative activities. By collaboration we refer to any association between two or more devices, such as a score digitalization using a musical instrument and a score editor, or a light show for a live concert, where instruments and lighting system have to be synchronized. As a proof of this idea, the article also introduces a preliminary prototype of an adapter that enables any MIDI standard device to join an MCWN network easily.

The article is structured as follows. Section 2 defines the requirements of the system to be developed. Section 3 gives a detailed description of the background of multimedia protocols, MIDI devices that can form part of MCWN networks and commercial wireless MIDI transceivers that could be used to implement some of the ideas put forward in this article. Section 4 is devoted to describing the MCWN network architecture, explaining how to extend a standard WSN network to support collaborative activities. Section 5 presents a prototype that allows any standard MIDI device to join an MCWN network and test basic collaborative activities. Section 6 describes real scenarios that could benefit from the inclusion of collaborative features in their normal operations. Finally, Section 7 is dedicated to conclusions and future work.

2. Problem Statement

This article describes a WSN-based system that enables electronic musical devices to establish collaborative communications. Our system has been designed according to the

following requirements, which should be present in an ideal collaborative wireless interface.

- (1) MIDI compatible. The ideal system must be compatible with the multimedia protocols implemented in most academic/commercial devices, otherwise its adoption in real scenarios would be limited. MIDI is a good option, as it has been widely adopted within the musical industry.
- (2) Transparent operation. The communications system should be able to transmit the data through a wireless network as if the musical device was using its regular nonwireless interface.
- (3) Collaborative activities. Communications among all the nodes are possible, allowing the intelligent association of two or more of them in order to carry out collaborative activities, such as a light show for a live concert. This requirement contrasts with the limitations associated with the master-slave communications model used by multimedia protocols such as MIDI.
- (4) Wireless communications. The use of cables can be problematic in certain cases, such as when assembling and disassembling musical devices on stage. Moreover, wireless devices are more user friendly, since they allow for automatic configuration and greater freedom of movement for the musician.
- (5) License-free band. The use of a license-free band is almost mandatory, since it reduces the expenses associated with the use of the technology.
- (6) Coverage extension. In some scenarios it would be necessary to extend the coverage to a larger area using relay nodes.
- (7) Bidirectional communications. Commercial wireless adapters only allow one-way communications, thus limiting the range of the activities to be performed within the multimedia network (e.g., devices could report problems in the configuration or malfunctions).
- (8) Auto-identification and Auto-configuration. Traditionally, the setup of multimedia systems has been performed manually using cables and local configurations. With a system of the characteristics proposed, the automatic configuration of the devices is possible, since devices can have enough knowledge about other devices within the network to make intelligent decisions as they are switched on.
- (9) Low latency. Musical performances require low latency devices since they occur in real time. Furthermore, sound delays can lead musicians to reject the use of the interface.

In the following section (Subsection 3.2) currently available interfaces are analyzed in order to determine whether they fulfill the previous requirements.

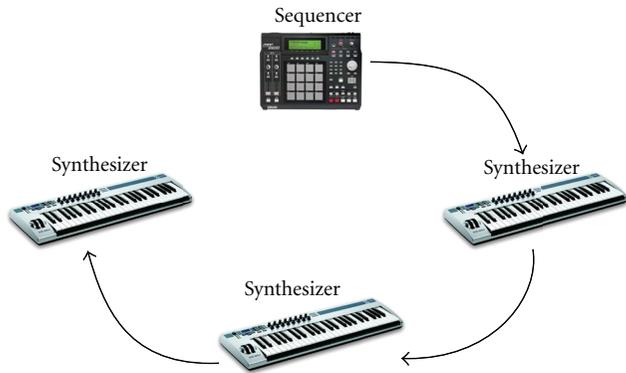


FIGURE 1: Network topologies available in MIDI technology (the synthesizer is an E-Mu Xboard49, while the sequencer is an Akai MPC2500).

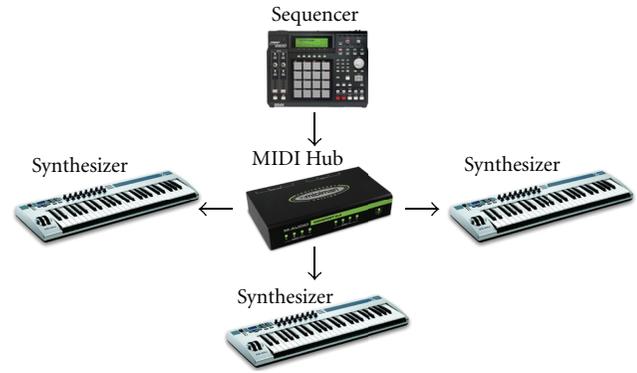


FIGURE 2: Network topologies available in MIDI Technology (the synthesizer is an E-Mu Xboard49, while the sequencer is an Akai MPC2500. The MIDI hub is an M-Audio MidiSport 4 × 4).

3. State of the Art

3.1. Traditional Multimedia Protocols versus WSN Protocols. MIDI, which stands for *musical instrument digital interface*, is one of the most widespread protocols found in the field of professional audio production [2]. It was designed by a group of manufacturers under the name of MMA Association [7] in the late 80s with the main aim of ensuring compatibility between their products, something that until then had proved almost impossible because each manufacturer had proprietary connectors, cables, and protocols. As time went by, MIDI became very popular among musicians and was extended to support more complex scenarios. Although nowadays several academic and commercial alternatives can be found (e.g., OSC [8], mLAN [9], or HD Protocol [10]), MIDI remains the most popular, as it is the protocol being used by most of the musical instruments, devices, and software currently available.

MIDI is a multimedia protocol characterized by being event oriented, state based, unidirectional, and master-slave. MIDI does not carry any sound, but it does carry the information that a synthesizer can use to produce sound. MIDI was initially designed to solve the basic problem of intercommunicating one keyboard with one or more synthesizers. Events are represented by MIDI messages, which are created whenever a musician generates an event during his/her performance.

MIDI can be regarded as a multimedia network protocol. Figures 1 and 2 show typical connections between one sequencer and several synthesizers. MIDI network topologies can be said to act like unidirectional one-to-one communications: data travel only from the master to the slave. To achieve bidirectional communications, two MIDI cables are required. It is possible to interconnect several slaves to one master by using the so-called “daisy-chain” technique (shown in Figure 1) or a MIDI Hub (shown in Figure 2). In the latter case, the data sent by the master is received by all the slaves, while the same MIDI cable is shared by all the devices. Also, note that MIDI defines channels, which allows synthesizers to respond to only one specific channel.

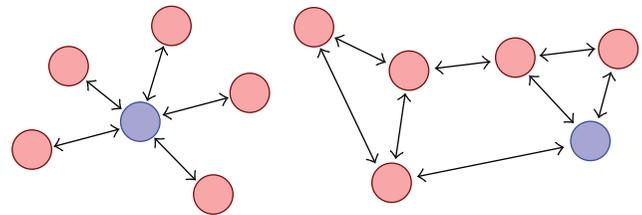


FIGURE 3: Network topologies available in WSN technology: star network and mesh network.

Although bidirectional communications are not strictly necessary, they can improve user experience, as the master can receive feedback from its slaves to facilitate maintenance operations (e.g., slaves can report hardware failures), to enable device-discovery within a MIDI network or even to solve configuration problems (e.g., to fix incorrect MIDI channel configurations or to set up a synthesizer). It must be said that this feature is already used by other multimedia protocols, for example, DMX512 [11].

The difference between WSN and MIDI network topologies can be observed when comparing Figures 1 and 2 with Figure 3, where examples of star and mesh network topologies are shown. WSN networks try to maximize the number of possible connections between network nodes, leading to more complex topologies such as the aforementioned star and mesh topologies. In star networks, all nodes interact with one central node, which is called the coordinator. Unfortunately, the area covered by star networks is restricted by the transmission power of the coordinator. This problem does not exist in mesh networks, as all nodes within the network can provide network connectivity, not only the coordinator. Thus, the area covered by mesh networks is wider than in a star network.

WSN technologies offer new possibilities for music applications: high interconnection capacity, bidirectional low-powered communications, low cost, and a communication range in 2.4 GHz of up to 120 meters (depending on the environment, but usually enough for a live concert stage). Due to these characteristics, WSN-based music applications emerge as an interesting field, whose optimization is critical

due to tight restrictions on bandwidth and latency, especially when performing live music.

Although MIDI has its own specific connectors (the well-known DIN-5), the fact is that MIDI protocol is independent from the transmission media being used to interconnect the musical devices. In fact, the MMA Association released documents dealing with USB, IEEE 1394 (or FireWire) [12], Ethernet [13], and RTP (Real-Time Protocol) [14] transmission media to use with MIDI. As detailed in Section 3.2.2, there are also wireless MIDI interfaces available on the market. Such interfaces are wireless replacements of one-to-one MIDI communications and therefore do not allow the one-to-multiple node communications that would support collaborations between several musical devices.

The features of multimedia WSN networks also include the fact that the intercommunication and integration of devices become easier. Moreover, a multimedia WSN network would extend communication to other devices, since it removes the natural boundary of cables and connectors, and thus facilitates collaboration between musical and nonmusical devices. For instance, it is straightforward to translate musical notes into light movements and colors (e.g., [15]), helping artists find new ways of expression.

3.2. MIDI Devices

3.2.1. Generic MIDI Devices. The MIDI protocol was initially conceived to support musical keyboards and synthesizers but, as time went by, the range of devices that could be connected to a MIDI network increased, extending to a high proportion of all the currently manufactured electronic musical devices.

Available MIDI devices can be divided into several groups. The first group is called “instrument-like” devices, and includes any MIDI device that emulates, at any level of detail, the physical appearance and touch feeling of an existing instrument. Examples of such group are the Yamaha WX5 [16] (a monophonic wind controller with a fingering similar to a flute, clarinet, or saxophone), the AKAI EWI4000S [17] (another wind controller), the Steiner MIDI EVI [18] (a trumpet-style wind controller), and the Morrison Digital Trumpet [19] (a brass-style controller designed by Steve Marshall with the Australian multi-instrumentalist James Morrison). The first group also includes novel and creative musical instruments that do not emulate physical instruments, but nevertheless bear some resemblance to traditional instruments. Examples are the Zendrum ZX [20] (a percussion instrument) or the Sonalog Gypsy MIDI [21], a performance instrument for controlling MIDI music through motion capturing.

The second group is the “Music-processing” group, which includes any device that deals with music as a piece of information, producing some sort of music as an output. One typical example is a sequencer, which is able to record musical events (a sequencer assigns a timestamp to each event and stores it into a file and vice versa, being able to reproduce previously recorded time-stamped events). A score editor can be regarded as a kind of sequencer with

graphical abilities that shows a graphic representation of a score in conjunction with additional information. Examples of score editors are MakeMusic Finale [22], Sibelius [23], or GVOX Encore [24].

The second group also includes devices that are able to compose music by following several composition rules. During the Baroque period the keyboard player was given a score with a bass line with certain numbers annotated with the objective of completing the given line with additional notes, respecting the harmony [25]. Nowadays, technology allows us to build a harmonizing device able to complete the bass line automatically in the same way live musicians used to.

Finally, there exists a third group known as “synthesizers,” which includes any device that transforms music information into something that can be perceived by one or more human senses. The most obvious example is music synthesizers: devices that generate audible signals as a response to MIDI events. Light synthesizers, firework firing systems, and water machines fall into this category as well.

Of course, all these categories are not mutually exclusive: in practice most devices implement characteristics of two or even three of the categories mentioned.

3.2.2. Wireless MIDI Devices. In this section, the features of some of the most relevant wireless devices currently on the market are analyzed in order to determine whether they are able to satisfy the requirements for constituting a musical collaborative wireless network such as that proposed in this article.

The first of these is the MIDIJet Pro Wireless MIDI [26], a low-latency 2.4 GHz wireless transmission system with a range of up to 20 meters. With a set of alkaline batteries it lasts about 30 hours, giving it excellent autonomy, but it does not fulfill two of the requirements mentioned in Section 2 (3 and 7).

- (i) It can only work as a transmitter or as a receiver, but not at the same time.
- (ii) It only admits up to 31 transmitter/receiver pairs at the same time, so would be inappropriate for use with large orchestras.

Furthermore, the device is quite bulky, making it hard to use with certain instruments.

Other examples of 2.4 GHz transmitters are the M-Audio MidAir Wireless MIDI Interface [27] and the CME WIDI-X8 Wireless MIDI System/USB Interface [28]. The M-Audio MidAir Wireless MIDI Interface works in half-duplex mode and has a range of up to 10 meters, which is quite small compared to the range achieved with the CME WIDI-X8 Wireless MIDI System/USB Interface (up to 80 meters with LOS (line of sight)). The latter system is only powered by two AA batteries, and allows full-duplex communications with up to 64 MIDI channels. There is another version of the CME transceiver, the CME WIDI XU Wireless MIDI Interface [29], which has been designed to work with computers acting as a wireless USB-MIDI adapter. The main drawback of these three transceivers is that they are limited to acting as

a mere wireless replacement of a one-to-one connection, so there would be no possibility of collaborative work among multiple devices (requirement 3 in Section 2). Moreover, such devices are not able to extend the network coverage using mesh techniques (requirement 6).

M-Audio also sells the M-Audio Mid Air 25 Midi Wireless 25-Key Keyboard [30], a small music keyboard MIDI Controller with an embedded wireless transmitter. This product offers basic MIDI-editing features, although its high current consumption (6 AA-batteries that last an average of 2 hours) is a significant limitation.

Another commercial system worth mentioning is the MIDiStream Wireless MIDI System [31], a specialized UHF (ultra high frequency) communications module for transmitting from a musical instrument to another device, reaching a maximum distance of 80 meters outdoors and about 30 meters indoors. The transmitter is relatively small (the size of a pack of cigarettes) and is powered by a 9V battery. Unfortunately, this module acts as a replacement of one-to-one MIDI communication, and thus collaborative intercommunications among multiple devices are not possible (so it therefore fails to fulfill requirement 3 of Section 2). Moreover, this device does not offer mesh network capabilities, so it is not possible to extend coverage through this mechanism (requirement 6).

A comparison of the most relevant features of the interfaces previously described is given in Figure 14, which also shows the main characteristics of our prototype (detailed in Section 4).

It is important to note that none of the wireless MIDI devices studied allows one-to-multiple node communications, as they all have been designed as transparent replacements of the original MIDI one-to-one communications. Only WIDI XV-8 permits a restricted one-to-multiple communications, allowing the user to switch among different slaves by pressing a button. Observing such lack of support for collaborative musical activities, we decided to implement our own prototype for performing the required one-to-multiple node communications. This prototype is called wi^m , making reference to “wireless musician,” and can be defined in a nutshell as a WSN-based MIDI interface.

4. Architecture Description

4.1. Architecture Description. Figure 4 presents the architecture of a musical collaborative wireless musical network (MCWN), a specific WSN especially designed to provide optimized communications between musical devices, thereby enabling collaborative activities among multiple participants. Our definition of collaborative activity includes any association between two or more devices, regardless of its complexity. For instance, a communication between a MIDI controller and a score editor is a good example of a simple collaboration. More complex applications are described in Section 5.

A star topology seems to be the most appropriate for musical applications since they typically use one-to-multiple (e.g., one keyboard to multiple synthesizers) or multiple-to-one (e.g., multiple instruments and one sequencer)

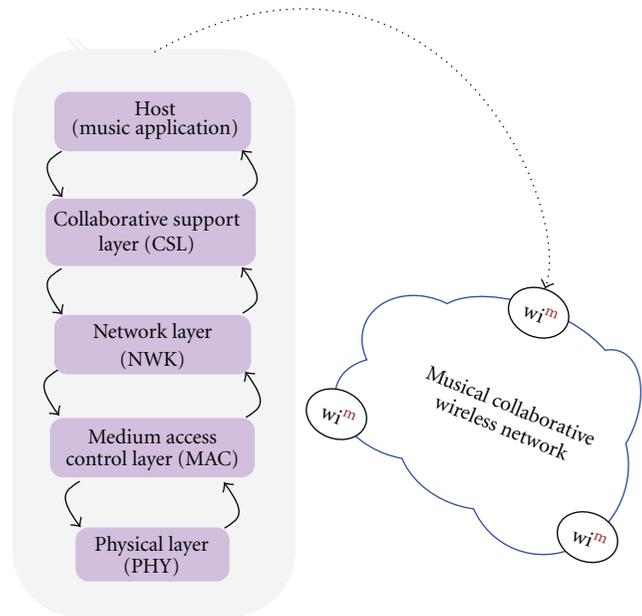


FIGURE 4: Collaborative wireless musical network.

communications schemes. This poses no problem for most WSN transceivers, because star and mesh topologies are in general available by default. For instance, the IEEE 802.15.4 standard [32] supports the star topology, whilst ZigBee [33] supports the mesh topology.

Our MCWN network is composed of two types of nodes.

- (i) wi^m -conductor, which coordinates the network and establishes communications with most nodes within the network.
- (ii) wi^m -instrument player, which joins the network and establishes communications with other nodes within the network.

The terminology we use is inherited from the field of WSN, where a central node (coordinator) is able to communicate with all the other nodes that form the network (end devices). It is useful to distinguish between both types of node in order to improve wireless communications performance. Also note that, at this point, the terms master and slave make no sense since one node can freely initiate communications with any other node.

Figure 4 also shows the complete network stack after adding a new network layer called collaborative support layer (CSL) at the top of a typical WSN stack. This special layer deals with device identification, communication, and association and, therefore, the ability to perform collaborative activities. To enable all these features a software library has to be developed. Such a library has to offer a list with several device categories (score editing, sound generating, etc.) and the corresponding set of operations each device is able to perform. The idea is similar to the ZDO Library used in ZigBee [33], which provides an easy configuration of wireless devices. In addition, devices could be programmed to behave in a smart way to help users detect hardware malfunctions or configuration errors (e.g., an incorrect MIDI channel setup).

The CSL is an extra layer explicitly added by us since it is specific for the application and, therefore, is not included in the default WSN stack. The CSL layer is placed between the network layer (NWK) and the host application. Ideally, communications with the host application are transparent, regardless of being connected to an MCWN network. Thus, it is possible to reuse existing designs with minor changes.

The operation of the network from a node perspective is as follows.

- (1) Network connection. The musical device is switched on and joins the wireless network.
- (2) Musical device discovery. The musical device searches for other musical devices within the network.
- (3) Definition of collaborative activities. The musical device determines which collaborative activities are feasible depending on the potential collaborators found in step 2.
- (4) The device asks the user which collaborative activity it would like to perform.
- (5) The device associates with one or several musical devices within the musical network to perform the collaborative activity.
- (6) The collaborative activity is performed.
- (7) When the collaborative activity ends, go back to step 2.

4.2. Collaboration Paradigm. Collaborative activities among different musical devices are not usually offered as a feature by MIDI devices due to the previously mentioned limitations in MIDI communications. Section 5 shows the benefits of performing such collaborations in several real scenarios. In contrast, this section presents a new musical instrument based on the concept of collaboration and WSN, which could be regarded as a paradigmatic application of the ideas expressed in this article.

The instrument uses concepts of location algorithms and modular synthesis. The idea consists in deploying a set of simple nodes, which collaborate with other nearby nodes to play music in real time. The performers could be, for example, visitors to a science museum or a contemporary art museum. Visitors would move the nodes within the exposition room, thereby changing the associations between the nodes. Also, the nodes could include some sort of basic controllers (buttons, sliders, touch screens, etc.) in order to allow real time control of synthesis parameters, for example, general *tempo*, filter depth, vibrato amplitude, or just note frequency.

Figure 5 shows a simple deployment of such an instrument. For the sake of simplicity, only three different types of node are used. The red node is the general input of the whole network and is the only node that cannot be moved by the visitors. The other two types are as follows.

- (i) *Wave-node*, which performs a note selected by another node. The waveform can be customized and, thus, the resulting sound will have a different timbre.

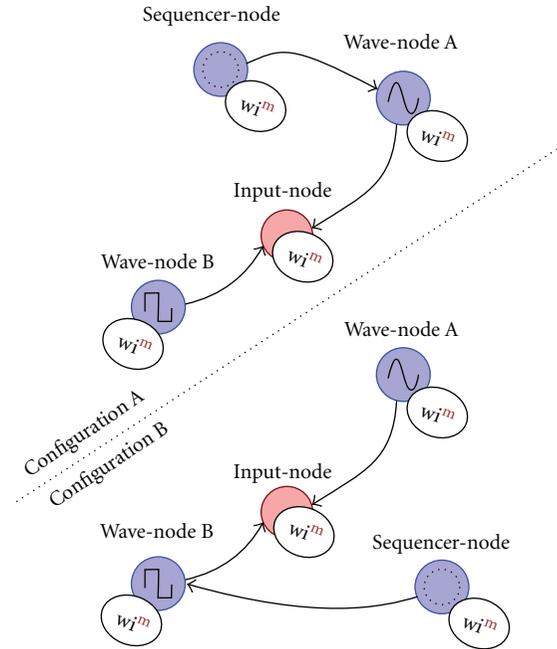


FIGURE 5: WSN-based collaborative musical instrument.

- (ii) *Sequencer-node*, which generates a random sequence of notes as output, with different duration and pitches.

When a *sequencer-node* associates with a *wave-node*, the notes generated by the *sequencer-node* are played in real time with the sound configuration selected at that moment in the *wave-node*. The association between nodes only occurs when visitors move two nodes and thus reduce the distance by up to half a meter, for example. Distances can be estimated by using the correlation between distance and the received signal power.

Figure 5 shows a scenario where the sequencer is moved from one point to another, so the sequencer-node is first associated with wave-node A, and then with wave-node B. Visitors can notice the change of association easily, as the notes being played have changed in timbre. Of course, more advanced types and interactions can be implemented, but the general idea of collaboration is still valid.

In [34] four scenarios are shown where WSN-based creative art was previously exhibited in two museums, namely, the National Museum of History and National Palace Museum, both in Taiwan.

- (i) *Smart Museum* [35]. It uses a ZigBee WSN tag-based system to identify museum visitors (age, sex, day of the visit, etc.), adapting the content of the presentations in order to encourage learning by challenging them with interactive games. Also, the data collected by the WSN network are used to grow an “e-flower,” and thus make the visitor part of the artwork.
- (ii) *One Million Heart Beats* [36]. As in the case of the Smart Museum, visitors are given a ZigBee tag, which also acted as a game controller. In the first phase,



FIGURE 6: Prototype design.

TABLE 1: Power consumption test.

Node state	Consumption (mA)
Sleeping	21.10
Awake	
Idle	69.80
Transmitting	109.80

visitors took control of virtual sperm fertilizing ova; the sex, blood type, and Chinese astrological birth sign was determined based on the combined action of multiple participants. In the second phase, users held a bottle with an embedded WSN node that collected a heartbeat for the growing fetus. The final resolution of the artwork is determined after one million heartbeats are collected.

- (iii) *Interactive WSN-bar* [37]. A WSN network receives environmental data from outdoor sensors: brightness, temperature, and CO₂ density. This information is used to make interactive flowers on a bar. Also, a WSN locating system detects the participant's movement, being used to animate the flight of butterflies among different bushes, according to the participant's moving between different rooms.
- (iv) *iFurniture WSN in a community* [38]. In this exhibition there are three types of furniture: *iBoxes*, *iChairs* and *iTables*. Each user that joins the artwork holds an *iBox*, which is a tag that stores his or her profile (gender, personal hobbies etc.) and mood. When the user sits on the *iChair*, the *iBox* transmits his or her profile and mood, so the *iChair* displays LED colors and images matching the user's information. When people gathered together in the same place have common interests, the *iTable* not only displays a visual representation, but also plays music, allowing people to meet and start a conversation.

5. Prototype

Figure 6 shows the prototype we have built to test the WSN-based collaborative paradigm proposed in this paper. This patent-pending design [39] is based on the popular

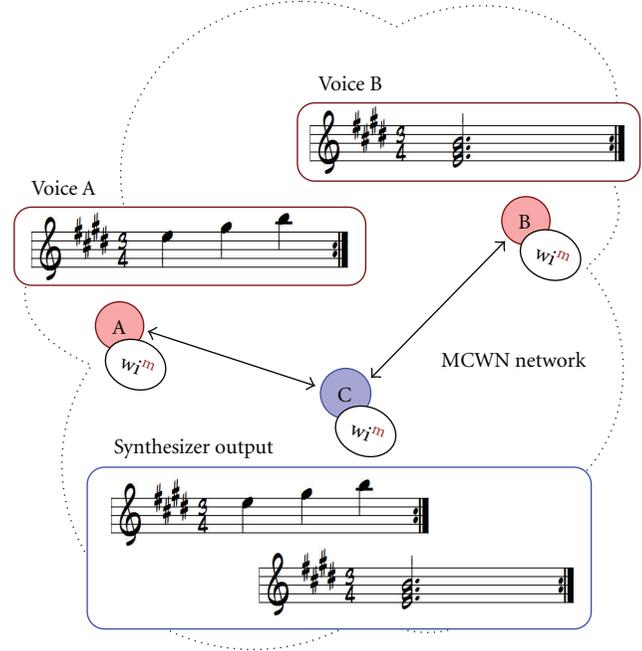


FIGURE 7: Multiuser test.

Arduino board [40], uses a Digi XBee ZigBee module [41] and has three types of connectors: MIDI-in, MIDI-out, and USB. Additionally, a special serial-to-MIDI driver was developed in order to connect the prototype to a PC that ran professional music software. The prototype was used to perform different tests.

- (i) *Multiuser tests*. The implemented scheme is showed in Figure 7, where a small MCWN network was deployed. All nodes are Arduino-based with an XBee module. Nodes A and B played the role of musicians, each playing two different musical excerpts using the MIDI protocol, making it necessary to develop a small program. On the other hand, node C was configured as w_j^m -conductor, which, in this example, is responsible for collecting all the notes transmitted by nodes A and B and redirecting them to a software synthesizer. As each node transmits its notes using different channels, different synthesizer instruments can be configured. In our test, voice A was assigned to a violin, and voice B was assigned to a piano. The audio output from the synthesizer was very easy to verify, as each voice was selected to be easily recognizable (voice A is a broken chord, whilst voice B is a static chord). It is interesting to note that as there is no synchronization mechanism (nodes start playing as soon as they are switched on), the resulting output is also not synchronized.
- (ii) *Power consumption tests*. Table 1 shows prototype consumption, measured with a multimeter. As can be seen, power consumption depends on the node state, for which there are three different possibilities: sleeping, awake and idle, and awake and transmitting. It is worth mentioning that the consumption values

TABLE 2: Signal loss in free space.

Scenario	Mean attenuation (dB)
50 cm	0.00
1 m	8.16
2 m	11.65
4 m	19.91
8 m	23.93
11 m	29.61

TABLE 3: Signal loss with different obstacles.

Scenario	Mean attenuation (dB)
Window (open metallic blinds)	1.04
Window (closed metallic blinds)	3.95
Wall with open door	0.39
Wall with closed door	1.19
Brick wall	1.46
Between floors	13.08

shown can be easily reduced since some of the electronic components of the prototype are not strictly necessary. For instance, the design includes two processing units, an AVR microcontroller at the Arduino board [40], and an EM250 microcontroller at the XBee ZigBee transceiver [41].

Given the above results, battery duration can be easily calculated. For instance, in the case of a 1,200 mAh lithium battery of 9 V, node autonomy is 10.93 hours in the worst case. It is important to note that power consumption in the awake and transmitting state is not constant: each transmission causes a peak of power consumption that lasts a very short time, thereby extending the calculated autonomy. Also, if an application does not need a continuous stream of music data, it is possible to allow the node to enter the sleep state, thus saving even more energy.

- (i) *Coverage tests.* This set of tests consisted of detecting the mean RSSI (received signal strength indicator) value obtained during the reception of 100 messages. Each measure was averaged five times for each experiment, to counteract the signal fluctuations caused by indoor fading. Both modules transmitted with a power of 3 dBm and *boost* mode enabled [41]. The results obtained are shown in Tables 2 and 3. Note that the total attenuation is the sum of the losses due to free space propagation and the existing obstacles [42].
- (ii) *Latency tests.* Latency in music systems is defined as the amount of time elapsed between the emission of a note execution request and the actual execution of such note. Latency is especially important in activities that demand real-time responses, such as live concerts. In these tests, we have focused mainly on analyzing the latency caused by wireless communications (i.e., the total communication time) when sending MIDI messages of different sizes. Although

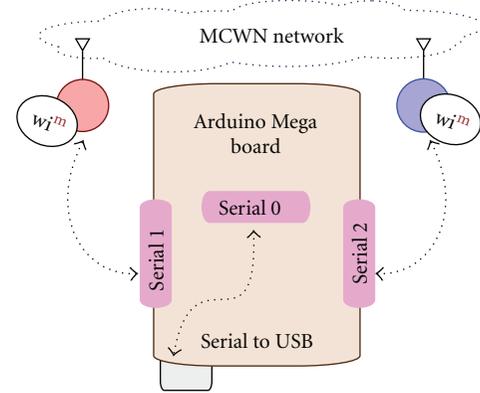


FIGURE 8: Latency test.

TABLE 4: Total latency values measured (1 byte).

Scenario	Mean (ms)	Variance
Relay-to-coordinator		
Test 1	13.09	1.15
Test 2	12.67	0.39
Coordinator-to-relay		
Test 1	12.40	25.36
Test 2	12.38	0.40
Total mean	12.63	6.88

the measurements performed are significant, we should point out that further study is required in order to take into account all the factors that may influence the data transmission.

Figure 8 represents the testing scenario for the latency tests. An Arduino Mega board [43] (which is completely compatible with the Arduino Diecimila we used for the prototype) was used because it facilitates the measurement process: it has four serial interfaces, more than enough to interconnect the two XBee modules needed to perform the latency experiment. A USB connection was used to allow wireless data to be sent from and received by a computer, where a Java-based application was run to measure the communications latency. The results obtained are shown in Tables 4 and 5. Depending on the test the communication direction varied (from a ZigBee relay to a ZigBee coordinator, and vice versa), as did the number of bytes transmitted (from one byte to three bytes, the latter being the typical length of a MIDI message). The distance between the two nodes was one meter, the wireless modules were configured in AT mode and each test consisted of sending 100,000 messages of one of the two lengths specified.

One of our concerns was that the underlying WSN protocol could introduce timing differences, as ZigBee was not initially thought to be a real-time protocol, but the results show that there are no significant differences between the measurements in both communication directions. Values are very stable, although sometimes there is a large delay in certain measures, thus causing a high variance. The cause of this behavior is being studied by the authors, and may be

TABLE 5: Total latency values measured (3 bytes).

Scenario	Mean (ms)	Variance
Relay to coordinator		
Test 1	20.64	0.53
Test 2	21.01	81.10
Coordinator to relay		
Test 1	20.89	54.88
Test 2	20.58	53.27
Total mean	20.78	47.48

motivated by external factors, such as intentionally delayed transmissions as a part of optimization techniques, or even issues related with Java timing management.

Finally, although latency values are not very high for simple applications, they would definitely have to be lowered when working in more complex scenarios. To optimize such latency, it is important to take into account the configurations available in the XBee modules and the inner workings of the ZigBee protocol. Similarly, it is important to note that the results shown have been obtained without performing any optimization, as this was not their initial objective.

6. Applications

In this section five different applications for the proposed scheme are discussed.

- (i) Score management: the w_i^m interface is embedded into an interactive iPad-like screen, which allows musicians to make annotations in their orchestral scores and to share them with their colleagues. Each musician and the conductor would have a device placed on their music stand.
- (ii) Stage rigging: a series of multimedia devices (lights, smoke machines, etc.) placed on a stage communicate with each other through the w_i^m interface in order to be synchronized with the music played. In this case, the lack of cables and automatic configuration is particularly interesting, because the stage setup requires a lot of manual work.
- (iii) Computer-aided learning: technology can help music students to improve their musical skills by means of electronic musical instruments. Sensors can also be useful in fields where traditional education is not very effective, such as music theory, harmony, counterpoint, ear training, and certain aspects of instrument playing.
- (iv) Computer-aided score edition: this application is aimed at digitalizing manuscript scores. The notes can be obtained in real-time from a group of musicians, speeding up the score edition process (the inputting of notes using a keyboard and mouse is a really time-consuming task).



FIGURE 9: Score management in musical ensembles.

- (v) Another application is Virtual Musician, where we would build up a musician that is equivalent in functions to a human musician, but with the advantage of being available any time, without agenda restrictions. It can play with other human musicians, adapting its performance as required by the musical discourse (tempo, dynamics, etc.).

6.1. Score Management in Musical Ensembles. One typical problem in large musical ensembles is the management of musical scores. Wireless nodes can collaborate among themselves to facilitate the management of such musical scores. This idea is represented in Figure 9, where an interactive device, like an iPad, is placed on each musician's stand and on that of the conductor. Such devices integrate the w_i^m module, so they would connect to the same MCWN. Apart from the logical advantages of using electronic versions of the scores, collaboration between the different devices may ease common tasks.

- (i) Score annotation exchanged between musicians. For example, violins must follow the bow indications marked by the leader. Thus, the leader would write the bowing in the electronic score and would share it with the rest of the violinists.
- (ii) Individual score annotations. Each music stand can save annotations that, although having a personal meaning for a particular musician, may be shared with other colleagues for their personal use. For example, a flautist could add a "more piano" entry to a given passage, because he or she considers that it is important to not to break the sound balance with the other sections. These annotations could be shared with other musicians within the orchestra, or even with the flute students in their charge.
- (iii) General score annotations. They are particularly useful to the conductor, as he or she can mark rehearsal sections, *tempos*, and dynamics relating to the whole orchestra directly on to the score. The collaborative network will take responsibility for announcing those annotations.

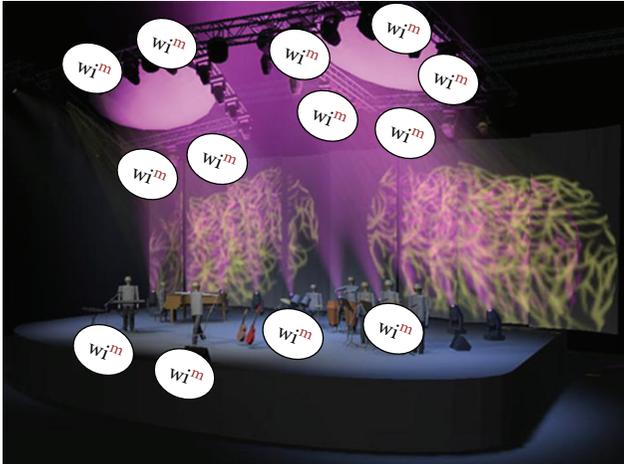


FIGURE 10: Stage rigging (The software shown is Capture Polar [44]).

- (iv) Additional features. As the score is in digital format, it is possible to offer additional features that may help the musicians or the conductor in several situations. For instance, a musical excerpt could be displayed in different clefs, which is very useful when working with transposing instruments. Transposing instruments are those for which the written and played sound does not match, as happens with the B-flat clarinet, for instance, which sounds one tone lower than written. Visual transposition is especially useful for conductors, music arrangers and composers, and sometimes even for instrument players.

6.2. Stage Rigging. Figure 10 shows an MCWN network where several multimedia devices (stage lights, lasers, smoke machines, etc.) are interconnected in a live performance. In this scenario, it is necessary to synchronize a large number of multimedia devices in order to make them “dance” with the music, which means that even musical instruments can join the MCWN network in order to set the main tempo (e.g., guitar, bass guitar, or battery). It is particularly easy to determine the tempo by monitoring percussion instruments.

In this specific application, it is important to emphasize the advantage of using wireless interfaces, as it greatly facilitates the assembling/disassembling of the orchestra stage, which can be repeated many times if the orchestra is on tour. In addition to making it easier for stagehands to deploy the multimedia network deployment, configuration is much simpler because devices can self-configure and even report abnormal situations to the technicians. Finally, musicians also have greater freedom of movement on stage, thus benefiting the show.

6.3. Computer-Aided Learning. Computer-aided learning can be greatly benefited by the close relationship between music devices and computers, which leads to a wide range of possibilities for interaction between students and

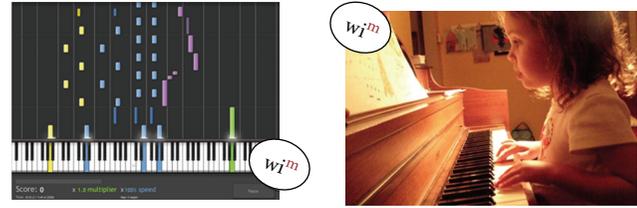


FIGURE 11: Computer-aided learning (The software shown is Synthesia [45] (left) The pictures shown in are under Creative Commons Attribution license. Their author is woodleywonderworks (right)).

smart virtual teachers (SVT). One advantage of computer-aided learning when compared to traditional systems is the possibility of a more effective training in practical subjects such as music theory, harmony, counterpoint, ear training, or instrument playing. For instance, there are pianos with a special circuit capable of emitting light in the correct keys to guide students, helping them to play difficult passages. Such a piano could also be able to interact with software that would challenge the student to perform short musical excerpts, as if it was a video game (shown in Figure 11).

6.4. Collaborative Computer-Aided Score Transcription. The transcription of music scores to electronic versions using keyboard and mouse is a very slow process due to the abstract nature of music notation. Figure 12 proposes a scenario where an MCWN network is used to collect notes played by a string ensemble using pickups and sensors capable of detecting the music performance. All those notes are sent to music transcription software, which is able to translate the musical performance to an electronic score, with a certain error rate. This collaborative activity enormously accelerates the transcription of musical scores. Score editing software like MakeMusic Finale [22], Sibelius [23], and GVOX Encore [24] could easily integrate music transcription through a collaborative network, saving the musical data using their own score formats.

6.5. Smart Virtual Musician. Technology allows us to have a virtual substitute for a musician, either for individual rehearsals, or even for public concerts. That is the idea represented in Figure 13, where a virtual musician plays a trumpet *concerto* with a human pianist. The idea of using a robot as a metaphoric representation of a synthesizer is very interesting, since we visualize a synthesizer as a smart musician that can be asked to play any instrument.

The advantages of using virtual musicians are clear for anyone accustomed to the scheduling problems of musicians. For instance, music students often have agenda issues, even in small ensembles (duets, quartets, etc.) and sometimes parts of orchestras are unbalanced due to the lack of musicians of a certain instrument. These problems can be partially solved by using smart virtual musicians (SVM), who can play any musical instrument and are available at any time of day. Although the use in professional environments requires the work of skilled technicians, their use for group music

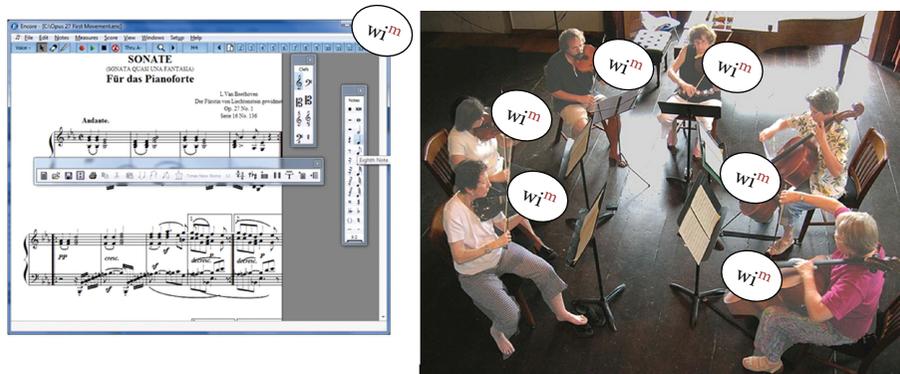


FIGURE 12: Collaborative computer-aided score transcription (The software shown GVOX Encore [24] (left). The pictures shown in are under Creative Commons Attribution license. Their author is Ravpapa (right)).

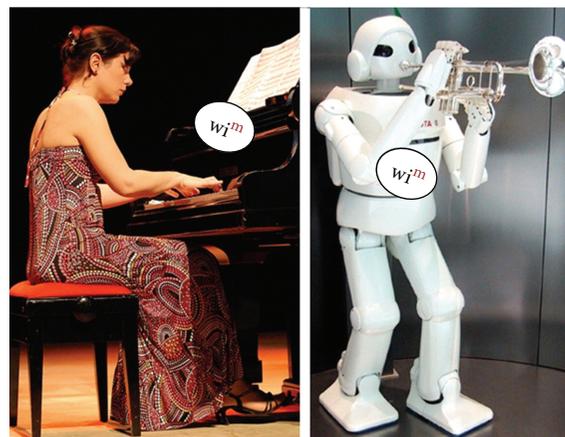


FIGURE 13: Smart virtual musician (The pictures shown in are under Creative Commons Attribution license. Their authors are Fundação Cultural de Curitiba (left) and Chris 73 (right)).



	MIDIjet Pro	MIDI air	MIDI stream	WIDI XV-8	WiM
Frequency band	2.4GHz	2.4GHz	400 MHz and 868/870 MHz	2.4 GHz	2.4GHz
Maximum communication range	150 m	9m	30 m indoor, 80 m outdoor	80 m outdoor	40 m indoor, 120 m outdoor
Communication direction	Unidirectional	Unidirectional	Unidirectional	Bidirectional	Bidirectional
MIDI channels	30	*	*	32	Unlimited
Communication channel selection	Manual	Automatic	One channel available	Automatic	Automatic
Latency	Low	Low	Low	*	Optimized
Coverage extension with relays	Yes, generic	Yes, generic	No	Yes, generic	Yes, optimized
Collaborative activity support	No	No	No	No	Yes

FIGURE 14: Devices proposed to create the prototype (*no information given by the manufacturer).

rehearsal is possible and relatively easy. The benefits of SVM musicians are clear, as they serve to get an overview of the general composition with no human musicians, thus saving time.

7. Conclusions and Future Work

This article has studied the possibility of using WSN as the technological basis for interconnecting various MIDI-compatible musical devices. WSN topologies make it possible to increase the number of communication links and enable musical collaborations to take place among different MIDI devices. Such collaborations lend themselves to activities related with self-identification and self-configuration, or even to exploring new ways of artistic expression. In addition, wireless communications facilitate the connection of MIDI devices with other multimedia protocols, such as DMX512, making it possible to achieve a more complex level of integration between lights and music. The proposed architecture offers the possibility of developing more advanced applications for smart musical devices. A number of the many applications possible have been put forward: score management, stage rigging configuration, computer-aided learning, score transcription, and the use of a virtual musician.

Moreover, it has been observed that, despite the benefits that can be obtained from collaborative activities in the music field, none of the musical interfaces studied is valid for this purpose. As a result we have developed a preliminary prototype based on the popular Arduino platform and which makes use of ZigBee communications. Our prototype was aimed at testing some of the ideas described in this article, but it was also valid for discovering the technical issues relating to their real-life implementation. In this way, we performed multiple-user, power consumption, coverage, and latency tests.

It has been shown that it is necessary to optimize wireless communications in order to decrease latency, probably by simplifying the protocols of the lower layers of the ZigBee stack. The study of such optimizations can constitute an interesting topic for further research.

Further interesting future work consists in the implementation of a demonstration platform, which can be used as a paradigm of the concepts described in this article. Such a platform could be designed as a collaborative sensor-based musical instrument with indoor location features that could be used in contemporary art and science museums.

Finally, another interesting area for future research is that relating to the development of distributed algorithms that could be used in MCWN-based collaborations among musical devices like those described in this article.

Acknowledgments

This paper has been supported by Xunta de Galicia (10TIC003CT) and Ministerio de Ciencia e Innovación of Spain with FEDER funds of the European Union (IPT-020000-2010-35).

References

- [1] B. Katz, *Mastering Audio. The Art and the Science*, Focal Press, 2nd edition, 2007.
- [2] A. Pejrolo and R. DeRosa, *Acoustic and MIDI Orchestration for the Contemporary Composer*, Focal Press, 1st edition, 2007.
- [3] J. Huntington, *Control Systems for Live Entertainment*, Focal Press, 3rd edition, 2007.
- [4] M. Pukkete, *The Theory and Technique of Electronic Music*, World Scientific, 2011.
- [5] P. Cuadra, A. Master, and C. Sapp, "Efficient pitch detection techniques for interactive music," in *Proceedings of the International Computer Music Conference*, pp. 403–406, Havana, Cuba, 2001.
- [6] E. H. Callaway, *Wireless Sensor Networks: Architectures and Protocols*, Auerbach, 2003.
- [7] MMA Association, <http://www.midi.org/>.
- [8] M. Wright, A. Freed, and A. Momeni, "Open sound control: state of the art 2003," in *Proceedings of the Conference on New Interfaces for Musical Expression (NIME '03)*, Montreal, Canada, 2003.
- [9] Yamaha MLan, http://www.mlancentral.com/mlan.info/mlan_ppf.php.
- [10] HD Protocol, 2011, <http://www.midi.org/aboutus/news/hd.php>.
- [11] Entertainment Technology—USITT DMX512-A—Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories, <http://webstore.ansi.org/RecordDetail.aspx?sku=ANSI+E1.11-2008>.
- [12] MIDI Media Adaptation Layer for IEEE-1394 (Version 1.0), MMA Association, [http://www.midi.org/techspecs/rp27v10-spec\(1394\).pdf](http://www.midi.org/techspecs/rp27v10-spec(1394).pdf).
- [13] IEEE P1639, <http://dmidi.l4l.ie/index.html>.
- [14] IETF RFC 4694: RTP Payload Format for MIDI, <http://tools.ietf.org/html/rfc4695>.
- [15] MIDI Control of Stage Lighting, <http://www.innovateshow-controls.com/support/downloads/midi-dmx.pdf>.
- [16] Yamaha WX-5 Owner's Manual, <http://www2.yamaha.co.jp/manual/pdf/emi/english/synth/WX5E.PDF>.
- [17] AKAI EW14000S Electric Wind Instrument Reference Manual (Revision D), http://www.akaipro.com/stuff/contentmgr/files/0/6b6cf69a9363e7bb0d784d61a46813646/file/ewi4000s_refmanual_revD.pdf.
- [18] Steiner MIDI EVI Owner's Manual, <http://www.patchmanmusic.com/MidiEviManualV111.pdf>.
- [19] Morrison Digital Trumpet Owner's Manual, <http://www.digitaltrumpet.com.au/MDTManual.pdf>.
- [20] Zendrum ZX Digital MIDI Controller Owner's Manual, http://www.zendrum.com/pdf/manual_01.pdf.
- [21] Sonalog Gypsy MIDI—Motion Capture MIDI Controller Suit, <http://www.soundonsound.com/sos/oct06/articles/sonalog.htm>.
- [22] MakeMusic Finale Music Notation Software, <http://www.finalmusic.com/>.
- [23] Sibelius Music Notation Software, <http://www.sibelius.com/>.
- [24] GVOX Encore Composition Software, <http://www.gvox.com/encore.php>.
- [25] W. Piston, *Harmony*, W. W. Norton & Company, 5th edition, 1987.
- [26] MIDIJet PRO Wireless MIDI Manual (Revision 2), http://www.organworks.com/Content/Downloads/MIDI_Products_Specs/MIDIjet%20Pro%20Documentation.002.pdf.

- [27] MidAir Wireless USB MIDI System User Manual, http://www.m-audio.com/images/global/manuals/061114_MIDAIR-SA_UG_EN01.pdf-SA_UG_EN01.pdf.
- [28] CME WIDI-X8 MIDI System User Manual, http://www.cme-pro.com/en/getfile.php?file_id=141.
- [29] CME WIDI-XU Wireless System Interface User Manual, http://www.cme-pro.com/en/getfile.php?file_id=165.
- [30] M-Audio Mid Air 25 Midi Wireles 25 Key Keyboard User Guide, http://www.m-audio.com/images/global/manuals/060614_MidAir_UG_EN01.pdf.
- [31] Kenton MidiStream MIDI System Operating Manual, <http://www.kentonuk.com/kmanualspdf/midistreamman.pdf>.
- [32] IEEE 802.15.4-2006, "IEEE Standard for Local and Metropolitan Area Networks: Specifications for Low-Rate Wireless Personal Area Networks," 2003.
- [33] ZigBee Standards Organization, "ZigBee 2007 Specification Q4/2007," <http://www.zigbee.org/Standards/ZigBeeSmartEnergy/Specification.aspx>.
- [34] S. Hsu and J. D. Tygar, "Wireless sensor networks: a building block for mass creativity and learning," in *Proceedings of the ACM Creativity & Cognition, Understanding the Creative Conversation Workshop*, October 2009.
- [35] S. Hsu, "Manipulating digital archives using digital art techniques," in *Proceedings of the 4th Digital Archive Conference*, pp. 71–78, 2005.
- [36] S. Hsu, J. Lin, C. Chen, Y. Chen, J. Lin, and K. Chang, "One million heartbeats," in *Proceedings of the ACM International Conference on Multimedia*, pp. 365–366, 2007.
- [37] J. Lin, S. Hsu, and Y. Chen, "Interactive WSN-Bar," in *Proceedings of the 1st International Conference on Arts and Technology (ArtsIT '09)*, Lecture Notes of the Institute for Computer Sciences, Springer, 2009.
- [38] S. Liu and S. Hsu, "iFurniture: application of wireless sensor network in community interactive furniture," in *Proceedings of the International Conference on Advanced Information Technology DVD*, 2009.
- [39] C. J. Escudero, T. M. Fernández, and S. J. Barro, "Módulo de Comunicación Inalámbrica para Dispositivos Musicales Electrónicos," Patent pending #P2010 31523, Spain, October 2010.
- [40] Arduino Duemilanove Board, Open-Source Electronics Prototyping, <http://www.arduino.cc/es/Main/ArduinoBoard-Duemilanove>.
- [41] XBee®/XBee-Pro® ZB OEM RF Modules Manual, ver. 11/15/2010, http://ftp1.digi.com/support/documentation/90-000976_G.pdf.
- [42] A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005.
- [43] Arduino Mega, Open-Source Electronics Prototyping, <http://arduino.cc/en/Main/ArduinoBoardMega>.
- [44] Capture Polar, lighting design software, <http://www.capture-sweden.com/>.
- [45] Synthesia, piano learning software, 2011, <http://www.synthesia-game.com/>.

Research Article

One-Time Broadcast Encryption Schemes in Distributed Sensor Networks

Pawel Szalachowski¹ and Zbigniew Kotulski^{1,2}

¹ Institute of Telecommunications, The Faculty of Electronics and Information Technology, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland

² Institute of Fundamental Technological Research of the Polish Academy Sciences, Pawinskiego 5B, 02-106 Warsaw, Poland

Correspondence should be addressed to Pawel Szalachowski, p.szalachowski@stud.elka.pw.edu.pl

Received 12 July 2011; Revised 8 December 2011; Accepted 10 December 2011

Academic Editor: Yuhang Yang

Copyright © 2012 P. Szalachowski and Z. Kotulski. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Broadcasting is a message-transferring method characteristic for majority of sensor networks. Broadcast encryption (BE) is broadcasting encrypted messages in such a way that only legitimate nodes of a network can decrypt them. It has many potential applications in distributed wireless sensor networks (WSNs) but perfect deploying of that method is very difficult. This is because of a WSN is a very dynamic network which includes nodes with limited computational, storage, and communication capabilities. Furthermore, an attacker in this environment is powerful. He can eavesdrop, modify, and inject messages or even capture a large number of nodes, so the solutions must be both secure and efficient. This paper describes several BE schemes from the point of view of WSNs. We present in details the schemes called *onetime*, and we show how these methods can be applied in distributed sensor networks. We mainly focus on data origin authentication and rekeying processes, crucial for security in such a hostile environment. An analysis and evaluations of proposed schemes are also provided.

1. Introduction

The popularity of WSNs is an effect of recent advances in communication and development of the microelectromechanical systems (MEMS) technology. Now we have low-cost, low-power, and small devices which are mainly used for sensing, measuring, gathering, and then transmitting data from distributed locations (or an environment) to some acquisition center. Potential applications of WSNs are military (surveillance, command control, reconnaissance, intelligence, etc.), environmental (flood or fire detection, pollution transport, hostile objects observation, etc.), medical (elder people or patients monitoring), and other home and commercial applications. Surveys on the sensor network technology and its applications are presented in, for example, [1–3]. WSNs need communication protocols with special properties. Such approaches must be self-organizing [4], self-healing, fault tolerant, and secure. It is difficult to achieve all these features for devices with as limited hardware resources as wireless sensors are. Therefore the protocols must be additionally lightweight.

Broadcast encryption is an ideal proposition for WSNs. Sending encrypted messages which can be decrypted only by a predefined group of nodes can be very helpful in a WSN. Additionally, with this method we can manage strictly selected nodes [5] or configure them. Furthermore, by BE, we can fix a group of nodes and equip them with a shared key limited to just this group. Since then participants of the group can communicate each other in a secure way. However, BE must be realized in a very efficient and secure way and this is the crucial problem of BE. There are BE schemes using symmetric and asymmetric (public key) cryptography; in this paper, we focus mainly on the symmetric cryptography approaches. As we already mentioned, a typical node in a WSN has very limited resources. A small battery and weak computation efficiency make usage of the public key cryptography (PKC) impracticable. Although there are many BE schemes based on PKC, in further considerations we will not focus on them in this paper. A lightweight PKC is an objective of many research papers, but its implementations are too slow for many applications, see for example, [6–8]. Furthermore, an imprudent application of PKC (or of

some other expensive method) can make a wireless network susceptible to denial of service (DoS) attacks. Another crucial resource is memory. Constructing a security protocol, we must minimize amount of a key-storage memory. Next hardware constraint is the radio bandwidth. Because of low-transmission power, messages sent and received should be as short as possible, which gives an advantage to the symmetric cryptography.

In contrast, adversaries in these environments are very powerful [9, 10]. Eavesdropping, modifying, replying, and injecting packets are very easy when radio waves are used as a medium. Moreover, we should assume that an adversary can physically capture a number of sensors or can take control over them. Hence, the BE schemes for WSNs should be resistant to these types of attacks or at least should discover them.

The rest of the paper is organized as follows. We define the BE (with related issues) and present some significant approaches in Section 2. In Section 3, we give detailed overview of the schemes called *one-time schemes*. Some improvements of these schemes we present in Section 4 and next we analyze them in Section 5. Finally, in Section 6 we give conclusions and propose future research.

2. Broadcast Encryption

BE is some special class of key distribution schemes, which belongs to the conference key distribution protocols. Its goal is to allow a broadcast center (BC) to distribute an encrypted content to an arbitrary dynamically changing set of destination nodes. Only these nodes are able to decrypt messages. Let U be a set of all nodes in a WSN, T be a set of privileged (destination) nodes, and $S = U - T$ be a set of unprivileged nodes. Each member of U is equipped with a secret key (or, in some protocols, with several secret keys) shared with the BC. This key is called the preshared key (PSK). Further in this paper, we use the terms “node” and “user” interchangeably. They both denote a regular sensor in the network.

The BC in a BE scheme uses two types of messages that are sent together. They are a *header* and a *ciphertext*. A transmission of these messages is called a *session*. The *ciphertext* is a message encrypted with the *session key* (SK) K_e , while the *header* provides information which is necessary for the privileged nodes to obtain the key K_e . The BC sends a communication package, *msg*,

$$msg = \langle \langle header \rangle, E_{K_e}(M) \rangle, \quad (1)$$

where $E_{K_e}(M)$ denotes a *ciphertext*, that is, the message M encrypted with a symmetric cipher and the key K_e . For the node x , PSK_x denotes a set of preshared keys stored by x . The BE protocol requires some pre-defined function F which should enable the SK recovering only by the privileged nodes:

$$\begin{aligned} \forall x \in T : F(\langle header \rangle, PSK_x) &= K_e, \\ \forall x' \notin T : F(\langle header \rangle, PSK_{x'}) &\neq K_e \end{aligned} \quad (2)$$

and such that $F(\langle header \rangle, PSK_{x'})$ must not provide any information about K_e to the node x' and to an attacker.

In BE schemes, the length of the *header* is a transmission (communication) overhead, the time of $F(\langle header \rangle, PSK_x)$ calculation is a computation overhead and the size of PSK_x is a memory (storage) overhead in nodes' functioning. In further considerations, we assume that all cryptographic primitives used are secure and all secret keys are sufficiently strong that means that no adversary with limited computational resources is able to break a cryptographic primitive or exhaustively search a secret key space.

Analyzing security of BE, we need basic definitions and terms. Now we define some of them.

Resiliency. To the set S of a BE scheme means that even if an eavesdropper obtained all preshared keys of nodes from some set S then he can obtain no knowledge about a secret common to the member nodes of any other set of nodes T , for the sets such that $S, T \subseteq U, S \cap T = \emptyset$.

The scheme is called k -resilient if it is resilient to any set $S \subseteq U$ of size k .

Managing. It is adding, deleting, and revocation of users; it should be performed without a significant overhead.

Backward Secrecy. It is a property of BE which ensures that newly added users (to the set T) are not able to decrypt previous broadcast content.

Forward Secrecy. It means that when a user is removed from the privileged set T , then he is not able to decrypt later broadcast content.

Traitor Tracing. It is a mechanism for identifying traitor users who gave keys (or decrypted ciphertext) to unprivileged users.

BE systems can be classified as *stateful* and *stateless* schemes. The stateful approach requires that users are always connected to a broadcast center. The BC is used for keys update. Users in stateless schemes cannot update their keys, so a permanent connection with the BC is not required.

In a distributed WSN environment we need an efficient and secure communication protocol. Since nodes are vulnerable to malicious tampering, the BE approach must be resilient or at least k -resilient for high k . The network management should be very efficient from sensor resources point of view. We must note that in a typical network the nodes are not designed for computing but rather for data transmitting [6]. Thus, to decrease energy consumption, the right strategy is to perform *heavy* computations at the BC. Wireless Sensor Networks are dynamic by nature, so the *backward* and *forward secrecy* must be provided by a BE scheme designed for WSNs. Broadcast encryption is also often used in digital rights management (DRM) systems. For these applications (pay TV, DVD protections, etc.), the traitor tracing feature is very helpful to counteract a copyright piracy. This problem has been addressed in past years in many papers, see for example, [11–15]. Full tracing traitors schemes seem to be too exhaustive for WSNs, which are low cost by nature. Nevertheless, in our review, of BE we treat this function as an advantage of the schemes.

2.1. Related Approaches. The broadcast encryption problem is connected with group key agreement protocols, multicast security, and other constructs designed for secure group

communication. Some of these methods can be used to solve the BE problem in sensor networks.

In literature there are many decentralized schemes for key agreement in distributed sensor networks, see for example, [16–18]. They play a similar role as BE but they work in a quite different way, because these protocols' objective is to agree on a key between two nodes (or, eventually, a larger group of nodes) and to achieve this a Broadcast Center is not deployed. Usually, this class of schemes consists of the following phases: *pre-distribution*, *key discovery* and *path/channel establishment*. *Pre-distribution* is realized during nodes' preparation before the network deployment, but the second and the third phases are realized when the network's nodes are active. Such protocols are ideal for self-organizing applications, but absence of a BC in a key discovery and the path/channel establishment phase may lead to security flaws. In such a case, a malicious attacker can abuse these phases and as a consequence, establish a fake path, revoke some legitimate nodes or perform DoS attack for example, by sending many *discovery messages*. Ramkumar in [19] described a BE scheme based on a random key pre-distribution. His solution is also decentralized and it does not need the PKC. In further considerations we will concentrate on a group key agreement driven by a broadcast center but the solutions based on the PKC are also noteworthy.

Other efficient constructs in related problems of key distribution are presented in [20]. Besides, introducing a novel authentication scheme, this paper presents a performance comparison of several authentication schemes (including PKC approaches). Brooks et al. in [21] introduced a special infrastructure for security in sensor networks. A network is partitioned into special regions with a chosen node as a key server. Further, within these regions a secure multicast communication is performed.

Papers [22, 23] present surveys of key management schemes in WSNs. The performance of selected schemes is also presented in [24]. A general conclusion from the above reviews is that no key distribution technique is ideal to all scenarios where sensor networks are used. A key distribution protocol selection must depend on a specific network's application, its resources, and characteristics.

2.2. The Broadcast Encryption Schemes. Let us assume for the rest of the paper that $n = |U|$ is a number of users in the whole network and r is a number of users that revoke cooperation. The first formal study of a BE problem was presented in the paper [25] where Fiat and Naor introduced several approaches with different properties. There are schemes that do not require a broadcast center to broadcast messages (they are called *Zero Message Schemes*). These protocols are 1-resilient. Later there were introduced some k -resilient approaches with a low-memory requirement. The most efficient protocol needs storing by each node $O(k \log k \log n)$ preshared keys and broadcasting by a BC $O(k^2 \log^2 k \log n)$ messages. In the paper [26] there is a survey of such related methods. Generally, the review papers focus on comparison of BE schemes in terms of a transmission overhead and a memory overhead. A computational overhead is only slightly remarked. However, in a sensors' case, the computation

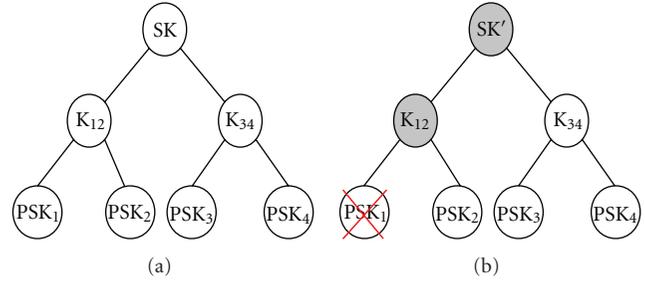


FIGURE 1: Tree-based broadcast encryption (forming and revocation processes).

efforts (and what it follows, energy consumption) is a crucial issue and must be thoroughly analyzed in each solution proposed.

The next class of protocols are *tree-based* schemes. This type of secure group communication was proposed independently in two papers. The paper [27] focuses on binary trees, while the paper [28] assumes the degree of the tree as a parameter. The keys are derived by a symmetric cipher and the properties of these protocols are similar. Each user needs to store $O(\log n)$ keys, and the protocol sends $O(\log n)$ (exactly $2 \log n$) messages to establish a new group key after a revocation. An example of the binary tree-based broadcast encryption is shown in Figure 1. Each node has its own PSK and it derives the upper keys with a neighbor using some pairwise key agreement mechanisms. A shared session key (SK) is produced as a root key. The revocation process is presented in the right-hand side of Figure 1. A wrong leaf is deleted, and all keys it possesses should be securely updated. The node with PSK₂ updates K₁₂ and SK; next, the nodes with PSK₃ and PSK₄ update only the session key. As we can see, the forming and deleting processes are very important in pairwise key agreement solutions. For such protocols, the communication overhead was reduced in [20, 29]. The improved protocols require only $\log n$ messages to send after a user revocation. Additionally, the methods base mainly on fast cryptographic constructs (hash functions or symmetric ciphers), so they should be applied in networks with limited capabilities. Other well-known, *tree-based* schemes are complete subtree (CS), subset difference (SD) [30], and a layered version of the Subset Difference (LSD) [31]. These schemes are very efficient. The CS method needs $O(\log n)$ PSKs for every node and its transmission cost is $O(r \log(n/r))$ messages sent. In the SD, each node needs to storage $O(\log^2 n)$ PSKs. Next a broadcast center forms a privileged subset by sending $O(r)$ short messages, which mark r revoked nodes. Later, each node has to execute $O(\log n)$ computations. The LSD protocol can achieve the same goal with $O(\log^{1+\epsilon} n)$ PSKs, $O(r)$ messages, and $O(\log n)$ computations.

Some attempts of reducing the nodes' key storage and eliminating the rekeying process are presented in [32]. Another hierarchical scheme that was designed to increase efficiency can be found in [33]. Daza in [34] presented a BE approach for mobile ad-hoc networks. In his method

the transmission overhead is low, but the approach bases on a secret-sharing scheme and the ElGamal-based PKC. The method is interesting but the application of PKC makes it too expensive for standard sensors. In paper [35] Yoo et al., basing on polynomial interpolations, increased the efficiency of the Naor-Pinkas scheme [14]. Their protocol requires $O(\log(n/m))$ PSKs and $O(\alpha r + m)$ messages, where m is the number of partitions of the users set (its choice influence the efficiency of the protocol) and $\alpha \in (1, 2)$ is a predetermined constant. Security of this scheme is provided by the Diffie-Hellman protocol, so deployment of that solution in a WSN may be inefficient.

Sometimes users (nodes of a network) are represented in an alternative way. For example, a novel approach where the set of privileged users is described by attributes, was introduced in [36]. Each node has a set of attributes and a decryption key depends on this set. The advantage of that proposition is that BC can revoke a group of the nodes (not only a single one) sending messages of a size linear with respect to the number of all attributes. It is realized by decryption and restriction of access policy, driven by AND/OR functions on attributes. This solution is resilient and the attributes make it easy to manage, but the challenge is to construct an efficient attribute-based encryption scheme. Another approach is the BE scheme proposed in [37], which operates on users' profiles. It is realized by introducing different types of broadcasts with corresponding probabilities. For each user his profile is created, which denotes probabilities that the given user will subscribe the given broadcast. Next, using profiles, the scheme can optimize some popular tree-based BE protocols. In particular, the solution can reduce a bandwidth required by the complete subtree and by the subset difference approaches. This reduction is significant when the scheme allows some unprivileged nodes to decrypt a content. A tradeoff between storage and communication in BE schemes is an important subject of research. The results of this aspect are presented in papers [38–40]. It is especially important for such environments like a WSN, because of nodes' limitations. A rational tradeoff between a number of preshared keys stored in each node and a volume of broadcast transmission is crucial especially for Wireless Sensor Networks.

3. The One-Time Schemes

The one-time schemes were chronologically the first solutions for the BE problem. We say that a protocol is *one time* if, while using this protocol, PSKs must be updated after every usage. Generally, this is treated as a disadvantage but we will show that the key update after a session period can provide us some additional security benefits. Moreover, such protocols are very efficient from the storage overhead point of view. In majority of *one-time* solutions, a node needs to store $O(1)$ PSKs that is a small number in many scenarios for protocols' efficiency. In Section 4, we will show how to improve the one-time schemes using different PSKs' derivation and authentication methods. Additionally, the one-time schemes can be easily transformed for example, into stateful tree-based schemes. The authentication of nodes

is a generic part of BE protocols. Now, we will describe two such schemes of authentication.

Chiou and chen in the paper [41] introduced methods for the BE using a secure lock. Their paper presents public-key and symmetric-key based broadcast protocols. The lock is constructed in such a way, that only a privileged node is able to open it. The construction is based on the Chinese Remainder Theorem (CRT). Each user belonging to T (a privileged node) adds its congruence equation to a set. The system of such equations must be solved by each node to obtain a common secret. The CRT is used as the solution algorithm. Unfortunately, the computational overhead of this scheme is acceptable only for very small groups of nodes, and it is definitely too expensive for sensors with limited capabilities.

Protocols presented by Berkovits in [42] belong to first authentication solutions for BE. The methods are based on two threshold secret sharing schemes: the Shamir's scheme [43] and the Brickell's scheme [44]. We will shortly describe only the first scheme; the second one has similar properties. The Shamir's method uses the polynomial interpolation. Each i th user has the point (x_i, y_i) shared with a Broadcast center as a preshared key. BC carries out the following steps:

- (i) it selects the random secret $(0, S)$ and some number (say, j) of additional dummy points (x_i, y_i) with x_i unassigned to a node,
- (ii) it finds a polynomial P of degree $k + j$ that passes through the points $(0, S)$ and (x_i, y_i) of the privileged set of k members and through j dummy points and no point of an unprivileged node,
- (iii) it broadcasts $k + j$ other points of the graph of P .

By the Lagrange interpolation polynomial, any privileged user is able to calculate $(0, S)$. Next, this secret is used as a session key. The scheme is secure (as the Shamir's scheme is), resilient, and the interpolation can be implemented in a fast way. The BC needs to broadcast $k + j$ messages, each node stores only one PSK. After any change in a privileged set the protocol must be repeated and the PSKs must be updated. In the next section we will show how to enhance that scheme, to make it more secure and flexible.

4. The One-Time Scheme with Additional Capabilities

In this section, at first we will propose methods for keys generation. We will show how a broadcast center and nodes can use a session key and preshared keys to achieve security goals (secret communication). Next we will consider the aspect of the source authentication. We will describe some popular methods realizing this security service that is crucial, especially for WSNs. Adequate keys management and authentication can really improve security and efficiency of BE schemes. The methods that will be proposed in this section are applicable in different schemes but we will show them in case of the Berkovits scheme [42] described in Section 4. Security analysis of the improvements proposed will be presented in Section 5.

Now, let us introduce a notation required in the rest of the paper: $H(\cdot)$ denotes a secure cryptographic hash function, $\text{Mac}_K(\cdot)$ denotes a secure message authentication code (MAC) with the key K , \parallel is a concatenation of two blocks of bits, \oplus is a XOR (exclusive OR) operation.

In the Berkovits' scheme, a node needs to store one secret point (a point of the polynomial's graph) that is coded as a block of bits. In each session, that point must be updated. During network's operation, one must enumerate the updated session keys. Let's assume that: K_e^i is a session key during i th session, k_x^i is a secret point (coded as a key) assigned to the node x during i th session. Its value is shared only with the BC, s_x is a secret seed of the node x . Its value is shared only with the BC, PSK_x^i denotes the set of preshared keys of the node x in i th session. The node stores only one set of PSKs at the same time (used in an actual session).

4.1. PSK and SK Derivation. As mentioned before, users using a *one-time* scheme must update keys every session. Now we describe this process. Let us define a generic function $\mathcal{G}en(\cdot)$, which is used for updating keys. In order to synchronize the keys $k_x^i = \mathcal{G}en(i, \text{PSK}_x^{i-1})$ is computed by the node x and by the BC. The BC and the nodes must share the same keys to make the steps of the protocol successfully. Additionally, the parties update PSK_x^{i-1} to PSK_x^i . Of course, the BC must generate a new session key (K_e^i) for a new session and create a new *header*. The number of a session (i) is passed by BC in the broadcast message msg :

$$msg = \langle i, \langle header \rangle, E_{K_e^i}(M) \rangle. \quad (3)$$

Now let us introduce two methods of key derivation. *Method I.* The first approach is simpler and therefore fast. In this solution, we define a key update as

$$\text{PSK}_x^i = \{s_x, k_x^i\}, \quad (4a)$$

$$k_x^i = \mathcal{G}en^1(i, \{s_x, k_x^{i-1}\}) = H(i \parallel s_x). \quad (4b)$$

The key derivation is realized by hashing the concatenation of the session number and the secret seed shared among the node x and the BC. We assume that the output length of $H(\cdot)$ is sufficient to produce a secure key. Each node holds only the actual key and the secret seed and it can generate a key for every session very fast.

Method II. The second method is more secure, but sometimes it requires more computations. The key derivation is realized as follows:

$$\text{PSK}_x^i = \{k_x^i\}, \quad (5a)$$

$$k_x^i = \mathcal{G}en^2(i, \{k_x^{i-1}\}) = H(k_x^{i-1}). \quad (5b)$$

A key for the first session (k_x^0) is preloaded before deployment.

In a new session the previous key k_x^{i-1} is replaced by k_x^i , and we must ensure that after such an exchange the old key k_x^{i-1} is erased from the node's memory. Each node stores only one key.

4.2. Source Authentication. Providing strong authentication in a distributed sensor network is a hard task [45]. To do this one must at first modify the broadcast message msg in (3). Now the BC sends

$$msg = \langle i, \langle header \rangle, E_{K_e^i}(M), AuthTag \rangle. \quad (6)$$

Besides a *header* and an encrypted content, the broadcast message must contain the authentication tag ($AuthTag$). Each legitimate node, using that tag, should be able to check authenticity of the message. Let us denote

$$msg = \langle i, \langle header \rangle, E_{K_e^i}(M) \rangle, \quad (7)$$

$$AuthTag = \mathcal{A}uth(msg).$$

$\mathcal{A}uth(msg)$ is an authentication function that can be realized in several ways. We will present two popular and one novel method.

The PKC Approach. PKC provides us digital signatures. It is an ideal method to authenticate BE messages, but only when the sender's resources are able to do this. The BC authenticates the broadcast traffic by signing it:

$$AuthTag = \mathcal{A}uth(msg) = \text{Sign}(msg). \quad (8)$$

$AuthTag$ is a message signed by the BC and when a user wants to check its authenticity, he uses the function of verification $\text{Verify}(msg, AuthTag)$. Such a function is easy to manage and is secure and scalable, but is rather impractical in environments like WSNs. Usually, PKC-based signatures are long and signing/verification operations require a computational overhead exceeding nodes' capabilities. We related to this aspect above, in Section 1.

The Standard Approach. Because PKC in WSNs is not recommended, we must use symmetric methods. Assume that K_a is an authentication key shared between legitimate nodes of a network and the BC. Source authentication is realized by means of MACs as follows:

$$AuthTag = \mathcal{A}uth(msg) = \text{Mac}_{K_a}(msg). \quad (9)$$

Verification in this and the next method is a simple computation of the tag $\text{Mac}_{K_a}(\cdot \cdot \cdot)$ as in (9) and checking if the computed tag and $AuthTag$ (appended with message) are equal. When the authentication key is shared among all members of a group then a sender of a message cannot be identified in a clear-cut way. Note that each owner of K_a can authenticate any message. It is acceptable when a BC has solely an ability to broadcast a content (e.g., in pay TV), but in sensor networks it may cause problems.

The Enhanced Approach. Now we want to improve the standard approach presented above making it useful for WSNs. We propose to attach the list of privileged nodes T to the broadcast message. Additionally, we XOR the session key K_e^i with K_a . Thus, the steps of the authentication protocol are

$$msg = \langle T, i, \langle header \rangle, E_{K_e^i}(M) \rangle, \quad (10)$$

$$AuthTag = \mathcal{A}uth(msg) = \text{Mac}_{K_a \oplus K_e^i}(msg).$$

Next, the BC distributes the broadcast message $bmsg$:

$$bmsg = \langle T, i, \langle header \rangle, E_{K_e^i}(M), AuthTag \rangle. \quad (11)$$

These small modifications have some consequences on security and efficiency of the protocol. We will present them in Section 5.

5. Analysis

Now, we will analyze the foregoing methods. We will start from analyzing functions of the BE scheme, next we will focus on the rekeying process and the authentication approaches while an analysis of security and performance will summarize this section.

The BE Functions. A *newcomer* is a node which is new in a privileged set in an actual session. j is the number of newcomers, and i is the number of an actual session. When we want *add* nodes to a privileged set, a natural way is to create a new set T , a new SK and a new *header*, and next broadcast them. In our case, this needs sending $O(|T|)$ messages, performing $O(1)$ computations in each node (using the key derivation in (5b) requires $O(i)$ computations for newcomers). Such a solution holds the *backward secrecy*, but the communication overhead is significant.

The easiest way is sending unicast to newcomers an encrypted message containing the session number i and the key k_e^i . It requires only $O(j)$ messages and no computations is required. However, this solution contradicts the *backward secrecy* and new nodes can decrypt all traffic along the session i . We must ensure that the newcomers are not able to decrypt previous messages. It can be achieved by sending only one additional message and performing one operation in the privileged nodes. During the session i the BC derives the new key $K_e^{i+1} = H(K_e^i)$, next it sends the key K_e^{i+1} to the newcomers and broadcasts an *update* message, which denotes that any privileged node should perform the calculation $K_e^{i+1} = H(K_e^i)$. The key K_e^i should be erased from the memory. Now, the session is updated to $i + 1$ and the broadcast traffic is encrypted with the key K_e^{i+1} . This is an efficient method. It requires only $O(j)$ short unicast messages and $O(1)$ computations in each node. The main advantage of that approach is assurance of the *backward secrecy*. The newcomers in the session $i + 1$ have the key K_e^{i+1} and they are not able to compute the previous key K_e^i .

Efficient *deletion* of nodes from T is one of the hardest tasks in BE protocols. Nodes leave, fail and sometimes we want to revoke malicious nodes. Presented BE scheme does not provide an efficient deletion function. If we want to delete some nodes then the BC must make a new session without these nodes. When we want to delete some nodes the BC must make a new session without these nodes. In that operation the *forward secrecy* is ensured. The BC generates a new SK independently, so the revoked node cannot decrypt present and future messages. Such an operation requires $O(|T|)$ messages and $O(1)$ operations in nodes.

An interesting solution for improving the performance of revocation is forming subsets of privileged nodes. We can divide T into several subsets and perform a BE scheme

on these subsets separately. This way we achieve subkeys and now we can repeat the process until we will generate a common SK. That strategy is related to *tree-based* schemes that are described in Section 2, or to other hierarchical constructions. Such a method can be used for pairwise key derivation in tree-based settings. Thus, the overheads are similar to overheads in other tree-based solutions presented in Section 2.

A *traitors-tracing* service known from DRM systems is not available, but the scheme allows to trace the source of a preshared key leak. When a pirate node uses a passed SK to decrypt the content, we are not able to determine the source of the leak. However, the SK for any session is different, so a traitor must pass a SK in each session. In a WSN it may be discovered by an anomaly detection or an intrusion detection system. More serious is an adversary's active attack. A privileged node can be captured and its session can be cloned into other nodes. Now an attacker is able to decrypt the traffic. When we detect a piracy hardware and tamper it, we can determine which node was captured. The BC keeps all seeds and actual keys of the node what requires $O(n)$ keys at the BC. When a protocol uses the method of key derivation proposed in (4b) then the BC needs to perform $O(n)$ operations, while using the tracing given in (5b) requires $O(in)$ computations. This second effort ($O(in)$) can be reduced using the method presented in [46] to $O(n \log^2 i)$. *Rekeying versus Not ReKeying.* BE schemes are designed to hold *backward secrecy* and *forward secrecy* properties in a sense of protection of unassigned messages against users joining or leaving the privileged set. However, in WSN-like applications, we should be more demanding. Consider a situation presented previously when an adversary captures a privileged node. It is standard assumption in environments like sensor networks. If a BE protocol does not require rekeying in each session, then an adversary having PSKs of a privileged node and its previous traffic can decrypt it all. In networks, where an adversary has capabilities to compromise nodes, we need *backward secrecy* assurance. To achieve this property, we must deploy some method of rekeying. We presented two approaches: in (4b) and in (5b). By solution given in (4b), we can generate a key of any session and at any moment. It is fast, but after compromising PSKs, also an adversary is able to obtain a key of any session. The approach given in (5b) holds *backward secrecy*. An actual user's key is produced from the previous key $k_x^i = H(k_x^{i-1})$. An adversary capturing a node has only one key which is the actual key. He can decrypt present and future messages, but he has no way to achieve previous keys. The only disadvantage is that a node which joins the privileged set must synchronize its key. In the extreme case a computational overhead of such a synchronization is $O(i)$. How to improve it by unicast messages we already described in this section.

Authentication. In this paragraph of the paper, we will present an analysis of authentication schemes presented in Section 4. We omit authentication schemes based on PKC due to reasons mentioned throughout the paper and we will focus on symmetric methods of authentication that are very efficient in WSNs, see for example, [47, 48]. At first we consider standard approach from (9). It is a good method

when authentication is realized between two parties. When a large group shares the same authentication key, each member can send or modify a message and the authentication will pass. We can improve (9) by a concatenation of the key K_e^i to an input of the MAC function: $\text{Mac}_{K_e}(K_e^i || \text{msg})$. Nevertheless, that approach is still insecure. First, when verification process fails, a node is not sure if a message is false or if it is outside of a privileged set (it is not able to achieve the key K_e^i). Next, an adversary with a compromised privileged node can disrupt sessions, each message can be authenticated by him. He can jam the original broadcast data and can create fake sessions with higher session numbers, in order to force nodes to synchronize session numbers and session keys. Such an attack can be destructive, when nodes derive keys using the method given in (5b). An adversary can also try to execute a DoS attack. Nodes, to verify *AuthTag*, must compute the key K_e^i . The adversary can just send (many times) big instance of BE problems and the nodes will be exhausted by computing them. These disadvantages are generic if a large group shares one authentication key.

Now, we will analyze next authentication method given in (10), which improves the previous one. The BC broadcasts the following message *mes*:

$$\begin{aligned} \text{AuthTag} &= \text{Mac}_{K_e \oplus K_e^i} \left(T, i, \langle \text{header} \rangle, E_{K_e^i}(M) \right), \\ \text{mes} &= \left\langle T, i, \langle \text{header} \rangle, E_{K_e^i}(M), \text{AuthTag} \right\rangle. \end{aligned} \quad (12)$$

Sending the privileged set T as a list of receivers T is an additional communication overhead. However, consider random nodes' enumeration from the set $0, \dots, n$. Only a node (and the BC) knows its own number. Then, we can encode the set T as a $\log_2 |T|$ -bits long binary mask. 0s and 1s on corresponding places in the mask denote that 0-marked nodes are unprivileged and 1-marked ones are privileged. Thus, even in large networks that overhead is acceptable. The solution presented influences efficiency. A node before any processing only checks if its bit is 1, otherwise it discards the message. In the previous methods, nodes always tried to achieve the SK. That improvement saves energy and provides other capabilities to the network (e.g., a possibility of routing). However, the main goal is the authentication. The BC, by sending list of T members, declares for which nodes the message is destined. This declaration means that any privileged node, after obtaining the key K_e^i from the *header*, is able to verify the signature *AuthTag*. If the verification fails then this means that the message is fake or a transmission error occurred. In both cases the node should send an alarm message. Consider now an adversary owning a group of nodes. He can create a message like in (12), but he must declare only the captured nodes in the list T of privileged nodes. Thus, any honest node even does not start processing a malicious message because it is not on the list of receivers attached by the adversary. If the adversary adds to that list a node, which is not compromised, then the node will start obtaining the key K_e^i and the signature's *AuthTag* verification will fail. This is because the adversary does not know the node's PSKs and he is not able to create a correct *header* without such a knowledge. An uncompromised node

should alarm the network. This way we achieved such useful properties of the network as

- (i) authentication of a fake message will pass if and only if the list of receiver nodes contains only compromised nodes,
- (ii) if an adversary adds an uncompromised node to the receivers' list then that added node is able to detect the forgery.

This means that a source which broadcasts a content must know all PSKs of the set of nodes T he declared, otherwise a regular node from that list can detect a forgery. However, the node processes a message only if it is on declared in the list, so a fake messages will not desynchronize a session. These features make our authentication scheme very useful in broadcast communication.

Security and Performance. Now we consider security of the scheme given in (10).

Security of Scheme based on Shamir's Secret Sharing Scheme which is *perfect secure* (information-theoretic secure) [42, 43].

Construction is resilient [42], so an attacker cannot obtain a session key.

Now we should consider confidentiality and integrity of the scheme.

Assume that encryption $E_K(\cdot)$ is indistinguishable under chosen plaintext attack (IND-CPA secure) and MAC scheme $\text{Mac}_K(\cdot)$ is strongly unforgeable under chosen-message attack (SUF-CMA secure).

Then, Theorems 4.4 and 3.2 from the paper [49] imply that construction from (10) is IND-CCA secure.

Now, we evaluate performance of our scheme. We assume that we use the presented scheme in tree-based setting (Figure 1), and we compare it also with the stateful tree-based broadcast encryption. We also assume that its security level is 128 bits. According to the properties of the schemes, their transmission and storage overheads are the same. The only difference is a computational overhead in the pairwise keys agreement process. The standard solutions use cryptographic primitives, thus for tests we chose assembler implementation [48] of the AES [50] block cipher. The polynomial interpolation required was implemented in C. ATmega1281 Microcontroller (with 8 MHz clock speed, 8 KB of RAM and 128 KB of Flash) was selected as a characteristic platform for a regular node in the sensor network. One pairwise rekeying function takes 0.4 ms on each sensor using the block cipher. At the same platform, the polynomial interpolation takes less than 0.1 ms. The implementation of the polynomial interpolation needs only 1329 bytes, while the AES uses 2141 bytes of storage.

6. Conclusions and Future Research

In this paper, we considered the BE problem in distributed wireless sensor networks. We defined the problem, described

main existing solutions, and next focused on *one-time schemes*, a type of schemes that are most suitable for WSNs. The one-time schemes are resilient, that is essential in such applications. Such protocols are very attractive for WSNs also in other respects. Their storage overhead of the range $O(1)$ (exactly up to 2 PSKs) is minimized as possible. We shown how standard BE operations may be realized to achieve efficient protocols, that is, with $O(1)$ operations and $O(j)$ messages in *add* operation and with $O(1)$ computations and $O(|T|)$ transmissions for *delete* operation. To improve transmission and revocation overheads, the protocols can be easily optimized, for example, by applying a tree-based structure. The schemes satisfy the *backward secrecy* and the *forward secrecy* properties. The *one-time* protocols are criticized due to a requirement of rekeying in each session. In our paper we showed that, unexpectedly, the *one-time* schemes in some applications are much better than other protocols. Because of the rekeying process in each session we can achieve the *backward secrecy* of PSKs. Another contribution to BE is including the authentication service to the protocols. We presented a symmetric cryptography-based method which has very useful security features. The scheme is also very efficient. The polynomial interpolation used is about four times faster than a fast block cipher encryption. This method ideally fits to WSNs and ensures that only a BC can generate authenticated sessions.

In this paper we extended the specific Berkovits' scheme [42], but we did not stress this fact, because our modifications are mainly generic and can be adopted in majority of BE schemes. The scheme used is relatively fast but in future research we will focus on designing more efficient *one-time* BE schemes. The new constructions will be dedicated to sensor networks. Another interesting subject of research is the *freshness* aspect in BE. It ensures that data transmitted is fresh beside of being confidential and authentic. Compiling the security services of data confidentiality, authenticity and freshness will make BE protocols more secure remaining them lightweight.

Acknowledgment

This work is partially supported by the National Science Center (NCN), under Grant with decision's number DEC-2011/01/N/ST7/02995 and by the 7FP NoE EuroNF project.

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] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [3] N. Xu, "A survey of sensor network applications," <http://courses.cs.tamu.edu/rabi/cpsc617/resources/sensor%20nw-survey.pdf>.
- [4] K. Sahrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, 2000.
- [5] P. Szalachowski, Z. Kotulski, and B. Ksiezopolski, "Secure position-based selecting scheme for wsn communication," in *Computer Networks*, A. Kwiecień, P. Gaj, and P. Stera, Eds., vol. 160 of *Communications in Computer and Information Science Computer Networks*, pp. 386–397, Springer, Heidelberg, Germany, 2011.
- [6] D. W. Carman, P. S. Kruus, and B. J. Matt, "Constraints and approaches for distributed sensor network security," Tech. Rep. 010, NAI Labs, The Security Research Division Network Associates, 2000.
- [7] M. Brown, D. Cheung, M. Kirkup, and A. Menezes, "Pgp in constrained wireless devices," in *Proceedings of the 9th USENIX Security Symposium*, pp. 247–261, USENIX, Denver, Colo, USA, August 2000.
- [8] A. S. Wandert, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom '05)*, pp. 324–328, March 2005.
- [9] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 53–57, 2004.
- [10] T. Kavitha and D. Sridharan, "Security vulnerabilities in wireless sensor networks: a survey," *Journal of Information Assurance and Security*, vol. 5, no. 1, pp. 31–44, 2010.
- [11] B. Chor, A. Fiat, and M. Naor, "Tracing traitors," in *Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '94)*, pp. 257–270, Springer-Verlag, London, UK, 1994.
- [12] D. Boneh and M. K. Franklin, "An efficient public key traitor tracing scheme," in *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '99)*, pp. 338–353, Springer-Verlag, London, UK, 1999.
- [13] E. Gafni, J. Staddon, and Y. L. Yin, "Efficient methods for integrating traceability and broadcast encryption," in *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '99)*, pp. 372–387, Springer-Verlag, London, UK, 1999.
- [14] M. Naor and B. Pinkas, "Efficient trace and revoke schemes," in *Proceedings of the 4th International Conference on Financial Cryptography (FC '00)*, pp. 1–20, Springer-Verlag, London, UK, 2001.
- [15] D. Naor, M. Naor, and J. B. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '01)*, pp. 41–62, Springer-Verlag, London, UK, 2001.
- [16] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02)*, pp. 41–47, Association for Computing Machinery, New York, NY, USA, November 2002.
- [17] R. di Pietro, L. V. Mancini, and A. Mei, "Random key-assignment for secure wireless sensor networks," in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor networks (SASN '03)*, pp. 62–71, Association for Computing Machinery, New York, NY, USA, October 2003.
- [18] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the 2003 IEEE Symposium on Security and Privacy (SP '03)*, p. 197, IEEE Computer Society, Washington, DC, USA, 2003.
- [19] M. Ramkumar, "Broadcast encryption with random key predistribution schemes," in *Proceedings of the 1st International*

- Conference on Information Systems Security (ICISS '05)*, Kolkata, India, 2005.
- [20] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast security: a taxonomy and some efficient constructions," in *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM-99)*, pp. 708–716, March 1999.
- [21] R. R. Brooks, B. Pillai, M. Pirretti, and M. C. Weigle, "Multicast encryption infrastructure for security in sensor networks," *International Journal of Distributed Sensor Networks*, vol. 5, no. 2, pp. 139–157, 2009.
- [22] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, "A survey of key management schemes in wireless sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2314–2341, 2007.
- [23] S. A. Çamtepe, B. Yener, and M. Yung, "Expander graph based key distribution mechanisms in wireless sensor networks," in *Proceedings of the 2006 IEEE International Conference on Communications (ICC '06)*, pp. 2262–2267, Istanbul, Turkey, July 2006.
- [24] Y. Amir, Y. Kim, C. Nita-Rotaru, and G. Tsudik, "On the performance of group key agreement protocols," *ACM Transactions on Information and System Security*, vol. 7, no. 3, pp. 457–488, 2004.
- [25] A. Fiat and M. Naor, "Broadcast encryption," in *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*, pp. 480–491, Springer-Verlag, York, NY, USA, 1994.
- [26] J. Horwitz, "A survey of broadcast encryption," Tech. Rep., 2003.
- [27] D. Wallner, E. Harder, and R. Agee, "Key management for multicast: issues and architectures," 1999.
- [28] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16–30, 2000.
- [29] D. A. McGrew and A. T. Sherman, "Key establishment in large dynamic groups using one-way function trees," *IEEE Transactions on Software Engineering*, vol. 29, no. 5, pp. 444–458, 2003.
- [30] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pp. 41–62, Springer-Verlag, Santa Barbara, Calif, USA, 2001.
- [31] D. Halevy and A. Shamir, "The lsd broadcast encryption scheme," in *Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '02)*, pp. 47–60, Springer-Verlag, Santa Barbara, Calif, USA, 2002.
- [32] C. Chang, Y. Su, and I. Lin, "A broadcast-encryption-based key management scheme for dynamic multicast communications work-in-progress," in *Proceedings of the 2nd International Conference on Scalable Information Systems (InfoScale '07)*, pp. 69:1–69:2, Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Suzhou, China, 2007.
- [33] A. D. Santis, A. L. Ferrara, and B. Masucci, "Efficient provably-secure hierarchical key assignment schemes," in *Proceedings of the 32nd International Symposium on Mathematical Foundations of Computer Science (MFCS '07)*, Cesky Krumlov, Czech, August 2006.
- [34] V. Daza, J. Herranz, P. Morillo, and C. Ràfols, "Ad-hoc threshold broadcast encryption with shorter ciphertexts," *Electronic Notes in Theoretical Computer Science*, vol. 192, no. 2, pp. 3–15, 2008.
- [35] E. S. Yoo, N. S. Jho, J. H. Cheon, and M. H. Kim, "Efficient broadcast encryption using multiple interpolation methods," *Lecture Notes in Computer Science*, vol. 3506, pp. 87–103, 2005.
- [36] D. Lubicz and T. Sirvent, "Attribute-based broadcast encryption scheme made efficient," in *Proceedings of the Cryptology in Africa 1st international conference on Progress in cryptology (AFRICACRYPT '08)*, pp. 325–342, Springer-Verlag, Casablanca, Morocco, 2008.
- [37] M. Ak, K. Kaya, K. Onarlioglu, and A. A. Selçuk, "Efficient broadcast encryption with user profiles," *Information Sciences*, vol. 180, no. 6, pp. 1060–1072, 2010.
- [38] C. Blundo, L. A. F. Mattos, and D. R. Stinson, "Trade-offs between communication and storage in unconditionally secure schemes for broadcast encryption and interactive key distribution," in *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '96)*, pp. 387–400, Springer-Verlag, Santa Barbara, Calif, USA, 1996.
- [39] R. Canetti, T. Malkin, and K. Nissim, "Efficient communication-storage tradeoffs for multicast encryption," in *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques (EURO-CRYPT '99)*, pp. 459–474, Springer-Verlag, Prague, Czech, 1999.
- [40] C. Padró, I. Gracia, and S. Martín, "Improving the trade-off between storage and communication in broadcast encryption schemes," *Discrete Applied Mathematics*, vol. 143, no. 1–3, pp. 213–220, 2004.
- [41] G. H. Chiou and W. T. Chen, "Secure broadcasting using the secure lock," *IEEE Transactions on Software Engineering*, vol. 15, no. 8, pp. 929–934, 1989.
- [42] S. Berkovits, "How to broadcast a secret," in *Theory and Application of Cryptographic Techniques*, vol. 547, pp. 535–541, 1991.
- [43] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [44] E. F. Brickell, "Some ideal secret sharing schemes," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, pp. 468–475, Springer-Verlag, New York, NY, USA, 1990.
- [45] B. Lampson, M. Abadi, M. Burrows, and E. Wobber, "Authentication in distributed systems: theory and practice," *ACM Transactions on Computer Systems*, vol. 10, no. 4, pp. 265–310, 1992.
- [46] D. Coppersmith and M. Jakobsson, "Almost optimal hash sequence traversal," in *Proceedings of the 6th international conference on Financial cryptography (FC '02)*, pp. 102–119, Springer-Verlag, Berlin, Germany, 2003.
- [47] P. Szalachowski, B. Ksiezopolski, and Z. Kotulski, "On authentication method impact upon data sampling delay in wireless sensor networks," in *Computer Networks*, A. Kwiecień, P. Gaj, and P. Stera, Eds., vol. 79 of *Communications in Computer and Information Science*, pp. 280–289, Springer, Berlin, Germany, 2010.
- [48] P. Szalachowski, B. Ksiezopolski, and Z. Kotulski, "CMAC, CCM and GCM/GMAC: advanced modes of operation of symmetric block ciphers in wireless sensor networks," *Information Processing Letters*, vol. 110, no. 7, pp. 247–251, 2010.
- [49] M. Bellare and C. Namprempre, "Authenticated encryption: relations among notions and analysis of the generic composition paradigm," *Journal of Cryptology*, vol. 21, no. 4, pp. 469–491, 2008.
- [50] J. Daemen and V. Rijmen, *The Design of Rijndael*, Springer-Verlag, Secaucus, NJ, USA, 2002.

Research Article

Error-Tolerant and Energy-Efficient Coverage Control Based on Biological Attractor Selection Model in Wireless Sensor Networks

Takuya Iwai, Naoki Wakamiya, and Masayuki Murata

Graduate School of Information Science and Technology, Osaka University, Suita, Osaka 565-0871, Japan

Correspondence should be addressed to Takuya Iwai, t-iwai@ist.osaka-u.ac.jp

Received 15 July 2011; Revised 5 November 2011; Accepted 10 November 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Takuya Iwai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A coverage problem is one of the important issues to prolong the lifetime of a wireless sensor network while guaranteeing that the target region is monitored by a sufficient number of active nodes. Most of existing protocols use geometric algorithm for each node to estimate the degree of coverage and determine whether to monitor around or sleep. These algorithms require accurate information about the location, sensing area, and sensing state of neighbor nodes. Therefore, they suffer from localization error leading to degradation of coverage and redundancy of active nodes. In addition, they introduce communication overhead leading to energy depletion. In this paper, we propose a novel coverage control mechanism, where each node relies on neither accurate location information nor communication with neighbor nodes. To enable autonomous decision on nodes, we adopt the nonlinear mathematical model of adaptive behavior of biological systems to dynamically changing environment. Through simulation, we show that the proposal outperforms the existing protocol in terms of the degree of coverage per node and the overhead under the influence of localization error.

1. Introduction

A wireless sensor network [1] has been attracting many researchers over the past ten years for a variety of its applications [2]. Among them, surveillance, monitoring, and observation of items, objects, and regions are most promising and useful. These applications require that a sufficient number of sensor nodes monitor the target region. Due to the uncertainty and instability of location and sensing area, it is difficult to deploy and manage sensor nodes in an optimal manner, that is, placing a minimum number of sensor nodes at the optimal positions. Therefore, a redundant number of sensor nodes are generally deployed in the target region. Then, for energy conservation, a sophisticated sleep scheduling mechanism is employed to keep the number of active sensor nodes as small as possible and let sensor nodes sleep as much as possible while satisfying the application's requirement on the degree of coverage. Such an issue to minimize the number of active sensor nodes while guaranteeing the required degree of coverage is called coverage problem [3–5]. There are many proposals on coverage problem. However, most of them rely on unrealistic assumptions, for example,

accurate location and perfect circular sensing area, and do not work well in the error-prone environment.

In this paper, to solve the problem, we propose a novel coverage control protocol, which is free from the above-mentioned unrealistic assumptions. Each sensor node does not need to know the shape and size of sensing area and the location and state of neighbor sensor nodes. A sensor node only relies on the information about the degree of coverage of the target region. To enable autonomous decision on sensor nodes, we adopt the nonlinear mathematical model called the attractor selection model. The model imitates flexible and adaptive behavior of biological systems to dynamically changing environment [6]. A biological system can autonomously and adaptively select an appropriate state for the environment only based on the condition of itself. Through simulation, we show that the proposal outperforms an existing protocol in terms of the degree of coverage per sensor node under the influence of localization error. In addition, our proposal requires less energy in monitoring the target region.

The remainder of this paper is organized as follows. First in Section 2, we briefly discuss related work. Next, in

Section 3, we introduce the biological attractor selection model. Then, in Section 4, we propose a novel coverage maintenance protocol adopting the attractor selection model. In Section 5, we evaluate the proposal through comparison with an existing protocol. Finally, in Section 7, we conclude this paper and discuss future work.

2. Related Work

There are many proposals on coverage problem, but most of them use geometric algorithm in order to estimate the degree of coverage. Based on the estimated degree of coverage, each sensor node determines whether to monitor around or sleep. For example, CCP [7] adopts the so-called K_s -Eligibility algorithm. First a sensor node identifies intersection points of borders of sensing areas of neighbor sensor nodes using a geometric arithmetic. Then, the sensor node evaluates whether all of intersection points inside its sensing area are inside sensing areas of more than K_s active sensor nodes or not. Since CCP assumes the accurate location information and perfect circular sensing area with radius R_s on all sensor nodes, it suffers from errors in the location information and the irregularity of the size and shape of sensing area. In addition, for a sensor node to evaluate the K_s -Eligibility algorithm, it has to obtain information about the location, sensing area, and state of neighbor sensor nodes at the sacrifice of bandwidth and energy in message exchanges. To increase the robustness against localization error, a location-free coverage maintenance protocol is proposed in [8]. The protocol adopts dominating set of graph theory, but it requires a sensing area to be circular and a transmission range to be adjustable. CARES [9] is another location-free protocol, where each sensor node stochastically and independently chooses its state based on general Markov model. However, sensor nodes must be uniformly distributed in the target region and the shape of sensing area must be circular. In the actual environment, localization error amounts to as much as several meters [10] and the shape of sensing area is not always circular at all. Therefore, these existing schemes do not work well outside the ideal environment, and an error tolerant coverage control method is desired.

3. Attractor Selection Model

The attractor selection model imitates the adaptive metabolic synthesis of *Escherichia coli* cells to dynamically changing nutrient condition [6]. A mutant bacterial cell has a metabolic network consisting of two mutually inhibitory operons, each of which synthesizes the different nutrient. When a cell is in the neutral medium, where both nutrients sufficiently exist, mRNA concentrations dominating protein production are at the similar level. This means that a cell can live and grow independently of the nutrient, which the cell synthesizes. Once one of nutrients becomes insufficient in the environment, the level of gene expression of an operon corresponding to the missing nutrient eventually increases so that a cell can survive by compensating the nutrient. Although there is no embedded adaptation rule as a signal transduction

pathway, a cell can successfully adapt gene expression in accordance with the surrounding condition.

In the attractor selection model, the dynamics of mRNA concentration m_1 and m_2 are represented by following equations:

$$\begin{aligned} \frac{dm_1}{dt} &= \frac{s(A)}{1+m_2^2} - d(A)m_1 + \eta_1, \\ \frac{dm_2}{dt} &= \frac{s(A)}{1+m_1^2} - d(A)m_2 + \eta_2. \end{aligned} \quad (1)$$

A ($1 \geq A \geq 0$) is the cellular activity such as growth rate and expresses the goodness of the current behavior, that is, the state of gene expression. Functions $s(A)$ and $d(A)$ are rational coefficients of mRNA synthesis and decomposition, respectively. In [6], $s(A) = 6A/(2+A)$ and $d(A) = A$ are used. η_i ($i = 1, 2$) corresponds to internal and external noise or fluctuation in gene expression.

Now let us explain the dynamics of mRNA concentrations following the attractor selection model. An attractor is a stable state, where a nonlinear dynamic system reaches after an arbitrary initial state. When the activity A is high, the nonlinear dynamic system formulated by the above equations has one attractor where $m_1 = m_2 = m^*$. Here, m^* is a constant and larger than one. When the sufficient nutrients are available, a cell grows well. Thus, a cell stays at the attractor and generates either one of two nutrients. Next, we assume that the environment lacks the nutrient, which a cell does not synthesize. Since it does not have the sufficient nutrient to grow, the activity decreases. When the activity becomes low, there appears two attractors, that is, $m_1 = m^*$ and $m_2 = 1/m^*$, or $m_1 = 1/m^*$ and $m_2 = m^*$, where either one of mRNA concentrations is higher than the other. Since the first two terms of the right side of (1) are multiplied by the activity, potential of attractors are shallow and dynamics is dominated by the noise terms. Consequently, m_1 and m_2 begin to change at random. When the mRNA concentration of the missing nutrient occasionally becomes large in a cell, the activity slightly increases as the cell can live better. The increase in the activity makes the potential of the attractor deeper and the state of a cell moves toward the attractor by entrainment. The activity further increases accordingly, and the influence of noise becomes smaller. Eventually, the state of a cell reaches an appropriate attractor and stays there stably as far as the nutrient condition does not change.

The attractor selection model is a kind of metaheuristics of optimization problem with dynamically changing given conditions. In the model, possible solutions are defined as attractors of the dynamic system by stochastic differential equations. An objective function to maximize is defined as the activity. In the biological case, a bacterial cell adaptively selects one of solutions, that is, synthesis of either one of two nutrients, so that the cell can maximize its growth rate according to the environmental nutrient condition. In our application of the attractor selection model to coverage control, a sensor node selects one of two states, that is, monitor around or sleep, to maximize the activity defined as the degree of coverage in the target region.

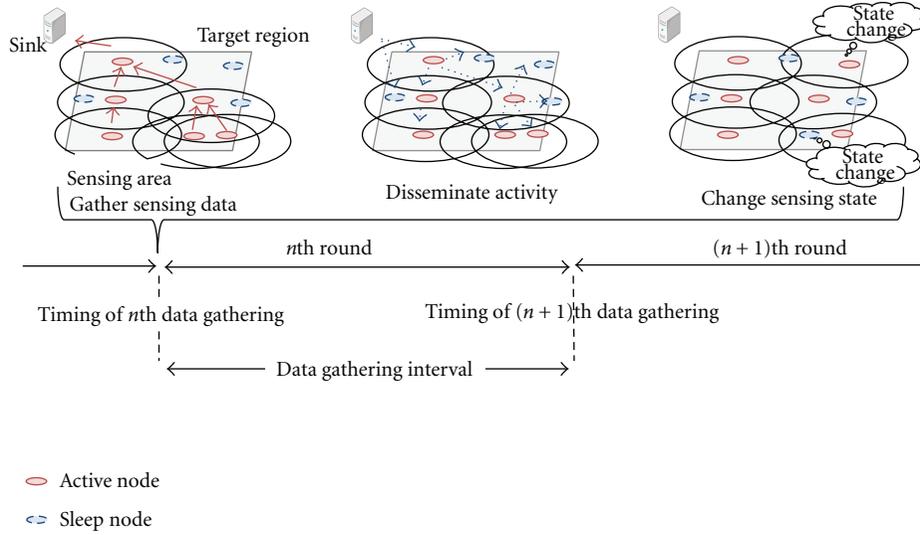


FIGURE 1: Overview of our proposal.

4. Attractor Selection-Based Coverage Control

In this section, we first outline the basic behavior of our proposal. Then, we describe the attractor selection model adopted in our proposal and the definition of the activity in coverage control. Finally, we describe the detailed behavior of sensor nodes in our proposal.

4.1. Overview of Our Proposal. In this paper, we consider a periodic monitoring application, where a sink collects sensing data from sensor nodes at regular intervals as illustrated in Figure 1. We refer to the interval as *data gathering interval* and the beginning of data gathering as *timing of data gathering*. We define the duration from the n th timing of data gathering until just before the $(n + 1)$ th timing of data gathering as the n th *round*.

At each timing of data gathering, each sensor node, which was active in the preceding round, transmits a message to a sink by single or multihop communication. A message consists of sensing data and the information for the sink to estimate the degree of coverage of the target region. Since we focus on coverage control, we do not assume any specific data-gathering mechanism to collect sensing data from sensor nodes. We also assume that the connectivity is maintained when the sufficient coverage is achieved [7]. Using received messages, a sink evaluates the degree of coverage of the target region. The way to evaluate the degree of coverage depends on the requirement of application and the information that sensor nodes can provide. When any localization mechanism is available at sensor nodes, the coverage is estimated based on the relative or absolute location of sensor nodes. An identifier of objects that a sensor node monitors is also useful information when a sink knows locations of the objects in the target region. In this case, each sensor node does not need to know its own location. From the degree of coverage, a sink derives the activity.

Then, a sink disseminates the activity information over a wireless sensor network by using any efficient dissemination

mechanism, for example, flooding, gossiping, or tree-based. Not only sensor nodes that are active in the preceding round but also one whose sleep timer expires at the timing of data gathering receives the activity. Sensor nodes that receive the activity decide whether to be active or sleep using the attractor selection model-based state selection mechanisms described below. If a sensor node decides to be active, it starts monitoring its surroundings. Otherwise, the sensor node sets its sleep timer at multiples of data-gathering interval and sleep immediately.

4.2. Extended Attractor Selection Model. In our proposal, we use the following attractor selection model, which is introduced in [11] for adaptive ad hoc network routing:

$$\begin{aligned} \frac{dm_1}{dt} &= \frac{\text{syn}(\alpha)}{1 + m_1^2} - \text{deg}(\alpha)m_1 + \eta_1, \\ \frac{dm_2}{dt} &= \frac{\text{syn}(\alpha)}{1 + m_2^2} - \text{deg}(\alpha)m_2 + \eta_2, \\ \text{syn}(\alpha) &= \alpha \times (\beta \times \alpha^\gamma + \varphi^*), \\ \text{deg}(\alpha) &= \alpha. \end{aligned} \quad (2)$$

This model has two attractors, that is, $m_1 > m_2$ or $m_2 > m_1$. β (>0) is a parameter related to the stability of attractor and γ (>0) is a parameter related to the speed of convergence. φ^* is a constant for the dynamic system to have stable attractors, and we use $1/\sqrt{2}$. α ($1 \geq \alpha \geq 0$) is the activity derived from the degree of coverage. The derivation of the activity will be explained in the next section.

4.3. Derivation of Activity. In our proposal, as stated in Section 4.1, any estimation algorithm of the degree of coverage can be adopted. In this paper, we consider the following derivation for the sake of easy implementation and comparison. First, the target region is divided into small regions of

$1 [m] \times 1 [m]$, which is called *patch*. In the target region of $x_t [m] \times y_t [m]$, a patch at the column x ($x_t \geq x \geq 1$) and the row y ($y_t \geq y \geq 1$) is indicated by (x, y) . The degree of coverage $C(x, y)$ of patch (x, y) is approximated by the number of active sensor nodes whose sensing area covers a center of patch (x, y) .

Guaranteeing any point of the target region to be monitored by k active sensor nodes is called k -coverage. When an application requires k -coverage, the sensing ratio S ($1 \geq S \geq 0$) of the whole target region is derived by the following equation:

$$S = \frac{|\{(x, y) \mid C(x, y) \geq k\}|}{x_t y_t}. \quad (3)$$

The sensing ratio S does not take into account the excess and deficiency in monitoring, that is, whether a patch is in the sensing area of more or less than k active sensor nodes. Therefore, coverage control using the sensing ratio S as the activity α leads to the waste of energy or deficient coverage. To solve this problem, we formulate the excess and deficiency ratio E (≥ 1) for the whole region:

$$E = \frac{\sum_{i=1}^{x_t} \sum_{j=1}^{y_t} |C(i, j) - k|}{x_t y_t} + 1. \quad (4)$$

Then, the activity α for the whole target region is derived as follows:

$$\alpha = \frac{S}{\max\{1, wE\}}, \quad (5)$$

where larger w ($1 \geq w > 0$) leads to more efficient control with less active sensor nodes, but it becomes difficult for sensor nodes to reach solutions, which are deficient or redundant coverage. Operator “max” is introduced to prevent the activity from exceeding one. We call the activity derived in (5) the *global activity*.

For fine-grained control, we can also define the area activity using the sensing ratio per small areas of the target region. In this case, the target region is divided into some subareas of $x_s [m] \times y_s [m]$, where x_s and y_s are divisors of x_t and y_t . A subarea at the column x and the row y is indicated by (x, y) , where $x_t/x_s \geq x \geq 1$ and $y_t/y_s \geq y \geq 1$. The sensing ratio $S'(x, y)$ of subarea (x, y) is derived by the following equation:

$$S'(x', y') = |\{(x, y) \mid C(x, y) \geq k, \\ (x' - 1)x_s + 1 \leq x \leq x'x_s, \\ (y' - 1)y_s + 1 \leq y \leq y'y_s\}|/x_s y_s. \quad (6)$$

We formulate the excess and deficiency ratio $E'(x, y)$ (≥ 1) for the subarea (x, y) as follows:

$$E'(x, y) = \frac{\sum_{i=(x-1)x_s+1}^{xx_s} \sum_{j=(y-1)y_s+1}^{yy_s} |C(i, j) - k|}{x_s y_s} + 1. \quad (7)$$

Then, the activity $\alpha'(x, y)$ of the subarea (x, y) is given as follows:

$$\alpha'(x, y) = \frac{S'(x, y)}{\max\{1, wE'(x, y)\}}. \quad (8)$$

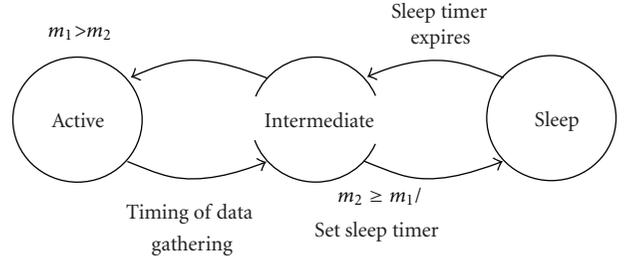


FIGURE 2: State diagram of our proposal.

The activity derived by (8) is called the area activity. In the case of the area activity-based control, a sink evaluates all area activities $\alpha'(x, y)$ in (8), and a message from a sink contains all area activities. A sensor node uses the area activity of a subarea in which the sensor node is considered to be located. It implies that a sensor node with inaccurate location information uses the area activity of an inaccurate subarea.

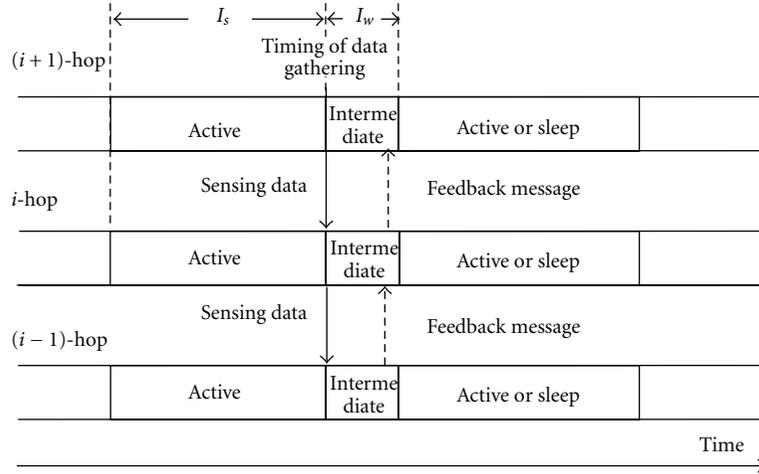
4.4. Node Behavior. A sensor node has three states, that is, active, sleep, and intermediate as illustrated in Figure 2. In each state, a sensor node behaves as follows.

4.4.1. Active State. A sensor node monitors its sensing area by turning and keeping sensor modules on and transceiver modules off for the fixed period I_s (>0) [s], which is called *sensing interval*. When the timing of data gathering arrives, a sensor node turns on transceiver modules and sends sensing data toward the sink. Then, it moves to the intermediate state.

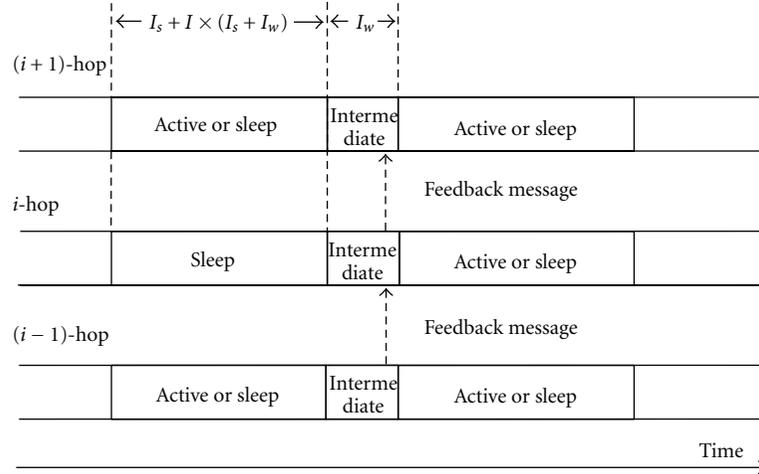
4.4.2. Sleep State. A sensor node turns and keeps all modules off to save its battery. When a sleep timer expires, a sensor node turns on transceiver modules and moves to the intermediate state.

4.4.3. Intermediate State. A sensor node waits for receiving a feedback message from the sink during the fixed period I_w (>0) [s], called *intermediate interval*. The feedback message contains the activity α , which reflects the degree of coverage. The node evaluates two equations in Section 4.2 to update m_1 and m_2 using the received activity. In this paper, we assume that above-mentioned transactions are finished within the constant time I_w . Using updated m_1 and m_2 , sensor nodes select the next state as follows. In case of $m_1 > m_2$, the sensor node moves to the active state. On the other hand, in case of $m_2 \geq m_1$, the sensor node sets its sleep timer as $I_s + l \times (I_s + I_w)$ and moves to the sleep state. l (>0) is a control parameter which is randomly chosen with uniform distribution between 0 and 4 to avoid synchronous behavior of sensor nodes. $I_s + I_w$ corresponds to the data-gathering interval introduced in Section 4.1.

Next, we briefly explain how a sensor node behaves in message transmission in simulation. In the case of a sensor node which was in the active state in the preceding round, it participates in both data gathering and feedback dissemination as illustrated in Figure 3. At the end of active period I_s ,



(a) Preceding state is active



(b) Preceding state is sleep

FIGURE 3: Behavior of sensor nodes on i -hop.

the timing of data gathering comes. Although mechanisms of data gathering and feedback dissemination are out of scope of this paper, here we consider a tree-based routing. A sensor node located at i -hop from a sink receives messages from its child sensor nodes and aggregates their sensing data with its own. Then, it sends a message containing aggregated sensing data to its parent sensor node located at $(i - 1)$ -hop from a sink and moves to the intermediate state.

During feedback dissemination, a sensor node located at i hop from a sink first receives a feedback message from its parent sensor node during the intermediate interval I_w . Then, it broadcasts the message to its child sensor nodes located at $(i + 1)$ -hop from a sink and determines the next state. On the contrary, when a sensor node located at i -hop from a sink was in the sleep state in the preceding round, it does not send sensing data. It wakes up at the timing of data gathering and immediately moves to the intermediate state. Next, it receives a feedback message from its parent sensor node, which was in either of the active or sleep state in the preceding round. Then, it forwards the message to its child sensor nodes and makes a decision on the next state.

4.5. Advantages of Our Proposal. Our proposal have advantages over existing protocols, which require a sensor node to obtain the information of neighbor sensor nodes, that is, location and state. First, our proposal is more robust against the inaccuracy of location information and the irregularity or uncertainty of sensing area than others. In our proposal, a sensor node only requires the degree of coverage of the whole target region or the located area. Even if the derivation of the degree of coverage at a sink uses location information of sensor nodes, the influence of localization error can be mitigated by considering the degree of coverage over the whole target region or the area of a certain size.

Second, our proposal requires less energy in coverage control than others. In other existing proposals, so that a sensor node can appropriately determine the next state using a geometric algorithm, it has to collect sufficient amount of information by receiving many messages from neighbor sensor nodes. Although a sensor node only needs to broadcast a message once to inform neighbor sensor nodes of its information, such message exchanges must be done in addition to regular message transmission for data gathering.

On the other hand, our proposal only requires a sensor node to obtain the activity for selecting its sensing state. A sensor node only needs to transmit one message for data gathering and one more for feedback dissemination. Therefore, a sensor node can effectively turn off its transceiver for longer duration than others. These advantages of our proposal will be proved by simulation in the next section.

5. Simulation Experiments

In this section, we first explain error models, that is, localization error and shape error. Simulation results follow to compare our proposal with CCP in terms of the sensing ratio, the number of active sensor nodes, the redundancy ratio, the contribution ratio, and the energy consumption.

5.1. Localization Error. Based on [12], we consider a simple model of localization error. The amount of error is uniformly distributed between $-u$ and u , where u is the maximum error in meter. Then, erroneous coordinates of a sensor node at geographical coordinates (x, y) is given at random in the area of $(x - u, y - u)$ as the left bottom corner and $(x + u, y + u)$ as the right top corner.

In our proposal, a sink evaluates the global or area activity with wrong location information received from neighbor sensor nodes. Therefore, the activity notified to sensor nodes is different from the actual degree of coverage. On the other hand, a sensor node with CCP calculates intersections of sensing areas based on wrong location information. Therefore, the K_s -Eligibility algorithm would give a wrong answer.

5.2. Shape Error. Since there is no model of the irregularity of sensing area, we adopt the model of the irregularity of radio propagation introduced in [13]. RIM (Radio Irregularity Model) models the variation in the received signal strength under the influence of heterogeneous energy loss. In wireless communication, the signal strength decreases in accordance with the distance from the transmitter. The following is the commonly used model to estimate path loss L [dBm] [14]:

$$L = C + 10n \log_{10} d, \quad (9)$$

where C is a constant and n expresses the quality of transmission path. Parameter d is the distance between the transmitter and the receiver. Then, RIM introduces the irregularity in path loss as

$$\begin{aligned} R &= T - \text{DOI Adjusted Path Loss} + F, \\ \text{DOI Adjusted Path Loss} &= L \times K_i. \end{aligned} \quad (10)$$

R represents the received signal strength and T corresponds to the transmission power. F corresponds to the fading effect. K_i implements the difference in path loss at the i th degree. K_i is given by the following equation:

$$K_i = \begin{cases} 1, & \text{if } i = 0, \\ K_{i-1} \pm r\text{DOI}, & \text{if } 360 > i > 0 \wedge i \in N, \end{cases} \quad (11)$$

where $\text{DOI} \geq |K_0 - K_{359}|$.

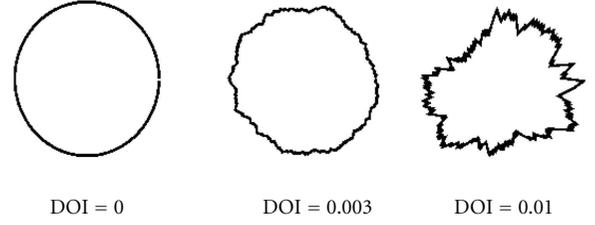


FIGURE 4: Irregular sensing area.

Here, DOI (degree of irregularity) is the coefficient of the irregularity. r is a random number following the Weibull distribution.

For example, we depict the impact of different DOI in Figure 4. Each shape shows the border of region where the received signal strength exceeds a certain threshold. As can be seen, $\text{DOI} = 0$ gives a circular shape. As DOI increases, the shape becomes more irregular. We first set parameters of RIM appropriately to obtain the regular circle shape of the desired sensing radius and then change DOI to see the influence of irregularity in simulation experiments.

5.3. Energy Model. We define the energy model based on MICAz [15, 16]. CPU consumes 8 [mA] when it is on and 15 [uA] when it is off. A transceiver module consumes 19.7 [mA] in listening a channel and receiving a message and 17.4 [mA] in transmitting a message. A sensor module consumes 10 [uA] when it is on and 0 [uA] when it is off. When a sensor module monitors objects, CPU is activated as well. We assume that a sensor node runs on two AA batteries of 3 [V].

As explained in Section 4.4, we consider a tree-based routing for data gathering and feedback dissemination. In data gathering, a sensor node receives sensing data from its child sensor nodes, generates the aggregated data of the same size of a single sensing data, and sends it to a parent sensor node. In disseminating feedback messages, a sensor node receives a message containing the activity from its parent sensor node and broadcasts it to all child sensor nodes.

5.4. Simulation Setting. We distribute about 10,000 sensor nodes in the square target region. A sink is located in the center of the target region. In the case of the global activity-based control, 10,000 sensor nodes are randomly deployed in the target region of 500 [m] \times 500 [m]. In the case of the area activity-based control, we first set the size of a subarea and then determine the size of the target region as the multiple of a subarea around 500 [m], while keeping the density 0.04 [node/m²]. For example, when the size of subarea is 15 [m] \times 15 [m], 10,404 sensor nodes are distributed in the target region of 510 [m] \times 510 [m]. An application requires 1, 2, or 3-coverage ($k = 1, 2, \text{ or } 3$). Data-gathering interval ($I_s + I_w$) is set at 10 [s]. Sensing interval I_s is 9 [s] and wakeup interval I_w is 1 [s]. At the beginning of a simulation run, all sensor nodes are in active state.

In our proposal, both m_1 and m_2 are initialized to 1 and the initial activity is initialized to 0. Parameter, β and γ are set

at 2.5 and 1.2, respectively. Weight w is set at 0.5. The parameter l of rounds of sleep state in our proposal is randomly chosen between 0 and 4 with uniform distribution. These parameters are selected through preliminary experiments. In CCP, HELLO interval, SLEEP, WITHDRAW, JOIN, and LISTEN timers are set at 1 [s], 10 [s], 1 [s], 1 [s], and 1 [s], respectively. Regarding details of these parameters, refer to [7]. For the purpose of comparison, we define ACTIVE and JOIN state of CCP as active state.

The communication range R_c is set at 20 [m]. We use our own simulator and we assume the ideal communication environment; that is, there is no loss or delay of message. The shape of sensing area is a circle of radius $R_s = 10$ [m] and identical among sensor nodes under the condition without shape error. In our proposal, a sink assumes the circular sensing area with radius 10 [m] and believes the location information reported by sensor nodes in derivation of the activity. In CCP, intersection points between borders of sensing areas of neighbor sensor nodes are calculated under the assumption that there is neither localization error nor shape error. For evaluation of the tolerance to localization error, we change the maximum location error u from 0 [m] to 10 [m], for example, GPS-based localization. For evaluation of the tolerance to shape error, we change DOI from 0 to 0.03.

5.5. Performance Measures. As performance measures, we use the number of active nodes N , the contribution ratio B , the redundancy ratio U , and the energy consumption O . The contribution ratio B indicates the degree of contribution of an active sensor node to coverage. B is derived as

$$B = \frac{M \times S}{N} [\text{m}^2], \quad (12)$$

where M [m²] is the size of target region and S is the sensing ratio derived by (3) with the accurate coordinates and sensing area. Therefore, the contribution ratio represents the average area that an active sensor node is responsible for monitoring. The larger contribution ratio means that sensor nodes are more efficiently monitoring.

Next, we define the redundancy ratio U as the averaged extra degree of coverage per patch for achieving k -coverage. The redundancy ratio is derived as

$$U = \frac{\sum_{i=1}^{x_t} \sum_{j=1}^{y_t} Z(C(i, j))}{|\{(x, y) \mid C(x, y) \geq k\}|}, \quad (13)$$

$$Z(x) = \begin{cases} x - k + 1, & \text{if } x \geq k, \\ 0, & \text{if } x < k, \end{cases}$$

where the target region is x_t [m] \times y_t [m] and the coverage $C(x, y)$ of patch (x, y) is approximated by the number of active nodes that has a center of patch (x, y) in its own sensing area. Therefore, the larger redundancy ratio means that too many nodes are in the active state.

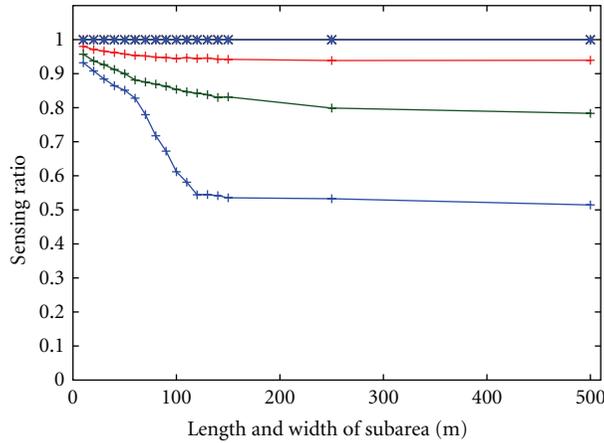
Finally, the energy consumption O is derived using our energy model described in Section 5.3. We take into account

state-dependent energy consumption and energy consumed in message transmission and reception. We should note here that the overhead related to management of location information is not considered in the evaluations. First, we assume that a sink obtains identifiers and location-related information from all sensor nodes in advance. We further assume that both of CCP and our proposal adopt the same localization technique. Messages sent from a sensor node contain its identifier, whose size is small enough. As a result, the amount of overhead regarding management of location information is almost the same among CCP and our proposal, and the difference is negligible. Influences of inaccuracy in location information are taken into account in the energy consumption O , since inaccurate location information affects states of sensor nodes and the amount of message transmission.

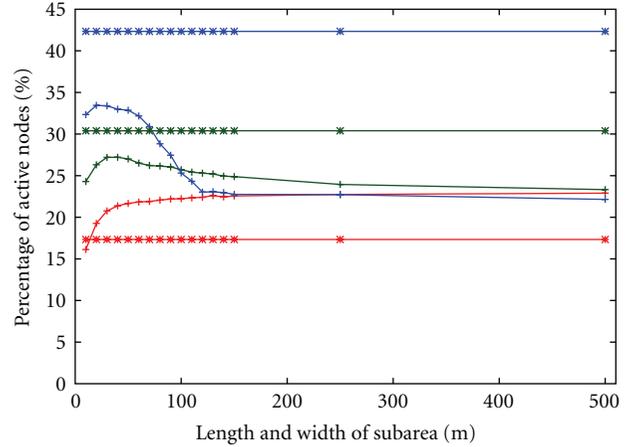
5.6. Basic Evaluation. First we compare our proposal with CCP under the ideal environment, where there is neither localization error nor shape error. In Figure 5, the x -axis indicates the width and height of a subarea, that is, x_s and y_s , for the area activity-based control. $x_s = y_s = 500$ [m] corresponds to the case of the global activity-based control where the target region is not divided into any subarea. The y -axis shows the sensing ratio derived by (3). When there is no error, CCP accomplishes the sensing ratio S of 1.0 for $k = 1, 2, \text{ and } 3$ as shown in Figure 5(a).

Under the ideal environment, sensor nodes can accurately estimate the degree of coverage inside sensing areas of themselves. Figure 5(b) shows that the percentage of active sensor nodes with CCP increases almost in proportional to the required coverage. In spite of a deterministic and geometric algorithm of CCP, the redundancy ratio is higher than 2 and up to 3.2 as shown in Figure 5(c). Even if an uncovered area inside a sensing area of a sensor node is small, a sensor node becomes in active state to cover the area. This results in the redundant coverage of the other area which is already covered. However, such redundancy is unavoidable for the irregularity of deployment of sensor nodes and the shape of sensing area.

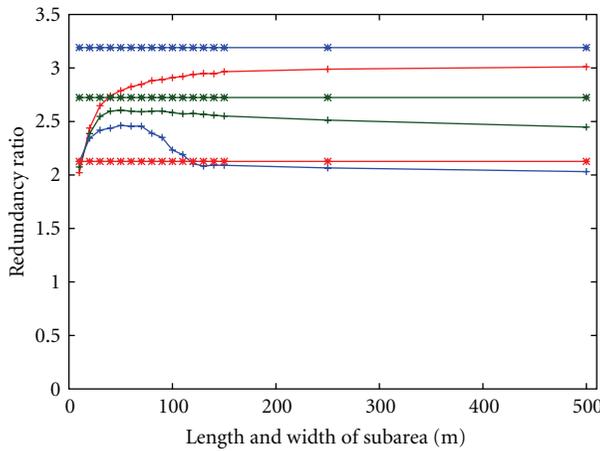
Compared to CCP, the sensing ratio with our proposal is lower especially when the size of subarea is large as shown in Figure 5(a). Our proposal adopts the metaheuristic algorithm, that is, attractor selection model, to find a solution. As such, the size of search space affects the optimality of the found solution. In case of the global activity-based control, the number of combinations of state of sensor node is as large as 2^{10000} . In addition, a state of a sensor node does not influence others very much. Therefore, our proposal often falls into local optimal. However, as the sizes of subareas decreases, the sensing ratio of our proposal approaches 1. When the size of subarea is smaller, the number of sensor nodes per subarea decreases. As a result, the size of solution space becomes smaller and there appears stronger interdependency among state of sensor nodes. In other word, with the smaller size of subarea, sensor nodes can find better solution, which has higher sensing ratio and less redundancy ratio. In general, when a sensor node selects the active state, it increases both of the sensing ratio and the redundancy ratio. When the sensing ratio is low, an increase in the



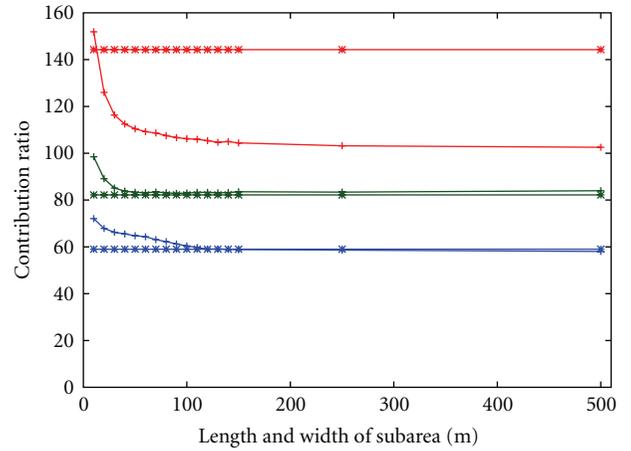
(a) Sensing ratio



(b) Percentage of active sensor nodes



(c) Redundancy ratio



(d) Contribution ratio

FIGURE 5: Comparison without errors.

sensing ratio increases the activity more than the decrease caused by increased redundancy ratio. It is a reason that there are more active sensor nodes with smaller subareas in Figure 5(b) for $k = 2$ and 3 . On the other hand, when k is 1 , even with a small subarea, it is hard for an additional active sensor node to increase the sensing ratio, which is already high enough. Therefore, the coverage control moves toward reducing the redundancy ratio to increase the activity as shown in Figure 5(c).

Regarding the contribution ratio, a smaller subarea leads to the higher contribution ratio. As shown in Figure 5(d), when k is 2 or 3 , our proposal can achieve higher contribution ratio than CCP in any size of subarea. On the contrary, when k is 1 , CCP achieves higher contribution ratio than our proposal in almost all size of subarea. When x_s and y_s are 500 [m] and k is $1, 2$, or 3 , about 22 percent of sensor nodes becomes active state. In comparison with CCP, in case of $k = 1$, the number of active sensor nodes is redundant to

achieve the perfect 1 -coverage ($k = 1$). In addition, due to the low optimality of the found solution, our proposal achieved less sensing ratio than CCP. Because of low sensing ratio and redundant active sensor nodes, our proposal achieves less contribution ratio than CCP. Using smaller subareas, our proposal can find better solutions, that is, achieving higher sensing ratio by less active sensor nodes. In particular, when k is 1 and x_s and y_s are 5 [m], the number of active sensor nodes in our proposal drops to below CCP, and the magnitude relation of contribution ratio is reversed. In addition, when k is 2 and 3 , the number of active sensor nodes increases unlike when k is 1 , but the sensing ratio also more increases. Therefore, higher contribution ratio can be achieved as subareas become smaller.

5.7. Influence of Localization Error. In this section, we compare CCP and two variants of our proposal, that is, the global activity-based control and the area activity-based control

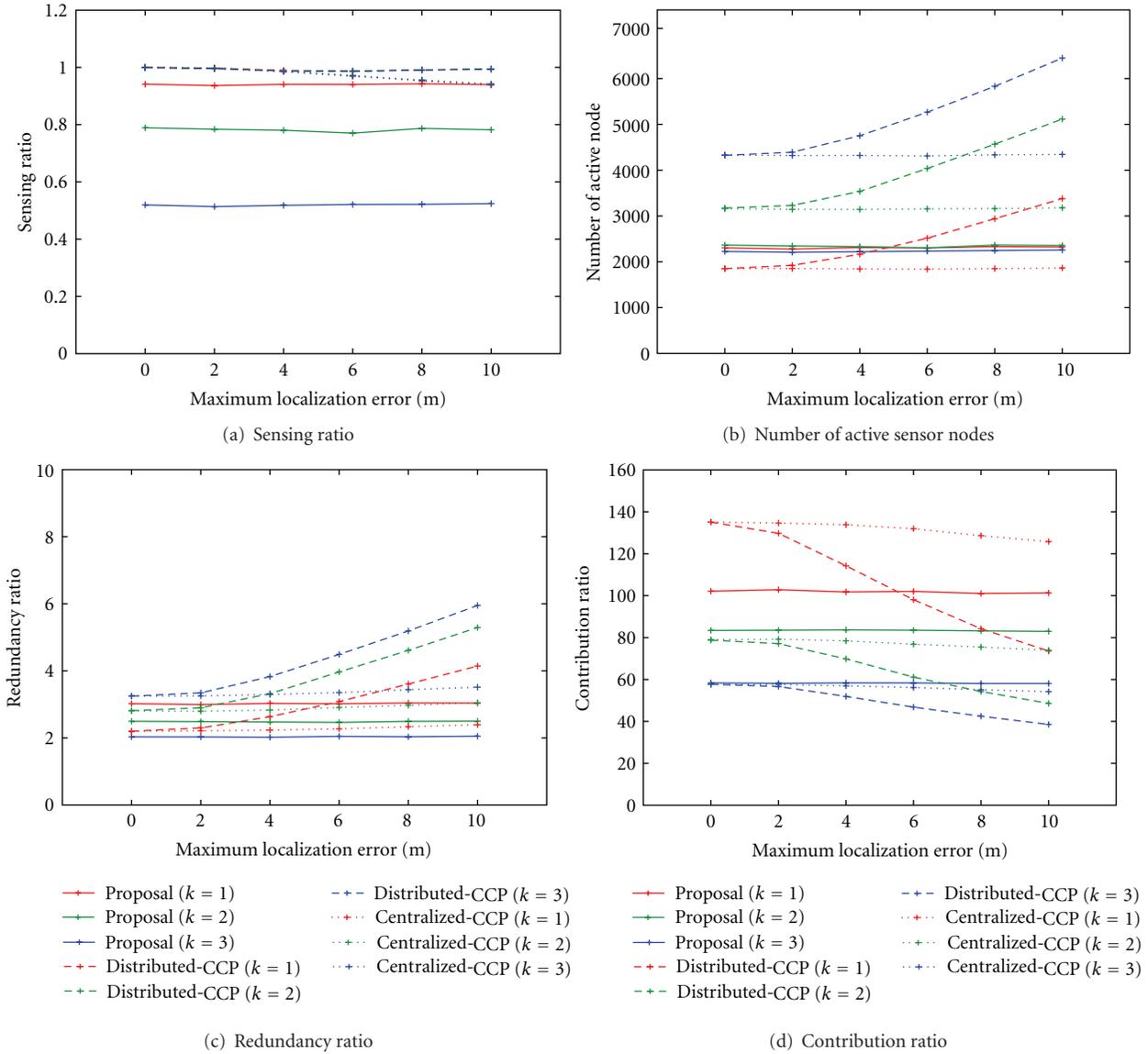


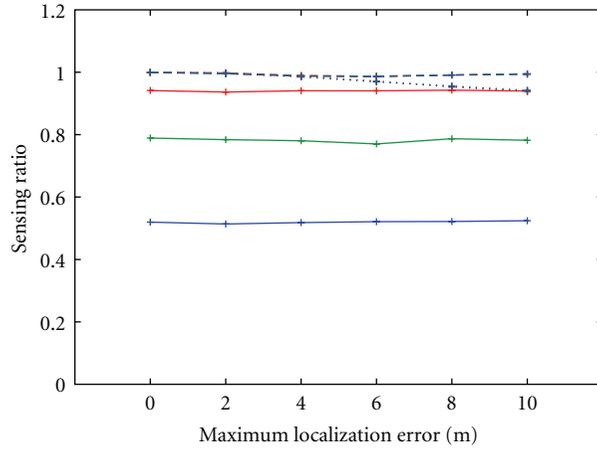
FIGURE 6: Influence of localization error (global activity).

whose subarea size is set at $10 [m] \times 10 [m]$, under the influence of localization error. For the sake of argument about the origin of the error tolerance of our proposal, we show the results of CCP with the center-point control in addition to the results of original CCP. We call the original CCP “distributed-CCP” and the CCP with the center-point control “centralized-CCP”. In the centralized-CCP, a sink collects the sensing state and location-related information from all sensor nodes and conducts the K_s -Eligibility algorithm for each of the sensor nodes. Then, the determined state is sent back to each sensor nodes. To ignore the influence of shape error, DOI is set at zero. Figures 6 and 7 summarize results averaged over 10 simulation runs.

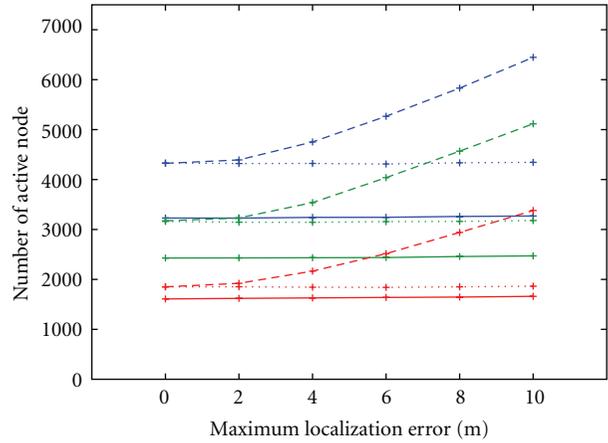
Figure 6(a) shows the average sensing ratio S of the global activity-based control against the different degree u of localization error. In the figure, it is obvious that neither our proposal nor distributed-CCP is affected by localization

error. In our proposal, a sink calculates the activity from collected sensing data. Since the effect of localization error is averaged over the whole region, the derived activity is not seriously affected by localization error. On the contrary, distributed-CCP uses geometric and deterministic algorithm, and as such state selection heavily depends on the accuracy of location information. Nevertheless, distributed-CCP keeps the high sensing ratio. The reason is that localization error and wrong state selection are compensated by the increased number of active sensor nodes and the higher redundancy as shown in Figures 6(b) and 6(c).

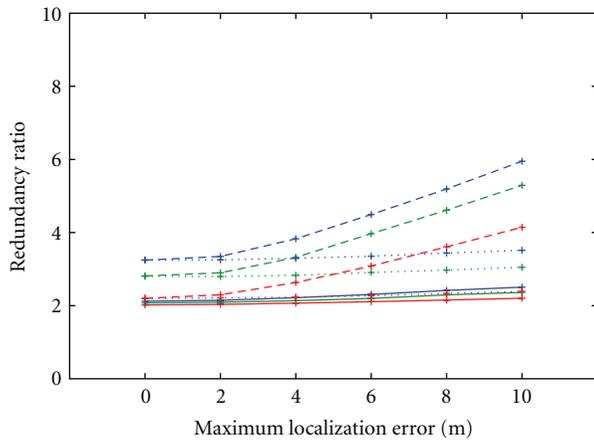
In CCP, localization errors contribute to both of increase and decrease in the number of active nodes. When a sensor node wrongly considers that a neighbor sensor node is far and there is no overlap between their sensing areas by localization error, it is likely to become active to monitor intersections which seem to be uncovered. At the same time,



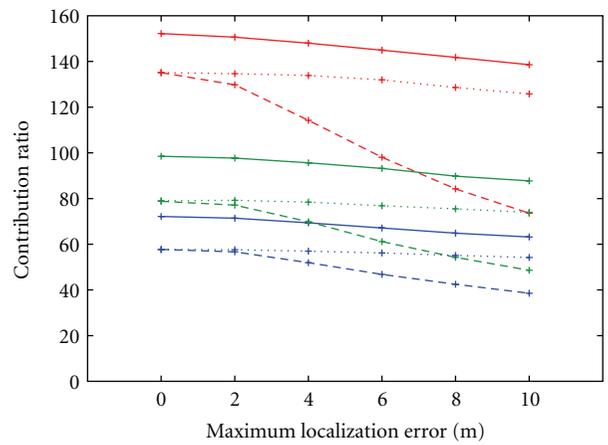
(a) Sensing ratio



(b) Number of active sensor nodes



(c) Redundancy ratio



(d) Contribution ratio

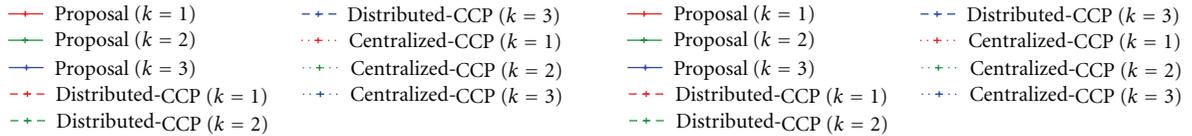


FIGURE 7: Influence of localization error (area activity).

localization error makes a sensor node consider a further neighbor to be located close. Consequently, the affected node is likely to move to the sleep state. In the case of the distributed-CCP, a decision of a sensor node is affected only by neighbor sensor nodes within its communication range. From results of the distributed-CCP in Figure 6(b), localization error results in the increase more than the decrease. On the contrary, in the case of the centralized-CCP, a sensor node is further affected by localization error of a sensor node whose actual location is out of its communication range. The actual sensing area of such a distant sensor node does not overlap with the sensing area of the sensor node. Therefore, even if the distant sensor node is considered to be located further by localization error, it does not influence a decision of the sensor node at all. However, when the sensor node considers the distant sensor node is located closer to itself by localization error, it would move to the sleep state. As a result,

the number of active sensor nodes becomes smaller than that of the distributed-CCP. Since the increase and decrease are occasionally balanced for uniformly random distribution of sensor nodes, the number of active sensor node becomes constant against localization errors.

As a result, the redundancy ratio with the centralized-CCP becomes smaller than the distributed-CCP (Figure 6(c)), and the centralized-CCP is more prone to the localization error than the distributed-CCP in terms of the sensing ratio (Figure 6(a)). Similarly, in our proposal, the derived activity is not also seriously affected by localization error by averaging error over the whole region, we can achieve the similar performance without increasing the number of active sensor nodes.

To evaluate the efficiency of coverage control, Figure 6(d) shows the contribution ratio B against the different degree of localization error. As can be expected from Figure 6(b),

the contribution ratio of distributed-CCP decreases as the maximum localization error increases. For example, when an application requires 1-coverage ($k = 1$), the global activity-based control accomplishes more efficient coverage control than CCP with maximum localization error u of 6 meters or more. When an application requires 2 or 3-coverage ($k = 2$ or 3), our proposal always outperforms both the distributed-CCP and the centralized-CCP in terms of the contribution ratio. When we divide the target region into subareas whose size is 10 [m] and apply the area activity-based control, we can achieve higher sensing ratio than the global activity-based control. Especially, in the case of $k = 1$, the similar degree of sensing ratio can be achieved with the smaller number of active sensor nodes. Moreover, the area activity-based control outperforms both distributed-CCP and centralized-CCP in terms of the contribution ratio while the sensing ratio is sufficiently high such as more than 0.8.

However, the sensing ratio gradually decreases as the localization error increases. In comparison with the global activity-based control, the redundancy ratio is lower and the contribution ratio is higher with the area activity-based control (compare Figure 6(c) with Figure 7(c), Figure 6(d) with Figure 7(d)). It implies that an uncovered patch has less chance to be covered by a nearby active sensor node than with the global activity-based control. However, even if there is the high localization error, the area activity-based control can achieve the sensing ratio similar to or better than the global activity-based control.

From the above results, we can conclude that our proposals are more robust than distributed-CCP. Although centralized-CCP exhibits the similar robustness in the number of active nodes to our proposal due to the center-point control, our proposal is superior to centralized-CCP. Further discussions will be given in Section 6. Although distributed-CCP can maintain sensing ratio close to one against localization error, the number of active sensor nodes considerably increases. It depletes batteries and shortens the lifetime of a sensor network. Although sensing ratio is slightly lower with the area activity-based control than distributed-CCP even without localization error, the number of active sensor nodes do not change much, and we can expect the similar lifetime under the influence of localization error, which is quite common in the actual environment. When we consider such applications that do not always require sensing ratio of 100%, for example, precision agriculture and environmental monitoring, our proposal is more practical and useful than distributed-CCP.

5.8. Influence of Shape Error. Figure 8 evaluates the influence of shape error on the sensing ratio under the condition without localization error. As shown in the figure, the sensing ratio decreases independently of protocols, and their order does not change against the degree of irregularity. When there are shape errors, a patch considered to be inside the ideal and circular sensing area of an active sensor node is not always inside the actual and irregular sensing area. It leads to decreasing the sensing ratio. On the other hand, even if a patch is covered by a distant active sensor node whose actual sensing area contains the patch, it does not contribute to

the sensing ratio calculated at a sensor node or a sink. It is because another node whose circular sensing area contains the patch decides to become in active state for insufficient coverage from a viewpoint of the sensor node and the patch becomes covered anyway. As a result, the shape error causes deterioration of sensing ratio.

5.9. Evaluation of Energy Consumption. Finally, we evaluate energy consumption of our proposal and CCP. Figure 9 shows the averaged energy consumption per sensor node over 10 simulation runs against time for cases with and without localization error. Results of our proposal with and without localization error overlap with each other. This is because the number of active sensor nodes does not increase even with high localization error. A reason why the global activity-based control requires more energy than the area activity-based control for $k = 1$, similar energy for $k = 2$, and less energy for $k = 3$ is that it requires more, similar number of, and less active sensor nodes for $k = 1, 2$, and 3, respectively, as shown in Figures 6 and 7. On the contrary, in the case of CCP, localization error depletes more energy for the increased number of active sensor nodes (see Figures 6 and 7). For 1-coverage ($k = 1$), the amount of energy consumption with localization error becomes 1.35 times as much as that without localization error, whereas the number of active sensor nodes increases by about 1.8 fold.

Independently of the required coverage, it is apparent that our proposal consumes only between one-sixth and one-third of energy of CCP. The primary reason lies in less communication overhead of our proposal. Our proposal does not involve any additional communication among sensor nodes except for dissemination of activity. Therefore, sensor nodes can turn off transceiver modules except for data gathering and feedback dissemination and hold down energy consumption. On the other hand, CCP consumes energy in the listen mode of transceivers for information exchanges and state transitions. To evaluate the K_s -Eligibility and confirm state transition, a sensor node has to keep a transceiver module listening to a channel for longer duration than our proposal. Furthermore, CCP requires a larger number of sensor nodes to be active than our proposal when there is a large localization error. Because of the smaller energy consumption, our proposals can accomplish the longer lifetime of sensor network than CCP. For example, although the sensing ratio with the area activity-based control is about 0.8 for $k = 3$ and $u = 10$ [m] as shown in Figure 7(a), the lifetime of a sensor network is about six times as long as that with CCP.

6. Discussion

As seen in the results of centralized-CCP, center-point control leads to the robustness against localization error in the number of active sensor nodes. This results in the higher contribution ratio of the centralized-CCP than that of the distributed-CCP. Since our proposal adopts a kind of center-point control, where the activity, expressing the degree of coverage of the whole region or each subarea, is derived at a sink, they have the similar robustness. However, the center-point control alone is not sufficient to explain the reason of higher

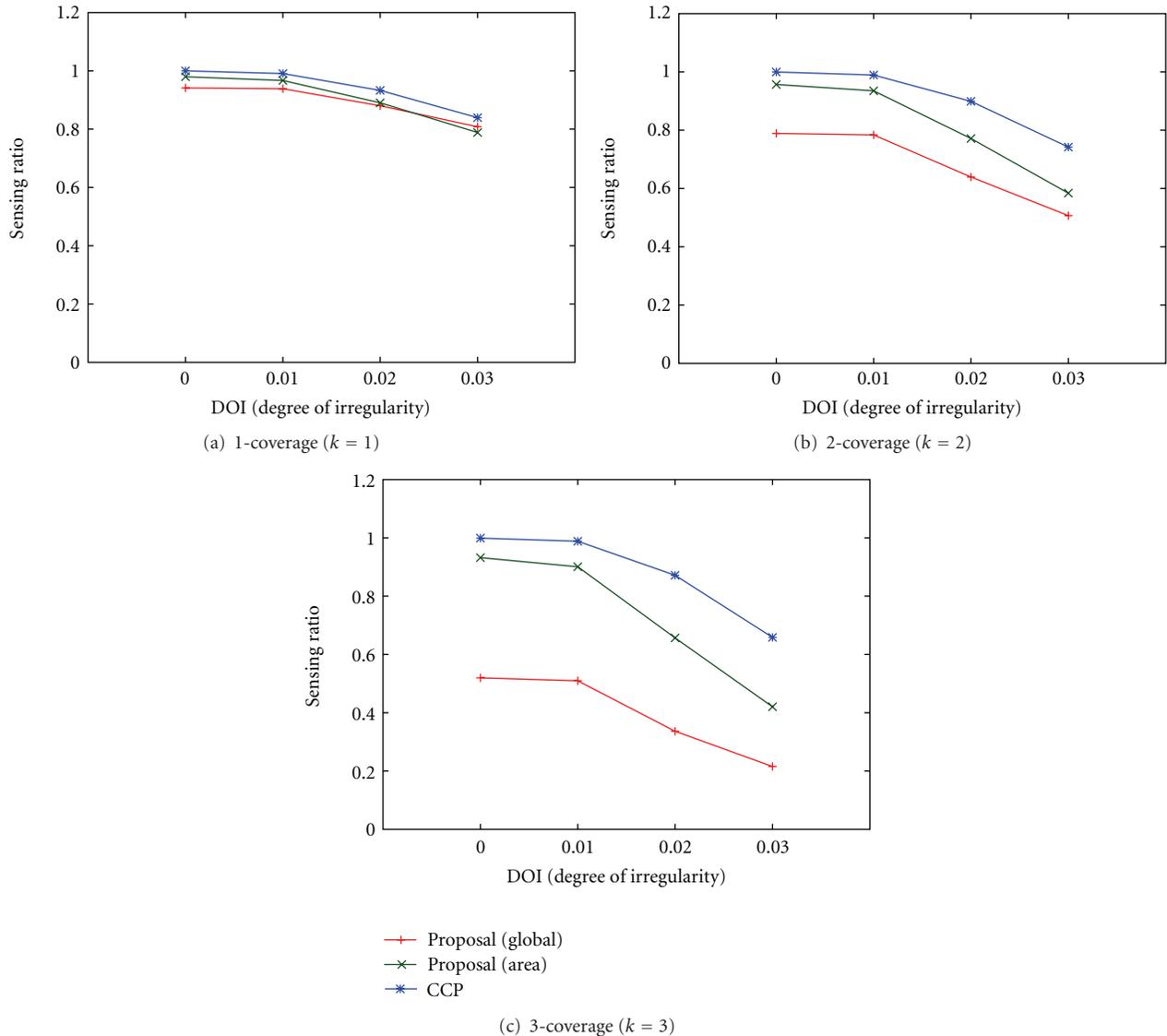


FIGURE 8: Influence of shape error.

performance of our proposal than the CCP-based control schemes. A reason that our proposal can outperform the CCP-based schemes by the smaller number of active sensor nodes is in the bio-inspired algorithm. CCP relies on the deterministic and rigorous algorithm, aiming at the perfect coverage. As a result, many sensor nodes are forced to be active to fully fill out the region with active nodes. For example, a sensor node decides to become the active state to cover a small void, whose size is less than $1/10$ of the sensing area. On the contrary, the bio-inspired algorithm is more flexible and relaxed. A single scalar, called the activity, is used to express the degree of coverage of the whole region or each subarea in a rough and vague manner. In addition, each sensor node decides its state stochastically and autonomously. As such, the number of active sensor nodes is efficiently reduced while leaving some voids uncovered with our proposal, and the sensing ratio is sacrificed to some extent.

7. Conclusion

In this paper, by adopting the attractor selection model of adaptive behavior of biological systems, we proposed an error-tolerant and energy-efficient coverage control and showed our proposal can achieve the sensing ratio S of up to 0.98 and prolong the life time of the network up to 6 fold by comparison with CCP.

As future research, we plan to conduct more realistic evaluation, where radio communication interferes with each other. CCP will suffer from collisions among control messages and the performance will deteriorate. On the contrary, feedback dissemination will be affected by loss of messages. As a result, some sensor nodes cannot update the activity, and the performance will deteriorate as well. We also need to investigate the influence of parameters of the attractor selection model. Biological models are insensitive to parameter

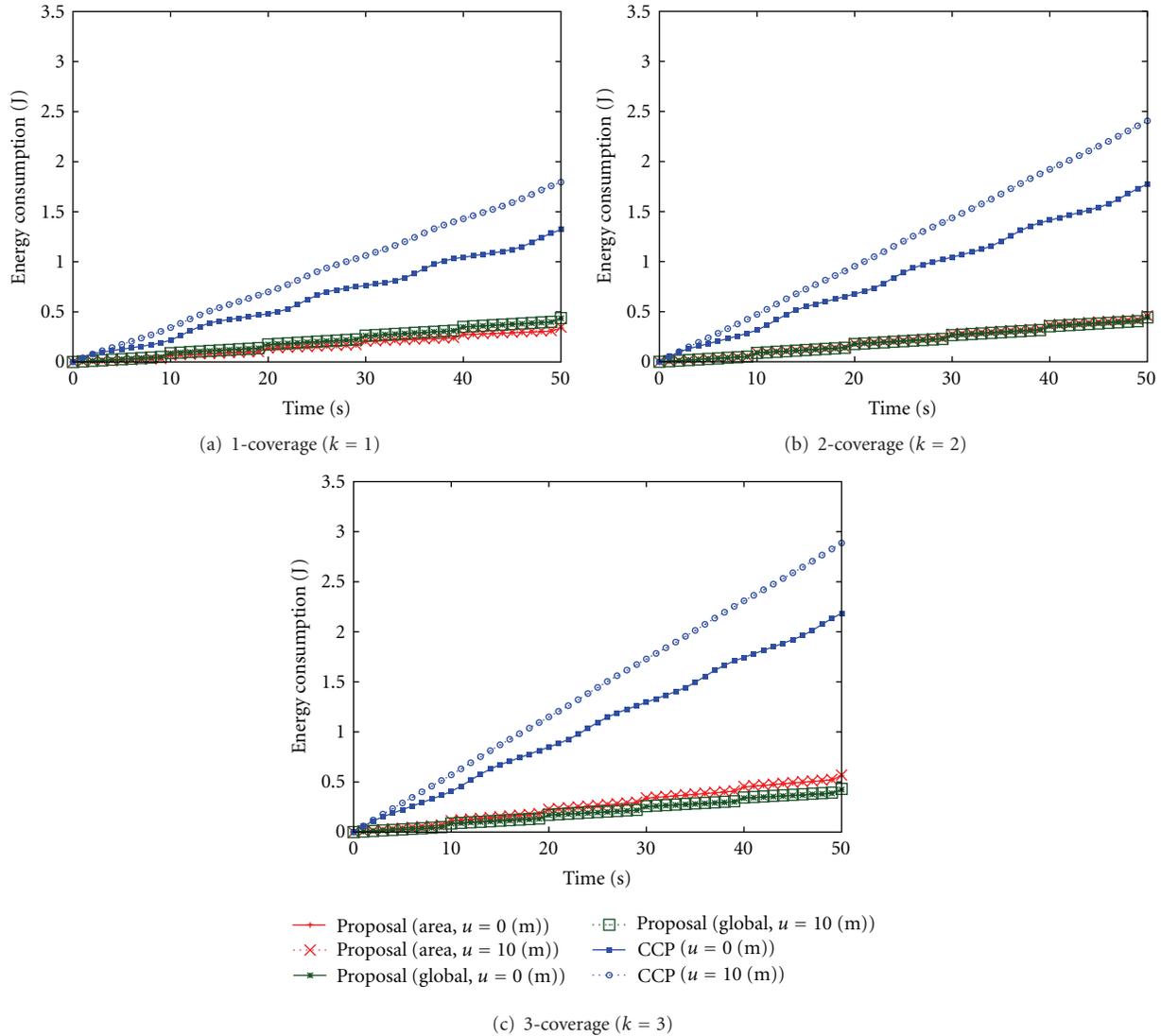


FIGURE 9: Energy consumption.

settings in general and it is one of benefits to be inspired by biological systems. We are going to evaluate our proposals with other simulation scenarios.

Acknowledgments

This research was supported in part by Early-concept Grants for Exploratory Research on New-generation Network and International Collaborative Research Grant of the National Institute of Information and Communications Technology, Japan.

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] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [3] J. Chen and X. Koutsoukos, "Survey on coverage problems in wireless ad hoc sensor networks," in *Proceedings of IEEE South East conference*, pp. 22–25, March 2007.
- [4] M. Cardei, J. Wu, and S. Yang, "Topology Control in ad hoc wireless networks with Hitch-hiking," in *the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '04)*, pp. 480–488, October 2004.
- [5] L. Wang and Y. Xiao, "A survey of energy-efficient scheduling mechanisms in sensor networks," *Mobile Networks and Applications*, vol. 11, no. 5, pp. 723–740, 2006.
- [6] A. Kashiwagi, I. Urabe, K. Kaneko, and T. Yomo, "Adaptive response of a gene network to environmental changes by fitness-induced attractor selection," *PLoS One*, vol. 1, no. 1, article no. e49, 2006.
- [7] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration for energy

- conservation in sensor networks,” *ACM Transactions on Sensor Networks*, vol. 1, pp. 36–72, 2005.
- [8] R. Zheng, G. He, and X. Liu, “Location-free coverage maintenance in wireless sensor networks,” Tech. Rep. UH-CS-05-15, Department of Computer Science, University of Houston, July 2005.
 - [9] B. Yener, M. Magdon-Ismail, and F. Sivrikaya, “Joint problem of power optimal connectivity and coverage in wireless sensor networks,” *Wireless Networks*, vol. 13, no. 4, pp. 537–550, 2007.
 - [10] J. Wang, R. K. Ghosh, and S. K. Das, “A survey on sensor localization,” *Journal of Control Theory and Applications*, vol. 8, no. 1, pp. 2–11, 2010.
 - [11] K. Leibnitz, N. Wakamiya, and M. Murata, “A bio-inspired robust routing protocol for mobile ad hoc networks,” in *the 16th International Conference on Computer Communications and Networks (ICCCN '07)*, pp. 321–326, August 2007.
 - [12] J. Lu, L. Bao, and T. Suda, “Probabilistic self-scheduling for coverage configuration in wireless ad-hoc sensor networks,” *International Journal of IEEE Pervasive Computing and Communications*, vol. 4, pp. 26–39, 2008.
 - [13] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, “Impact of radio irregularity on wireless sensor networks,” *the 2nd International Conference on Mobile Systems, Applications and Services (MobiSys '04)*, pp. 125–138, 2004.
 - [14] T. S. Rappaport, *Wireless Communications: Principles and Practice*, vol. 207, Prentice Hall, Englewood Cliffs, NJ, USA, 1996.
 - [15] Crossbow Technology, “MICAz Datasheet,” <http://www.xbow.com>.
 - [16] V. Shnayder, M. Hempstead, B. R. Chen, G. W. Allen, and M. Welsh, “Simulating the power consumption of large-scale sensor network applications,” in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 188–200, November 2004.

Research Article

SWR: Smartness Provided by Simple, Efficient, and Self-Adaptive Algorithm

Mujdat Soy Turk and Deniz Turgay Altılar

Computer Engineering Department, Istanbul Technical University, Maslak, 34469 Istanbul, Turkey

Correspondence should be addressed to Mujdat Soy Turk, msoy turk@itu.edu.tr

Received 15 July 2011; Accepted 20 October 2011

Academic Editor: Yuhang Yang

Copyright © 2012 M. Soy Turk and D. T. Altılar. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Resource limitations of sensor nodes in wireless sensor networks (WSN) bound the performance on its implementations. Main concern becomes utilizing these limited resources (CPU, memory, bandwidth, battery) as efficient as possible. Their efficiency is mostly affected by the applied routing algorithm, which carries gathered data to inclined/intended destinations. In this paper, a novel routing algorithm, stateless weight routing (SWR), is proposed. The SWR differs from other protocols in many ways. Major feature of the SWR is its simplicity. It is a completely stateless protocol without requiring any network or neighborhood information for routing. This feature decreases packet transmissions and energy consumption dramatically. For reliability, data flows to the sink node over multiple paths. Moreover, nodes have the ability of recovering from voids. Nodes process each packet independently and apply an adaptive approach according to the current conditions. These mechanisms are part of the applied simple routing algorithm, the SWR. The resultant of these features assures flexibility and smartness at nodes and in the network. Therefore, topological changes have a little effect on data packet transmissions. Performance evaluation of the proposed approach shows that the SWR is scalable for WSNs whose topology change instantly and frequently as well as remain stationary.

1. Introduction

Properties of the wireless sensor networks (WSNs), such as energy constraints, limited availability of resources, communicating over unreliable wireless links, and ad hoc property, preclude scalable solutions. Since scalability is mainly affected by the implied routing protocol, many studies and efforts have been put to solve this well-known problem. Among the previously proposed approaches in the literature, geographical routing protocols that make use of location information for routing are more scalable than the others [1–6]. There are essential drawbacks of geographical approaches which can be listed as follows: requiring complex calculations at nodes, depending on the MAC-layer (e.g., IEEE 802.11), being susceptible to local-minima, possessing inability to go around voids, and reliability. Moreover, frequently changing topologies harden the problem mentioned above [6–8].

As the sensor node technologies take place in the market, demands on diverse application areas of WSNs draw on some requirements such as being able to work on different topolo-

gies, staying alive as long as possible without disruptions. Unfortunately, solutions aimed to cover these objectives introduce complex calculations at nodes and require information exchange between nodes which may even spread all over the network. Moreover, some approaches introduce solutions which require use of system-wide parameters that could not be defined clearly and/or deterministically, thus always require man-in-the-loop.

In this paper, a novel stateless data flow approach and routing algorithm, namely, stateless weight routing (SWR) for ad hoc and sensor networks is proposed. It is a stateless and reactive routing protocol utilizing the geographic location information for routing. Nodes do not have to be aware of either local or global topology information. Routing is achieved without keeping tables. Routes are constructed on-demand and spontaneously without requiring any neighborhood information. This approach avoids the beacon messaging and advertising. The delay and complex calculations encountered in geographical routing protocols and the communication overhead encountered in beacon-based

approaches are eliminated in the proposed approach. Moreover, the proposed approach is flexible since it is able to adapt itself according to the network dynamics.

In the SWR, a new metric called weight is introduced that is derived from nodes' own positions to be used in routing process. Weight value simplifies the routing algorithm and makes data flow spontaneously and simultaneously over multiple paths. Multiple paths provide reliability, eliminate the void problem substantially, and provide more robust routes including the shortest path. To keep energy-limited nodes out of the route, the decision to transmit includes QoS parameters such as power-left at the node.

The proposed algorithm:

- (i) provides scalability since neither routing tables nor beaconing is used,
- (ii) simplifies the routing process by designing an appropriate algorithm, which utilizes a weight metric,
- (iii) decreases calculations, delay, and resource requirements (such as processor and memory) at nodes since a weight metric is used instead of time consuming operations on routing tables,
- (iv) decreases energy consumption by:
 - (a) not beaconing,
 - (b) considering the remaining energy levels at nodes,
 - (c) limiting the number of relaying nodes,
- (v) provides reliability by exploiting multiple paths and recovering from voids,
- (vi) executes routing process completely in the network layer, independent of the MAC layer underneath.

The key contribution of the SWR is eliminating the communication overhead and energy consumption required in topology learning approaches. The SWR utilizes resources allowing simultaneous data flows over multiple paths rather than prior topology learning and path construction. The SWR is a self-healing algorithm that a failure in the network or links or nodes does not affect on-going data communications and data flows. With a smart approach, nodes make their own decision according to their own current conditions. Simulations prove that the SWR is scalable even for large-scale networks.

In the next section, related work is given. Design goals are given in Section 3. In Section 4, the proposed routing algorithm, SWR, is described. Analysis of the proposed algorithm is given Section 5. Performance evaluations are given in Section 6. In the final section, the paper is concluded.

2. State of the Art

Routing without tables can be achieved by using location information of the nodes retrieved from GPS or by applying a localization algorithm. In such protocols, which are called geographical routing protocols, nodes know their actual or relative positions with respect to a reference point and share

this information with their immediate neighbor nodes for routing process [1–6, 9–11]. Geographical routing protocols are more scalable with respect to conventional routing protocols due to only local topology information kept at nodes and no or less update overhead.

The taxonomy for position-based/geographical routing algorithms for ad hoc networks is given in [1, 2]. Surveys of the proposed protocols are given in [3–5] and [12]. Formerly proposed position-based/geographical routing protocols use greedy approaches by utilizing either distance or angle as metric [13–18]. In greedy approaches, there is a possibility that they may not find the route due to the constraint of using only local topology knowledge, even if there is a path to destination that can be found with global topology knowledge [19–22]. Besides that, beaconing-based greedy approaches consume excessive energy because of beaconing and introduce control traffic overhead. Furthermore, as the topology changes, providing proactively local topology knowledge reduces the performance and the scalability. Therefore, stateful protocols are not suitable for these types of networks [2]. However, stateless (table-free) protocols are not affected too much from the topological changes and network dynamics, but they use broadcasting to find routes as in flooding which wastes resources. They use MAC-layer integrated approaches to achieve this and introduce delay [10]. MAC-layer integrated approaches make them dependent to the MAC-layer used. In addition to these, some of the geographical routing protocols are prone to void problem. Routing algorithms should be able to cope with the void problem. Approaches and algorithms for void problem in geographical routing in sensor networks are well defined in [19–21].

Six well-known stateless routing protocols providing better performances among all others are described in this section: GPSR [23], SPEED [24], CBF [25], IGF [26], GDBF [11], and DDB [2]. GPSR [23] requires a priori local topology information. Nodes broadcast periodically the beacon messages independent of data packets to provide local topology information. Receiving neighbor nodes update their neighborhood tables accordingly. On a transmission need, best next node is selected by calculating the distances of neighbor nodes. Beaconing introduces communication overhead and consumes energy. Continuous table updating introduces processing overhead and buffers overflow due to periodic beaconing. While the GPSR tries to find the shortest path, it may experience the local minima problem. SPEED [24] is very similar to GPSR. However, it provides real-time communication and recovers from voids.

In contention-based forwarding scheme (CBF) [25], forwarding nodes select the next-hop through a distributed contention process using biased timers. All nodes which receive a packet check if they are closer to the destination than forwarding node and set their timers according to the progression toward the destination. Best suitable nodes respond in advance suppressing the other nodes. Forwarding node selects the best candidate node as next node from the responding nodes set. In this approach, next node selection phase introduces greater delay and energy consumption on route construction phase with respect to greedy approaches. Moreover, rebroadcast decision is based on RTF/CTF

(Request to Forward/Clear to Forward) packets and timers, which are completely processed in MAC-layer. Used energy models are not defined in [25], and it does not consider the energy efficiency.

Implicit geometric forwarding (IGF), which is introduced in [26], is very similar to CBF. As in CBF, it integrates the routing protocol with the MAC-layer, namely, the IEEE 802.11 protocol. It uses RTS and CTS packets with some modifications and additional functionalities. IGF defines a forwarding region which is a destination-directed sector. Each node in the forwarding region sets a response timer regarding the weighted sum of the distance to the destination, the remaining energy, and a random value and competes to acknowledge to the sender. First, acknowledging a node suppresses the others. However, the values of the set timers induce essential amount of delay due to existence of a number of nodes in the forwarding region. Consequently, a packet holder may have to wait for a long time before hearing an acknowledgement back. For the case of absence of a candidate forwarding node, IGF proposes two methods. In the first one, MAC layer informs the network layer to increase the range of transmissions. In the second one, although the use of backpressure method is proposed, no implementation details are given. As IGF induces delay in addition to delay encountered in collision resolution, it is completely dependent on MAC layer IEEE 802.11 protocol and is bounded by the use of modified RTS/CTS packets.

Another beaconless position-based routing protocol that guarantees the delivery of the packets, namely, guaranteed delivery beaconless forwarding (GDBF), is proposed in [11]. GDBF protocol selects appropriate next node by means of RTS/CTS packets. Forwarding a node broadcasts the RTS packet to its neighbors and the neighbor nodes compete with each other to forward the packet and set a timeout depending on their suitability. After timeout, nodes send CTS back to the forwarding node by using the suppression technique. Forwarding node decides one of the neighbor nodes as next node and forwards the message to it in a greedy manner. In case of a failure in this greedy mode such as reaching to local minima (no CTS response), guaranteed delivery is provided by the recovery mode. The drawbacks of the GDBF are as the same as the ones in the CBF. On the other hand, GDBF is a solely MAC-layer solution for the routing.

CBF, IGF, and GDBF are very similar to each other in terms of next node selection. A different approach which is called dynamic delayed broadcasting Protocol (DDB) is proposed in [2]. DDB allows nodes to make locally optimal rebroadcasting decisions by dynamic forwarding delay (DFD) and allows the nodes that have higher retransmission probability to rebroadcast first by suppressing the transmissions of other nodes. However, it cannot avoid multiple transmissions and introduces delay. On the other hand, during each receive operation; nodes have to recalculate/adjust their timers, which is computationally complex. Since packet scheduling is achieved in the MAC layer, on each receive operation an additional MAC-Network-MAC inter-layer communication is required to reschedule the packet transmission. A scheduled packet can even be dropped

following a number of calculations and scheduling operations, which are costly in terms of time and energy.

These beaconless and stateless algorithms introduce MAC-layer-involved solutions for routing. They rely on MAC layer and utilize IEEE 802.11 protocol functions for routing decisions. Note that timing and packet scheduling are the functions of MAC layer while decisions of broadcast, multicast, and unicast are the functions of network layer. In well-defined communication architecture, routing and node addressing should be independent from the MAC layer functions. Combining routing function with MAC layer introduces overhead and makes the routing protocol dependent to the MAC scheme used within the system.

These stateless algorithms integrate MAC layer to select forwarding node based on calculated timer values to schedule packets. This approach introduces a computational overhead in MAC/Network layer. Moreover, scheduling is triggered whenever a node receives a new packet. Consequently, timing and scheduling changes very frequently. Worse is that scheduling algorithm reduces the performance of the network layer by disturbing it for the sake of routing. Moreover, there are some unconsidered cases in these protocols, such as unforeseeable erroneous timer setting, deceive of the validity of applied timer value in collision resolution, and unpredictable terminations of nodes. In these protocols, timer is generally set according to node's geographical position and a number of node and system-wide-parameters. The idea is to find a "forwarding node" which is supposed to be best among all candidates considering the given set of parameters. Although the selected node may be considered as the optimal "forwarding node," contention based problems has to be rediscussed in detail to verify the optimality. When a node receives a packet, packet encounters a delay in buffers. This delay includes queuing delay and processing delay and varies according to nodes' current conditions including in-buffer situation, processing ability, out-buffer situation—current medium conditions which affect the propagation of a packet, and collision resolution delay in MAC layer. Therefore, a better node that is expected to calculate shorter time may process the packet later than others. In addition, when the packet is passed to the MAC layer, depending on the state of out-buffer of the node, packet may not be transmitted immediately and encounters delay along with possible delay in collision resolution. These all affect the timers calculated at nodes and make them inaccurate. Therefore, selected forwarding node may not be the expected "optimal" node. Added delay according to the timer in packet scheduling is an addition to the delay encountered in collision resolution in MAC layer. It increases the overall delay during transmissions. Moreover, the proposed stateless protocols' performance is sensitive to the node terminations and nodes' unpredictable come-ups and go-downs. Making data flow over a single path is prone to failures at any time, which is very common in frequently changing topologies. Some of these protocols are also prone to void problem. Routing algorithms should be able to cope with the void problem as well.

3. Design Goals

Five essential design goals are considered in this research, which are:

- (i) providing simple and efficient architecture that comprises of flexibility, rapid deployment, low configuration-setup-management, and high data reception rate and accuracy,
- (ii) providing reliability for a number of different severe circumstances and conditions such as low density deployment, existence of voids, node terminations, link failures, and frequent topology changes,
- (iii) producing an uncomplicated, robust and scalable algorithm that works with modest resources of disposable nodes,
- (iv) enabling nodes to adapt themselves to changing network conditions such as variable network density, void or congestion occurrences, and link or node failures,
- (v) providing a steady delivery speed where the end-to-end delay is proportional to the distance between the source and destination.

According to defined design goals given above, the proposed approach satisfies the following design objectives.

3.1. Stateless Architecture. Physical limitations are the main constraint of wireless sensor networks. To cope with this constraint, the SWR algorithm is designed to minimize the use of resources such as memory and processor. Neither routing tables nor any information on topology are not kept at nodes yielding memory requirement to be kept at minimum. The SWR does not require routing table, beaconing, or any information exchange that is unaffordable in other protocols. This approach also reduces the processing overhead at nodes. Complex calculations existing in stateless approaches are not required in the SWR. Thus, CPU requirements are kept at minimum as well. The SWR is designed as a very simple algorithm that can be executed at sensor nodes having the lowest possible processing capability.

3.2. Reliability. Routing algorithms that use a single path to the destination suffer from route breakages and packet losses. Using multiple-paths provides reliability but introduces communication overhead to build and maintain these paths up to date. In the SWR, data flow spontaneously and simultaneously over multiple-paths without excessive energy consumption and communication overhead. Path construction, maintenance and update are not required in the SWR. The algorithm is simple and inherits the benefits of flooding. Moreover, mechanisms to recover from voids and congestion are developed as defined below.

3.3. Void Avoidance and Recovery. Local-minima problem dictates that greedy algorithms may fail to find a path to the destination even when one or more paths actually exist. These algorithms may also experience void problem. Only few of those [19–21] propose solutions for local-minima

problem and to recover voids. Although the SWR does not suffer from local-minima problem, it utilizes a void recovery algorithm to pass around large void areas [27] if and only if such a problem is not overcome implicitly by the use of multiple paths.

3.4. Robustness to the Congestion. Reactive protocols and stateless protocols construct routes on demand. Dynamic nature of these protocols may cause fluctuations in network traffic, which may also introduce additional congestion. Moreover, coping with congestion introduces additional overhead and reduces performance. Routing algorithms in WSN generally use the backpressure mechanism to cope with congestion. In the SWR, the above-mentioned void recovery and avoidance approaches inherently solve the congestion problem.

3.5. Scalability. As the size of the network increases, the performance degrades dramatically in terms of throughput, reliability, delay, and energy-efficiency. The SWR algorithm presents a scalable solution for WSNs by avoiding table-keeping at nodes, avoiding information exchange between nodes, and utilizing locally implemented approaches to avoid/recover from problems such as void and congestion.

3.6. Large-Scale Applicability and Multiple-Sink Usage. Using multiple sinks has been proposed as a feasible approach to overcome scalability issues in large-scale WSNs in the literature [28, 29]. However, most of the protocols are based on previously proposed ones developed for single sink networks. Therefore, adapted such multisink protocols inherit the deficiencies of single sink network protocols such as scalability. However, the SWR algorithm can be used with either multiple sinks or a single sink without any adaptation or modification in the routing algorithm at all.

3.7. Self-Adaptation. Protocol design considerations for link problems, communication environment, node failures, and mobility require management. Management solutions include information exchange between neighbor nodes or dissemination of control information throughout the networks. Solutions may include observation of the designed network and manipulation on the parameters to provide suggested performance. The SWR algorithm provides this key feature by adapting itself dynamically to changing conditions with no user involvement and no information exchange between nodes. Note that there is no control packet in our protocol design. Nodes take their own decisions by considering the current network conditions and the parameter values via previously received data packets. This feature, in addition to the others given above, makes the SWR a unique routing algorithm among the ones designed for WSNs.

3.8. Traffic Load Balancing. Traffic load balancing is another challenging requirement in wireless sensor networks. As known balancing, the traffic load at nodes prolongs network-wide connectivity. The SWR algorithm uses spontaneous data flow approach in which a node is participated in routing process according to the availability of its own resources and

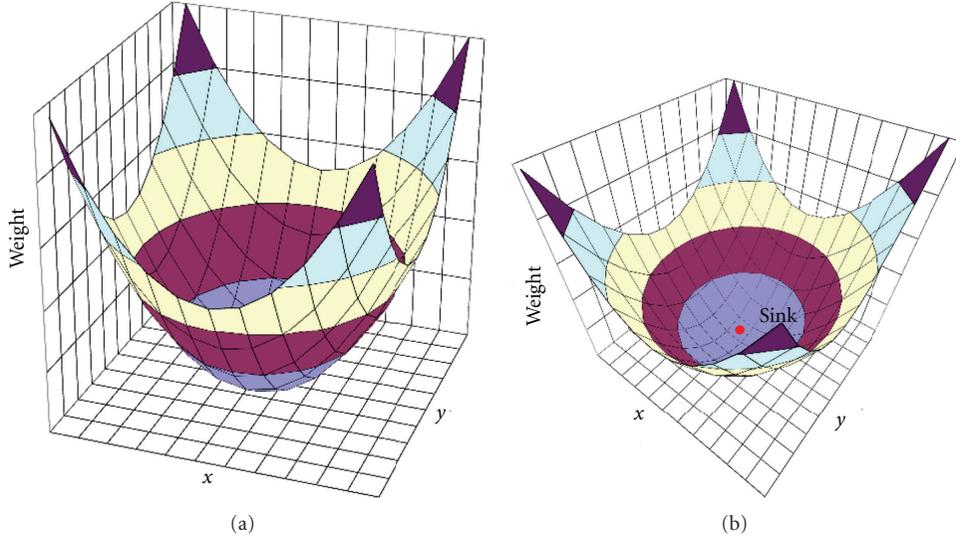


FIGURE 1: Weight metric using only Euclidean-distance provides a spontaneous flow toward the sink.

the parameters received within the data packet. Therefore, the data may flow from source to destination over different paths. Note that congestion avoidance provided with the SWR supports load balancing as well.

3.9. Localized Behavior. System-wide operations, such as beaconing or topology, update messaging that yields high message traffic and reduce the performance of the networks. Also, problems such as congestion and void have to be sorted out locally to preserve scalability. The SWR algorithm uses localized distributed operations to deal with such problems. Each node makes its own decision in an isolated manner according to its own knowledge and availability of resources. Note that the SWR algorithm does not use periodic beaconing which also contributes to the localized behavior.

4. Stateless Weight Routing Algorithm

The stateless weight routing (SWR) is a stateless and reactive routing protocol that utilizes the geographic location information for routing. Routing tables and local/global topology information are not kept at nodes. Nodes do not exchange information prior to sending packets, and they do not even need to know the identities of their neighbors. Routes are constructed on-demand spontaneously without requiring any neighborhood information. Routing is achieved with aid of *weight* values of nodes (w_i , for node i), which is derived from the geographical positions of the node and a number of parameters (1). These parameters may belong to either the node itself (parameters _{i} , such as energy left at the node), overall network system (parameters_{network}, such as network's instantaneous situation), or a combination of these two sets of parameters

$$w_i = \text{location}_i + \text{parameters}_i + \text{parameters}_{\text{network}} \quad (1)$$

For the case in which only the location comprises of the weight function, the weight value indicates the square of the

Euclidean distance to the sink node. Regardless of the content of the weight function, nodes away from the sink node have usually greater weight values with respect to closer ones, whereas the sink has always a weight value of 0. A generic weight diagram only considering the distance from the sink is shown in Figure 1, where the sink is positioned in the center of the area. This approach provides spontaneous natural data flows from nodes toward the sink when data is sent according to weight values of nodes. Weight values are not exchanged between nodes separately, but inserted into data packets. The use of weight metric makes the routing process simple and minimizes delay, energy consumption, and processing requirements at nodes in routing decision phase.

4.1. Making Data Flow over Multiple Paths. The SWR utilizes multiple simultaneous paths on data transmissions. The source node inserts its weight value into the packet and broadcasts. When a node receives a packet, it compares its own weight value with the weight value in the packet. Any node whose weight value is smaller than the transmitting node's weight value rebroadcasts the packet after replacing *Sender ID* and *Sender Weight* fields with its own values. The packet is dropped otherwise. In order to limit both the number of transmissions and the number of multiple paths, a *Threshold* value in terms of weight metric is used. Only the nodes have weight difference greater than the threshold value can rebroadcast the packet. By this way, nodes closer to transmitting node are avoided to rebroadcast. Rebroadcasting nodes are those that make more advances toward the destination. As seen in Algorithm 1, Euclidian distances to neighbor nodes are not calculated. Rather, only the weight value of the sending node that is retrieved from the packet header and the node's own weight value that is already known are compared. $w(i)$ defines the weight of node i .

The *Threshold* value in terms of weight, which is also inserted into packet sent, gets value between 0 and r , where r is the transmission range known by all nodes. *Threshold* value

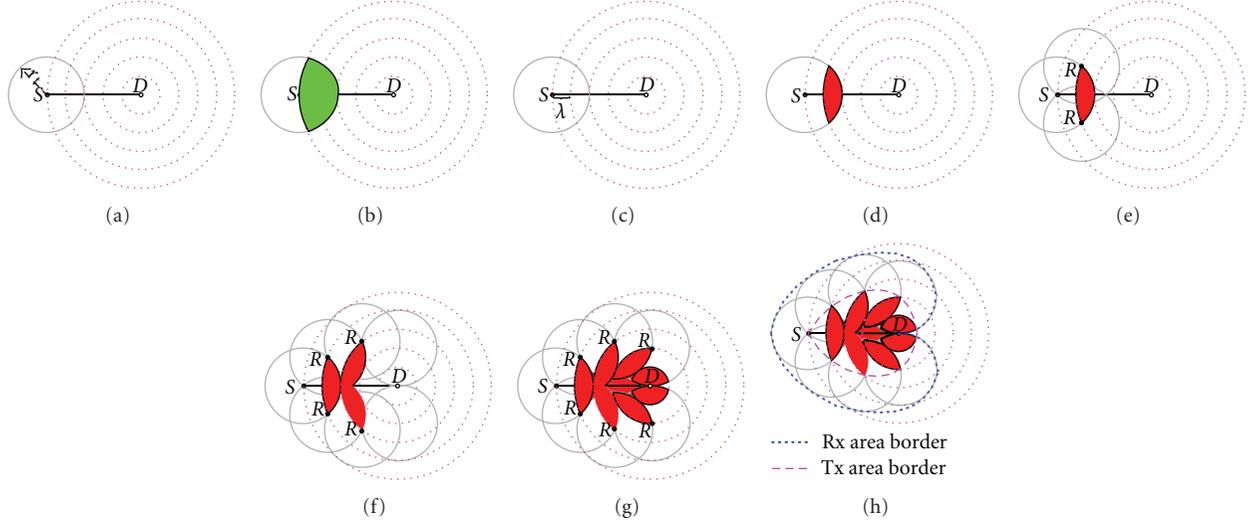


FIGURE 2: Possible transmission and receive areas between the source node S and destination node D .

```

if (( $w(\text{sender}) - w(i) \geq \text{Threshold}$ ) then
    rebroadcast;
else
    drop the packet;

```

ALGORITHM 1: Pseudo code of data flow algorithm for node i .

is used for three purposes: to reduce energy consumption by regulating the number of transmitting nodes; to adjust the number of possible multiple paths; and to recover from voids. The *Threshold* value limits the number of retransmissions by avoiding relatively closer nodes to retransmit. This approach has two favorable outcomes: first of all, avoiding the closer nodes to transmit reduces the energy consumption; second of all, making more nodes to transmit rather than only the farthest one, which is prone to the link failures, constructs more robust paths.

4.2. Coverage Area. Figure 2 explains the covered area on transmissions. Assume that source node S has a data packet to send to the destination node D (Figure 2(a)). If conventional geographical routing algorithms were used, on the transmission of S , the nodes positioned in the shaded area between the S and D would be candidate retransmitting nodes (Figure 2(b)). In conventional greedy algorithms and stateless algorithms, *selection of next retransmitting node* schemes is used. In our approach, there is no such node selection scheme running at the sender. A node decides to relay or to drop the packet itself. The number of candidate retransmitting nodes is reduced inherently by using the threshold value as seen in Figures 2(c) and 2(d). All these candidate nodes can retransmit the received packet applying the Algorithm 1. Figures 2(e)–2(g) shows the covered area after successive transmissions until the destination D is covered. The covered area in Figure 2(g) shows the *worstcase* scenario since the outermost edge nodes (R) are selected as the retransmitting nodes.

In Figure 2(h), the covered area is bordered with a dashed line to show the maximum possible transmission area, while the area bordered with dots shows the possible area that is affected after these transmissions (reception area). Mathematical model of the presented approach and its analysis are presented in Section 5.

4.2.1. Reducing Transmissions by Adjusting Threshold Value. The threshold value is adjusted to save energy by limiting the number of retransmitting nodes. Figure 3 shows the covered area after multiple successive transmissions between the source node S and the destination node D when Algorithm 1 is applied with different threshold values. Threshold values 50% and 85% of r are used in Figures 3(a) and 3(b), respectively. As seen, increasing the threshold value reduces the number of candidate retransmitting nodes.

4.2.2. Adjusting Threshold Value in Dense Topology. The default threshold value can be adjusted according to the node density in the network. In dense networks, the threshold value can be set to be high by default to limit the retransmitting nodes. In non-dense networks, the default threshold value can be set to be small value to allow enough nodes to participate in data flow. In dense topologies, sleep scheduling is a typical approach to reduce transmissions and energy consumption. Sleep scheduling can also be applied to the presented approach but it is not covered within the scope of this paper.

4.3. Data Packet Transmissions. If a node has a data to send the sink, it inserts its identification number, current packet sequence number, and the intended destination (here it is sink node)'s identification number into the appropriate fields. Relaying nodes do not change these three values until the packet reaches its destination. The sender inserts also its identification number and the current weight value into the *Sender ID* and *Sender Weight* fields, respectively, where these values are changed by the relaying nodes to their own values.

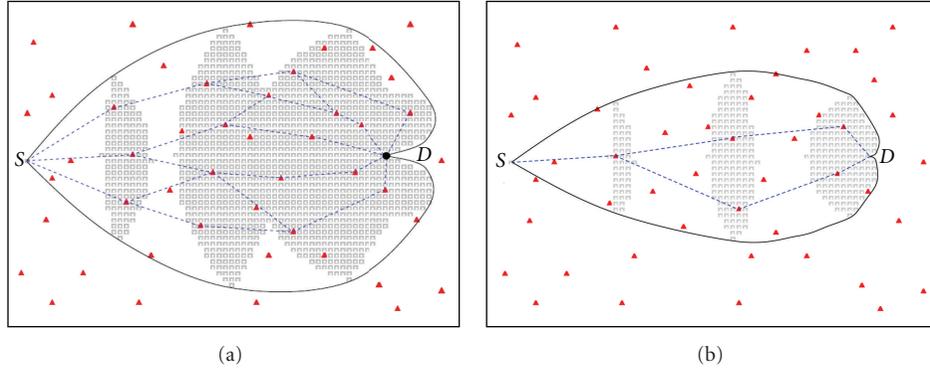


FIGURE 3: Possible retransmission area and candidate nodes and paths.

Threshold field is set to system-wide default value which is actually 50% of r , but can be changed according to the network dynamics. Then the node broadcasts the packet. Actually, the packet is passed to the MAC layer to be sent to the addressed nodes. Relaying intermediate nodes only make changes on *Sender ID* and *Sender Weight* fields on their transmissions. In case of a void, each relaying node also makes its own decision to change the *Threshold* value according to its needs. This approach is described in Section 4.5.

4.4. Reliability. Using multiple paths is the most reliable method to convey the data to the destination. In MANET and WSN, paths should be constructed on-demand due to frequently changing topology and propagation conditions. However, almost all of the proposed protocols in the literature [30–33] have a route construction phase and these multiple paths are constructed before sending the data packet. In these approaches, generally the packet is sent over the primary path. In case of a route failure, the packet is sent over the alternate path. Switching from the failed path to the alternate one introduces an additional delay. Failed paths cause packet drops and retransmissions, exacerbating the delay. If all paths known in-advance failed, a new route recovery or route reconstructed is required, which increases the delay longer.

Reliability in the SWR is provided by using multiple paths. Contrary to the known protocols in the literature, these on-demand paths are not constructed ahead of packet sending. Algorithm 1 spontaneously constructs simultaneous multiple paths while the packet is on the way toward the sink. Data packet is simultaneously carried over every constructed path. Information about paths is not kept at nodes for future use. Keeping such path information is unnecessary in frequently changing topology and introduces overhead and delay as depicted above. Therefore, comparing to the protocols using multiple-paths, the SWR provides the minimum delay.

The use of simultaneously active multiple paths provides a continuous connection in the case of a broken path. While the spontaneous dynamic data flow is regenerated at the broken link of the broken path, the data packet, meanwhile, is carried over other paths. This approach naturally eliminates the problem of a possible breakage in end-to-end communication. Moreover, simultaneous establishment of multiple

dynamic data flows naturally isolates problems triggered by a single node failure. Thus, route breakages are not observed in the SWR.

The number of the paths depends on the distance (length in hops) and the applied threshold value. As the distance increase, the number of the constructed paths increases. As a result, for the same source-destination pairs, data packets may follow different paths. Considering the construction and use of multiple-paths, the SWR provides high reliability.

Multiple paths exploited in data flow are the braided multiple paths. They overlay with each other in some part and utilize the advantages of the best path. One design measure of SWR is providing guaranteed delivery. Simultaneous data flow over multiple paths in SWR substantially achieves this goal. In case of a failure in data flow, void recovery algorithm (Algorithm 2) is invoked. If a node experiences void, it rebroadcasts the packet with a reduced threshold value. This approach allows the sender's neighbors, which have the same or smaller weight values, to integrate into flow process. Further step for on-going void problem is using a fake weight value instead of actual weight value of the sender. This second approach allows all neighbors to participate in flow process. These two approaches increase the covered area of the data packet sent, building more paths on-demand. Actually, void recovery is not a separate process; it is inherently done in route construction. Although, the void recovery is only invoked for the data flow that encounters the void, the data packet continues to flow over other paths, meanwhile. Figure 4 illustrates the constructed multiple paths and sub-paths in void recovery. In Figure 4(a), there are k different braided multiple paths from source to destination, which are constructed spontaneously and simultaneously. If a node on a possible path cannot forward the packet (node 2, n_2 , on path 1, P_1 , in Figure 4(a)), alternate subpaths are constructed spontaneously and simultaneously as shown in Figure 4(b). These subpaths may overlap as well with other paths, as defined above.

4.5. Void Avoidance and Recovery. Void and coverage hole problems and related studies in the literature are summarized in [19–21]. Only a few routing protocols in the literature, propose solutions to recover from voids. The methods used in these solutions include the use of multiple paths and

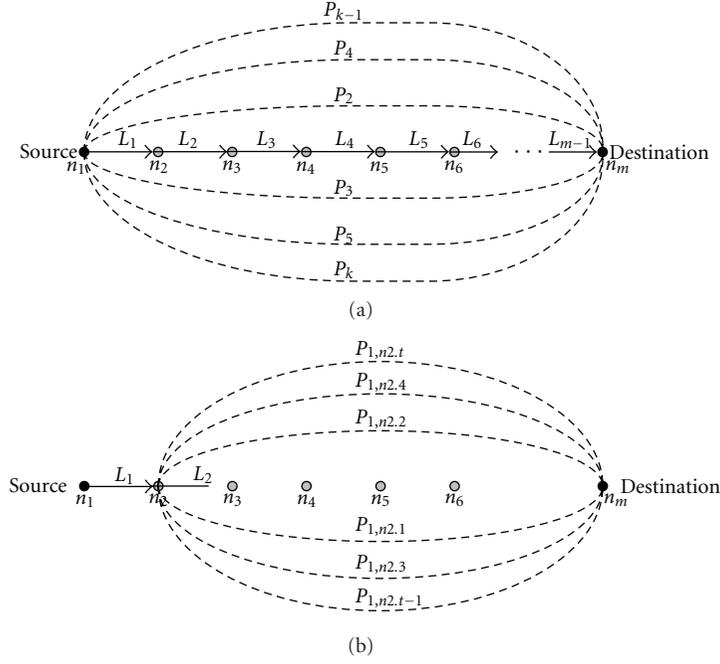


FIGURE 4: Multiple paths exploited in SWR are braided paths. Void recovery algorithm provides construction of new paths.

```

if (Threshold > 0) then
  set Threshold to 0 (zero);
  rebroadcast;
if (the packet cannot be relayed) then
  set the  $w_{\text{sender}}$  to  $w_{\text{sender}} + w'$  in header;
  rebroadcast;

```

ALGORITHM 2: Void avoidance algorithm.

alternating paths, retransmissions, broadcasting, flooding or localized flooding, and discovery of the voids and boundary of voids. Unfortunately, these protocols present poor performance, especially when the topology changes frequently. Reasons of performance decrease can be explained as follows. These methods are highly dependent on the use of information exchange between nodes due to topology information requirement at nodes to recover from voids. Success, effectiveness, and responsiveness of these methods depend on the frequency and reliability of the retrieved topology information. Therefore, a tradeoff exists between the provisions of the topological information and the accuracy of this information [7, 16, 22]. Frequently exchange of information consumes energy and introduces communication overhead. On the other hand, infrequent information exchange causes the nodes have unreliable topology knowledge.

Moreover, some protocols such as the stateless geographical routing protocols propose solutions to be implemented at the MAC layer and generally have local-minima problem. Recovery from voids using such protocols is too complex and costly. With respect to these methods, simple and efficient void avoidance and recovery methods are provided in the

SWR. The proposed methods are peculiar to the SWR and guarantee the delivery of data to the destination.

4.5.1. Implicit Void Avoidance Approach. One use of *Threshold* value is for void avoidance. Increasing the threshold value provides fewer nodes in number to be able to relay the data packets, and decreasing the threshold value provides more nodes in number to be able to relay the data packets (Figure 3). In case of void detection, the transmitting node decreases the threshold value allowing more nodes to apply the data flow algorithm (Algorithm 1) as depicted in Figure 5. Nodes can understand the existence of a void by the nonretransmission of the packet with the same parameters by the nodes that have lower weight values. Adjusting the threshold value adjusts the number of multiple paths. Implicitly, the void problem is eliminated substantially due to utilizing multiple paths. For the case of large gaps in the topology, a void elimination algorithm (Algorithm 2) is proposed to solve the void problem.

4.5.2. Explicit Void Avoidance Algorithm. Threshold value introduces limitations on data dissemination area. However, if the void-experiencing node cannot deliver the packet to the recipients due to large size of the void, an explicit void elimination approach is used. On encountering a void, the node executes the void elimination algorithm (Algorithm 2). The algorithm consists of two steps. In the first step, the algorithm tries to transcend the void by decreasing the threshold value to 0 (zero). Therefore, larger area can be covered to forward the packet. If the packet still cannot be forwarded due to void, the second step is performed. Transmitting node retransmits the packet with a weight value greater than its own weight embedded into the sender's weight field in the packet and the

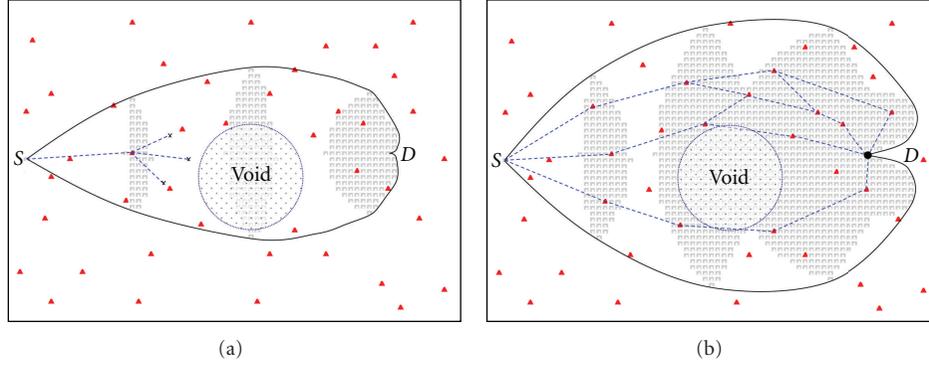


FIGURE 5: Void recovery in SWR.

threshold value set to 0 (zero). Assuming that w is the weight of the transmitting node and w' is the additional weight to be added where $w' > 0$, the new value for weight would be $w + w'$. By increasing the original value for weight, the transmitting node enforces the rearward nodes to participate into the routing. Therefore, a void can be passed by without any complex calculations.

5. Analysis of the SWR Algorithms

In this section, the analysis of the SWR algorithm in terms of the number of transmissions, energy consumption and reliability is presented. Compared to the other geographical routing algorithms, the SWR algorithm does not use beaconing, but utilizes multiple simultaneous paths on data transmissions. First, the transmissions and then the energy consumption are analyzed.

5.1. Analysis of the Transmissions. Geographical routing algorithms send beacons periodically to inform neighbors. At an instant time, $1/f_{\text{beacon}} \times N$ beacon transmissions occur, where f_{beacon} is the beacon frequency and N is the number of nodes in the network. During a time interval T ,

$$\begin{aligned} \text{Total Transmissions} &= \text{Beacon Transmissions} + \text{Data Transmissions}, \\ \text{Total Transmissions} &= \left(\frac{T}{f_{\text{beacon}}} \times N \right) \\ &+ (\text{number of generated packets} \times \text{hop count}), \end{aligned} \quad (2)$$

where hop count is assumed as an average path length.

On the contrary, in the SWR, only data transmissions occur. The number of these data transmissions varies according to the distance between the source and destination, and the parameters applied in routing algorithm such as threshold value. It is not possible to calculate exact number because of spontaneous data transmissions over multiple paths. However, maximum number of transmissions can be found analytically by considering the covered area on transmissions (Figure 2(h)). Since each node can only transmit once, total

transmissions are equal to the covered area multiplied with node density (ND). Figures 6 and 7 show the analytical representation of the covered area. This area is found approximately with finding the area of each triangle, starting from the triangle near source (1st triangle) and ending with the K th triangle close to sink

$$\text{Covered Area} = 2 \times \sum_{i=1}^K A_i, \quad (A_i \text{ is the triangle area } i), \quad (3)$$

where K is the amount of triangles that covers the area and can be found as

$$\begin{aligned} K &= \max \left\{ n \mid \left(\sum_{i=1}^n \theta_i \right) \leq 180^\circ, \right. \\ &\left. \cos \theta_i = \frac{(d_i - \lambda)^2 + d_i^2 - r^2}{2(d_i - \lambda)d_i}, \quad d_{i+1} = d_i - \lambda \right\}, \end{aligned} \quad (4)$$

where d_1 is the distance between the source and the sink, r is the transmission range, and λ is the threshold value.

Total transmissions in the SWR is

$$\begin{aligned} \text{Total Transmissions} &= \text{Data Transmissions}, \\ \text{Total Transmissions} &= \left\{ \left(2 \times \sum_{i=1}^K A_i \right) \times ND \right. \\ &\left. \times \text{number of generated packets} \right\}. \end{aligned} \quad (5)$$

The calculated area will always be greater than actual covered area. Gaps on the actual covered area (Figure 2(h)) are included in the analytical calculation (Figure 7). Comparison of the calculated and actual covered areas is given in Figure 8. Actual covered area has been found by simulations and considering the worst cases. Figure 8 shows that there is a great difference when there is a high threshold value. With high threshold value, far distant nodes will be selected. In this case, the gaps between the successive transmitting nodes will

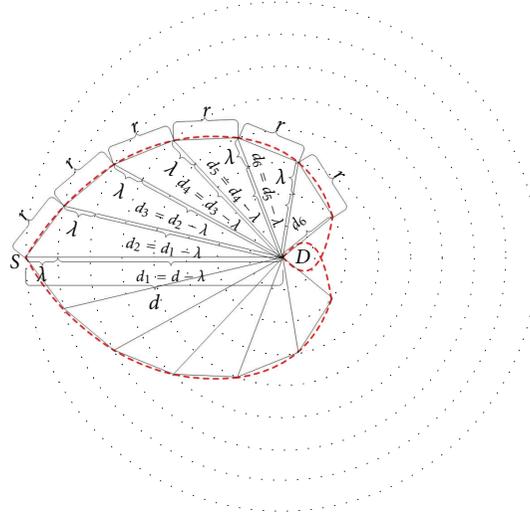


FIGURE 6: Covered area calculation for SWR.

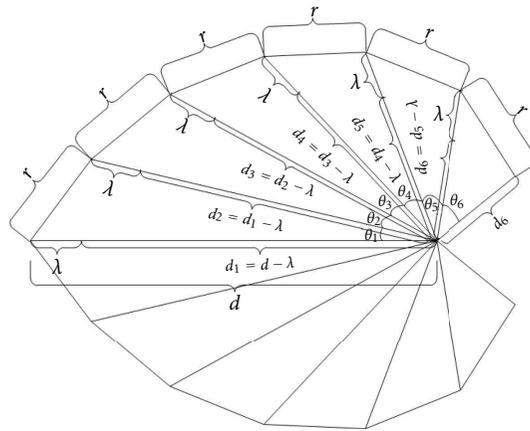


FIGURE 7: Covered area calculation for SWR. Transmission range circles are removed from Figure 6.

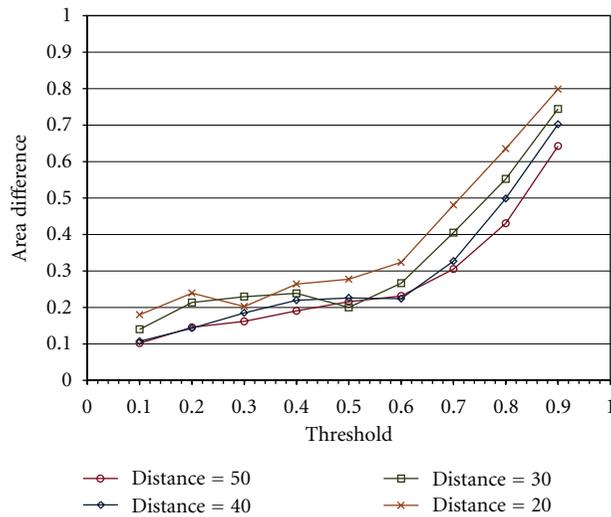


FIGURE 8: Comparison of covered area differences between the calculated area and the actual worst case covered area. In this comparison, distance between the source and destination pair varies between 20 m and the transmission range is $r = 10$ m. Threshold value changed to observe the variations between calculated and actual covered area.

be greater. Since the calculation with triangulation includes these areas, the difference gets higher. Experiments support these results. It was observed in the SWR that number of transmissions are much less than the calculated ones. Compared to the other geographical routing algorithms, this outcome affects energy consumption at nodes gratifying the SWR algorithm.

5.2. Analysis of Energy Consumption. Routing algorithms in the literature propose approaches to minimize the energy consumption, but consider only energy consumption on transmissions. Energy consumptions on receptions, calculations, and sensing are not involved or modeled. Regretting these energy consumptions, especially energy consumption in receive process, outcomes unrealistic performance results. Related studies in energy consumption [34–37] emphasize that receive process consumes as much power as the transmission process. Then the total energy, E_{total} (6), consumed by a node at an arbitrary time is the sum of these energy requirements [34]. It is also defined in [34] that efficient sensing circuitries and computation algorithms reduce E_{sensing} and $E_{\text{computation}}$ substantially. Therefore, they are considered as constant values

$$E_{\text{total}} = E_{\text{transmit}} + E_{\text{receive}} + E_{\text{computation}} + E_{\text{sensing}}. \quad (6)$$

On a transmission, transmitting node consumes the energy, E_{transmit} , and a receiving node consumes the energy, E_{receive} . If the transmitting node has n neighbors, the overall system consumes the energy, E_{network} , for one transmission;

$$E_{\text{network}} = (n \cdot E_{\text{receive}}) + (1 \cdot E_{\text{transmit}}). \quad (7)$$

If it is assumed that $E_{\text{transmit}} \approx E_{\text{receive}}$, the overall system consumes $(n + 1)E_{\text{transmit}}$ for only one transmission. Neglecting such an amount of energy consumption causes unreliable system performance results. According to (7), beacon-based geographical routing protocols consume most of their energies in the beaconing processes.

The energy consumption in a system, during a time period, T ,

$$E_{\text{network}}(T) = E_{\text{beaconing}} + E_{\text{events}}. \quad (8)$$

Note that $E_{\text{beaconing}}$ is only consumed in beacon-based protocols. Considering the transmissions in (2), overall energy consumption becomes

$$\begin{aligned} E_{\text{network}}(T) &= \left\{ \left(\frac{T}{f_{\text{beacon}}} \times N \right) \right. \\ &\quad \left. + (\text{number of generated packets} \times \text{hop count}) \right\} \\ &\quad \times \{(n \cdot E_{\text{receive}}) + (1 \cdot E_{\text{transmit}})\}. \end{aligned} \quad (9)$$

The SWR protocol consumes energy only on data transmissions. Nodes that remain in the shaded area retransmit the received packet for only once. The maximum energy

consumption in data transmissions for the SWR can be found by multiplying (5) and (7)

$$\begin{aligned} E_{\text{network}}(T) &= \{\text{Covered Area} \times \text{ND} \\ &\quad \times \text{number of generated packets}\} \\ &\quad \times \{(n \cdot E_{\text{receive}}) + (1 \cdot E_{\text{transmit}})\}. \end{aligned} \quad (10)$$

$$\begin{aligned} E_{\text{network}}(T) &= \left\{ \left(2 \times \sum_{i=1}^K A_i \right) \right. \\ &\quad \left. \times \text{ND} \times \text{number of generated packets} \right\} \\ &\quad \times \{(n \cdot E_{\text{receive}}) + (1 \cdot E_{\text{transmit}})\}. \end{aligned} \quad (11)$$

Comparing energy consumption (9) and (11), it is seen that, as the covered area in the SWR remains smaller than the network area, the SWR will consumes less energy than any beacon-based protocols. Compared to the any other protocol, for a generated data packet, the SWR consumes less energy when nodes in the covered area in the SWR are less than the total transmissions to transmit a packet including the control and management packets in other protocols. The SWR, on the other hand, carries the data packet on multiple paths to provide reliability. The SWR utilizes the energy to provide reliability rather than topology learning, route construction, and maintenance.

5.3. Analysis of the Reliability. There is a number studies in the literature that analyze the reliability issues in MANET and WSN. In [30], an analytical framework is developed to characterize the random behavior of a multihop path and derive path metrics to characterize the reliability of paths, modeling and analyzing the mean path duration, and the path persistence. Supporting results are provided in [31] with experiments. They characterize link reliability measures in an actual sensor network setting and analyze how reliable data transfer mechanisms impact overall path reliability. For the path P which consist of $|P|$ links and where $\mathcal{P}_{i,j}$ is the probability of a failure on a transmission link $l(i,j)$, the success probability, $\mathbb{P}^{\text{success}}$, is

$$\mathbb{P}^{\text{success}} = \prod_{\forall l(i,j) \text{ on path } P} (1 - \mathcal{P}_{i,j}^{\text{fail}}). \quad (12)$$

Simultaneous multiple paths increases the reliability as given below

$$\mathbb{P}_M^{\text{success}} = 1 - \prod_{m=1}^M \left(1 - \prod_{\forall l(i,j) \text{ on path } P_m} (1 - \mathcal{P}_{i,j}^{\text{fail}}) \right). \quad (13)$$

Figure 9 shows the reliability for variable number of paths with respect to variable link failure probability. It is seen that using simultaneous multiple paths provides considerable higher reliability with respect to single path.

The bandwidth in WSN and MANET is also limited. Usage of a single path for routing may not provide the

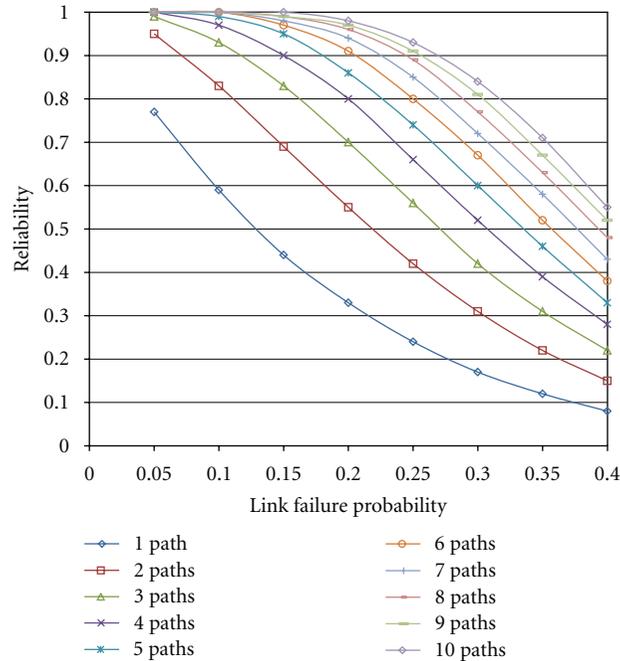


FIGURE 9: End-to-end reliability for variable number multiple paths with 5 hops.

required bandwidth for the current communication. In case of transmission of packets over multiple paths, the aggregate bandwidth of multiple paths may satisfy the bandwidth requirement of the current communication [38]. Therefore, a smaller end-to-end delay may be achieved [38].

6. Simulation and Results

In order to evaluate the performance of the SWR, it is compared with well-known benchmarking protocols. Performance results present efficiency of the proposed approach with respect to other protocols. A number of energy-related metrics are evaluated in detail as well as the impact of the node density. Moreover, there are some additional results to show the characteristics of the SWR.

6.1. Compared Protocols. The proposed approach is compared with known benchmark protocols. One of the benchmark of all geographical routing protocols is the Greedy Perimeter Stateless Routing (GPSR) protocol, which is also a stateless geographical routing protocol [23]. However, it uses neighborhood topology information for packet forwarding in greedy manner. GPSR collects the local topology (neighborhood) information by periodic beaconing messages. We used three beaconing periods for GPSR as 1 sec, 3 sec, and 6 sec. Shorter beaconing intervals increase the energy consumption in an effort to provide more accurate and up to date topology information. Different beacon periods yield different performance results even though they use the same routing approach. Therefore, to differentiate the results of GPSR for different beacon periods, naming strategy that includes the beaconing period such as “GPSR 1 sec” is used. The most well known routing algorithm is flooding. Actually

most of the routing protocols for WSN and ad hoc networks are the variants of flooding with some modifications and optimizations. Flooding is the simplest stateless routing protocol since it does not require any routing table. The original data packet traverses on every path in the network including the shortest one. We also compare the results with a real-time protocol called SPEED [24]. Another well-known energy efficient-stateless protocol, IGF [26] is compared with SWR to compare energy efficiency and other performance metrics. The results are also compared with an imaginary routing protocol, which is called *Virtual Optimal Routing* protocol. It is assumed in this protocol that all nodes always know the optimal path to the destination. Thus, it has not any routing overhead and the data packets are carried over optimal path towards the destination. Therefore, the transmissions and the energy consumption will remain minimal. Such a naturally optimal protocol provides a good mean to compare other protocols for various performance metrics such as energy consumption, lifetime, and remaining energy.

Protocols are experimented with two different scenarios as presented in Table 1. Parameters for the scenarios are subject to observe the performance of the system in different conditions. Nodes are randomly distributed in a well-defined topology [39]. Network is designed with the methodology defined in [40]. Randomly generated, UDP-based constant bit rate (CBR) traffic is used for evaluations. Nodes randomly generate 128 Byte payload packets with a probability of 0.05 packet/min. Packet generation frequency is increased to 1 packet/sec to observe the effects of load to the energy consumption. To provide the double range property [39], nodes have a sensing range (R_S) of 50 meters and a transmission range (R_C) of 100 meters ($R_C/R_S = 2$).

TABLE 1: Scenarios used in simulations.

Scenario number	Number of nodes	Area (m ²)	Density	
			Nodes per m ²	Nodes per node coverage
Sc. 1	100	500 × 500	0.04%	13
Sc. no. 2-A	500	1000 × 1000	0.05%	16
Sc. no. 2-B	1000	1000 × 1000	0.1%	31
Sc. no. 2-C	10000	1000 × 1000	1%	314

6.2. *Performance Evaluation*. Different factors are considered to observe the effects on the performance of the system. In performance evaluations, in addition to the canonical metrics such as energy consumption in routing and effects on network lifetime, routing overhead, some additional metrics such as load of components are observed.

Two energy-related performance metrics in WSNs are the energy consumption and the lifetime of the system. Although some nodes might consume major portion of their energy, the system may continue to live by load balancing and selecting more powered (with respect to energy residual) nodes as retransmitting nodes. Therefore, the lifetime of the system may be prolonged by avoiding the transmissions through energy-limited nodes. If the energy level of the nodes is not considered, the system may fail due sudden energy depletion at some nodes even though most of the nodes may still keep major portion of their energies. Therefore, energy consumption and lifetime have to be investigated to understand the behavior of the system. Note that besides the energy level of the system, the remaining energy of individual nodes is also investigated.

This information gives clues on node redeployment strategies. Thus, the performance evaluations have been performed. To make it clear, the following definitions are made.

- (i) *System Energy*: the ratio of the cumulative energy of the nodes in the system at startup.
- (ii) *System Lifetime*: it is measured from startup until first failure occurrence on path construction from source to the sink.
- (iii) *Remaining Energy Level*: the ratio of the energy at an instant time to the energy at startup.

6.2.1. *Energy Consumption and System Lifetime*. As discussed in Section 5, nodes consume as much power in the receptions as in the transmission. In our experiments, we measure energy consumptions in both transmissions and receptions. Energy consumptions in sensing and computation are considered as negligible constant values as in [34].

Figure 10 shows the remaining system energy percentages against the applied routing algorithms in Scenario 1. The x -axis shows the elapsed simulation time in seconds. The y -axis shows the remaining energy levels of the system. Only the energy consumption on data packet transmission and routing processes are considered (including all transmissions and receptions).

System energies are measured until the lifetime of the system. The lifetime is considered as the first failure on finding

any route to the destination. GPSR with 1 sec beaconing, SPEED, and the flooding algorithm deplete the allocated system energy very quickly. Their lifetimes are very close: 124 seconds, 125 seconds, and 153 seconds for SPEED, GPSR (1 sec beaconing), and flooding, respectively, (Table 2). SPEED and GPSR deplete most of their energy for beaconing, while the flooding depletes its energy on routing process. The system energy of the GPSR protocol is slightly higher than flooding. However, flooding has longer lifetime because it uses all paths at once to reach the destination. Node terminations do not affect flooding if there is a path to the sink. In GPSR, energy consumption decreases and lifetime extends as the beaconing periods increase (e.g., GPSR 3 sec and GPSR 6 sec). However, they fail to live longer than beaconless protocols. Extending the beaconing period in GPSR avoids energy consumption at nodes; however, they fail to keep routing tables up to date. The SWR protocol survives when the simulation ends after 900 sec. Energy consumption in the SWR is close to the energy consumption in the optimal routing. In SWR, nodes consume their energy only in data packet transmissions. However, nodes in IGF consume their energy both on probe and data packet transmissions. Remaining system energy is lower than SWR. IGF fails to find routes at simulation time 531 though it has high remaining system energy. It fails to find route due to node terminations close to the sink node because of double transmissions of nodes on-the-route (one for probe reply and one for data packet).

In addition to the lifetime of the protocols, information about node terminations is presented in Table 2. Each protocol would fail to find routes after node terminations. However, their response to node terminations varies, and that affects their lifetime. In GPSR 1 sec, the paths are constructed until second 124. When the number of terminated nodes reaches 23, GPSR 1 sec fails to find routes to the sink. SPEED presents similar results with GPSR 1 sec. Although there are more terminated nodes in flooding (34 terminated nodes) than both SPEED and GPSR 1 sec, paths are constructed until second 153 since every possible path is tried in flooding even if there are many terminated nodes. Besides that in GPSR and SPEED, nodes closer to the sink deplete energy more quickly because roughly all paths toward the sink involve these nodes. Such a quick depletion thus composes a gap surrounding the sink. On the other hand, in flooding, nodes deplete their energy almost equally, because every node equally involves in routing. These results are presented in Figure 11.

It is better to analyze Figure 11 and Table 2 together. It is seen in Figure 11(a) that when the GPSR 1 sec fails to find any route at 125th second, the other nodes almost have depleted

TABLE 2: Comparisons of the protocols with respect to lifetime and node terminations.

Routing protocols	Flooding	GPSR 1 sec	GPSR 3 sec	GPSR 6 sec	SPEED	IGF	SWR
Average system lifetime	153 sec	125 sec	331 sec	571 sec	124 sec	531 sec	>900 sec
Time of the first node termination	109 sec	106 sec	268 sec	481 sec	104 sec	498 sec	NONE in 900 sec
Average number of terminated nodes on destination unreachable	41	23	24	15	29	8	NONE in 900 sec

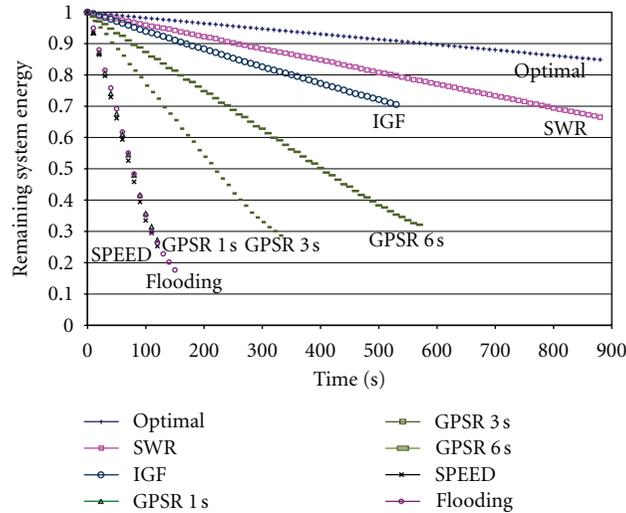


FIGURE 10: Remaining system energy levels of the protocols in Scenario 1.

their energies. Energy consumption has been diffused all over the system. If the system lifetime has been prolonged a few more seconds, almost all of the nodes would terminate. Although the remaining nodes in flooding have higher energy levels than the remaining nodes in GPSR 1 sec (Figure 11(a)) at this moment, similar results to GPSR 1 sec are observed in flooding. Due to flooding, all nodes participate equally to the routing process. This makes the nodes have almost equal energy levels. Similar results are observed in GPSR 3 sec and GPSR 6 sec except the extended lifetime due to beacon period extension as seen in Figures 11(c)–11(e). In SWR, nodes have higher energy levels and longer lifetime. In IGF, due to probing on data packet transmissions, nodes deplete their energies earlier than SWR. In SWR energy levels of the nodes are close to the optimal. The reason is that energy is consumed only in data packet transmissions in these protocols. Node terminations do not occur in these two protocols in the simulation course of 900 s.

6.2.2. Effects of Node Density. Routing protocols should perform well on both densely and sparsely deployed environments, however, fall beyond this expectation. They suffer from the node density due to introduced overhead and more energy consumptions at nodes. Scenario 2 is used to compare protocols in dense environments. Number of the nodes per

unit area is increased to observe the effects of node density. Node density ratios are 0.05%, 0.1%, and 1% to total area; in terms of numbers of nodes 500, 1000, and 10000, respectively. In this scenario, to be able to observe and compare the effects of node density, unlimited energy is given to each node. Other parameters are same as in Scenario 1. Results of energy consumption with different node densities are presented in Figures 12(a)–12(c) for node densities 0.05%, 0.1%, and 1%, respectively.

The x -axis shows the elapsed simulation time in seconds. The y -axis shows the system-wide energy consumption in joules. Only the energy consumption related with the routing processes (transmissions and receptions) are considered. Note that the y -axis is in logarithmic scale. It is clear that as the node density increases, energy consumption increases (Figures 12(a)–12(c)). Similar results are observed for all protocols. GPSR 1 sec and SPEED present almost identical, so they overlaps in Figures 12(a)–12(c). The difference due to beaconing in GPSR 1 sec, GPSR 3 sec, and GPSR 6 sec is seen clearly. Unsurprisingly, there is gap between the beacon-based protocols and beaconless protocols. The difference between these protocols considering the energy consumption in the logarithmic scale emphasizes the excessive energy consumption in beacon-based protocols and the energy-efficiency of beaconless protocols. Beaconless protocols outperform the others. Though the SWR uses multiple paths, it has

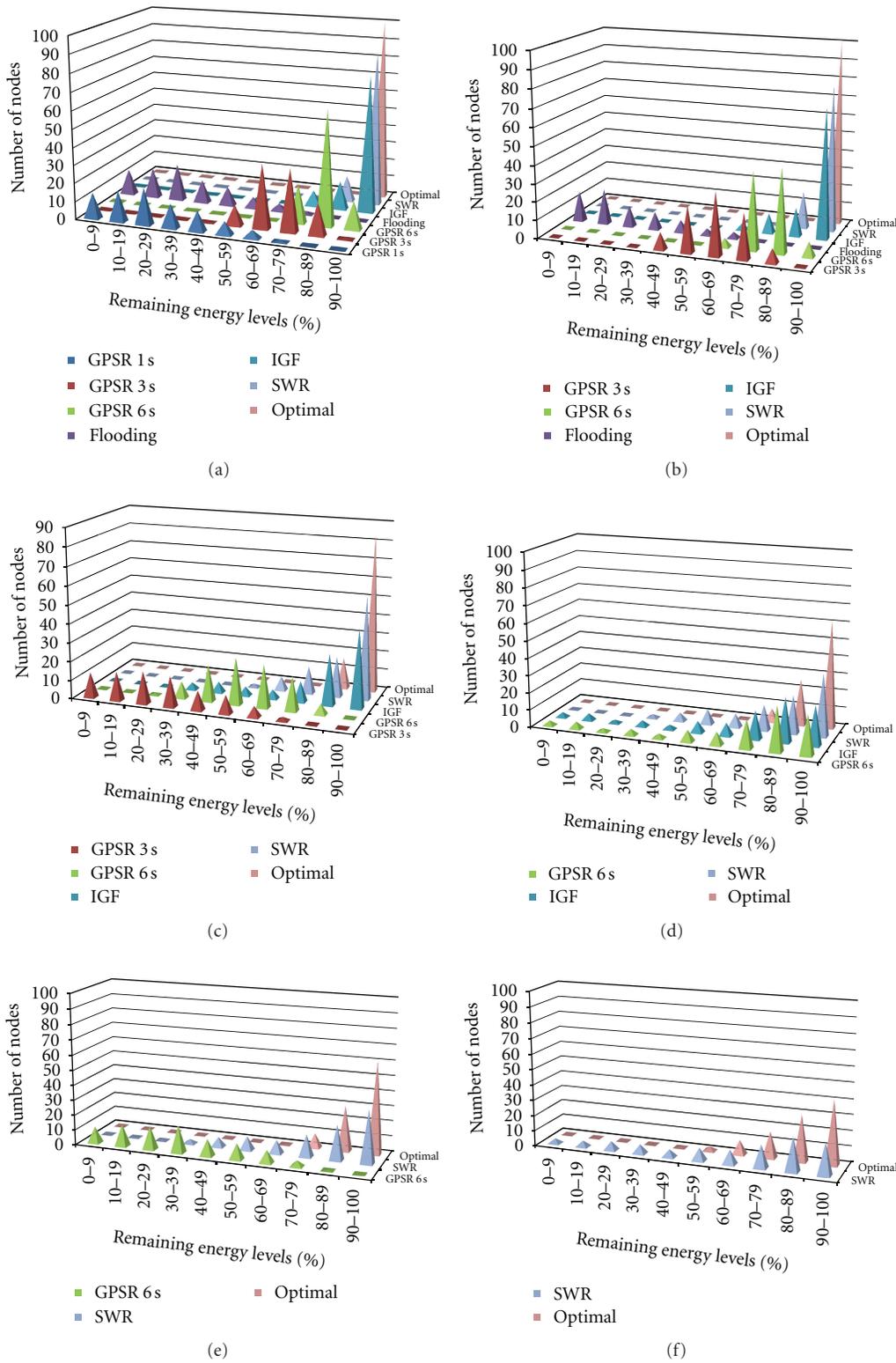


FIGURE 11: Remaining energy levels of the nodes when (a) “SPEED” fails to find a route at time 124 sec, (b) “GPSR 1 sec” fails to find a route at time 125 sec, (c) “Flooding” fails to find a route at time 153 sec, (d) “GPSR 3 sec” fails to find a route at time 331 sec, (e) “IGF” fails to find a route at time 531 sec, and (f) “GPSR 6 sec” fails to find a route at time 571 sec.

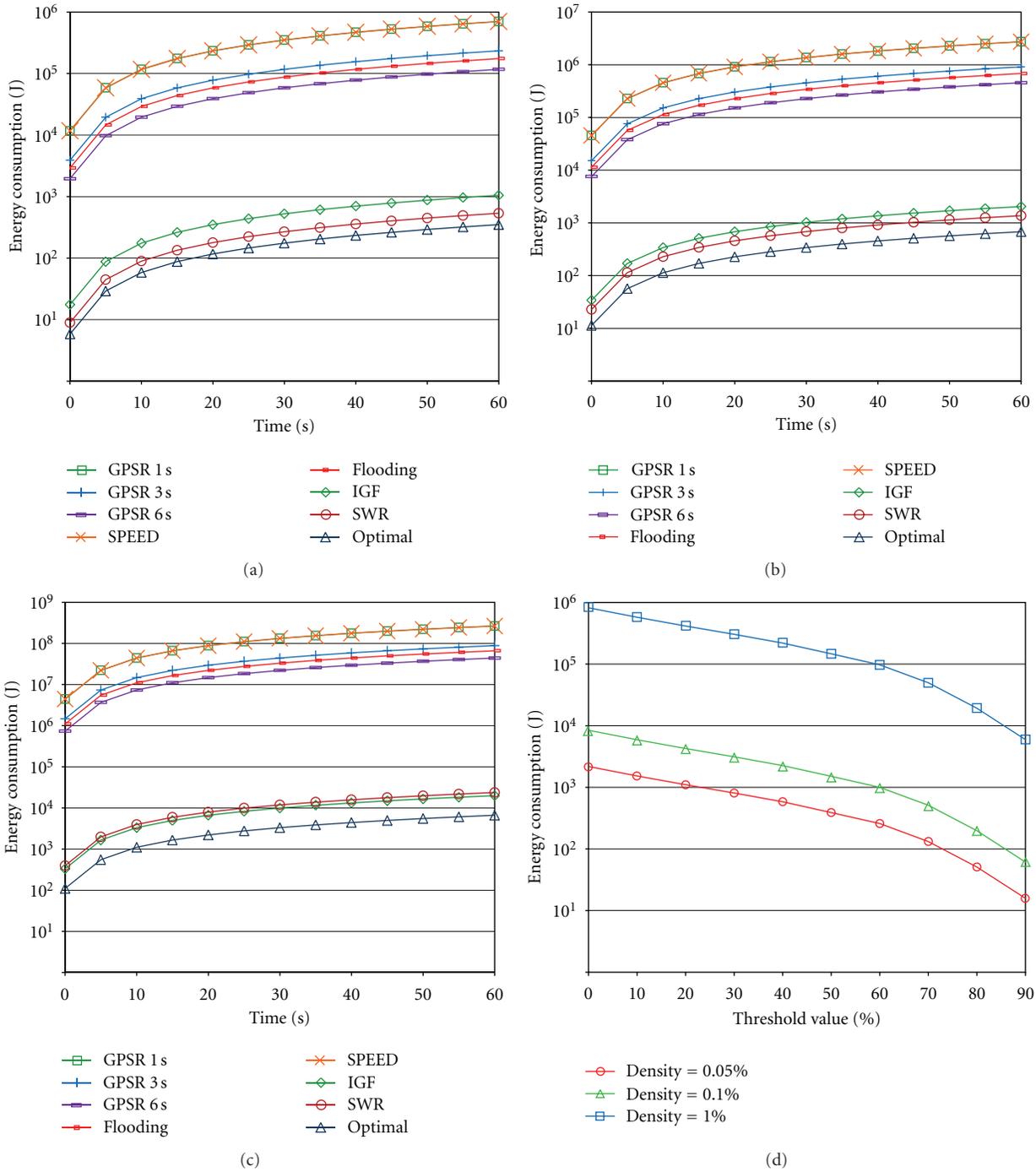


FIGURE 12: Energy consumption of protocols with different node densities: (a) 0.05% nodes/m², (b) 0.1% nodes/m², (c) 1% nodes/m², and (d) effects of the threshold value and the node densities to the energy consumption in SWR.

low energy consumption even in high density networks. Effects of node density to SWR protocol are not more than other protocols. In fact, the effect remains at a moderate level. On the other hand, the amount of consumed energy in SWR can be decreased by adjusting the threshold value (Figure 12(d)). Increasing the threshold value reduces the number of retransmissions, as described in Section 4. In this respect, SWR presents a great opportunity and flexibility

with respect to other protocols to adjust the threshold value according to network density and environmental parameters.

Figure 12(d) shows the energy consumption with different threshold values for SWR. The x-axis shows threshold values applied in routing algorithm for SWR. Note that the y-axis shows the system-wide energy consumption in logarithmic scale. The amount of energy saving is very high when a higher threshold value is used. For example, for the density

0.05%, the amount of energy consumed with threshold value 0.5 is 388 joules, and with threshold value 0.9 is 16. There is a 95% energy saving with these values. When the density of the network increases, it becomes more noticeable. For the density 0.1%, the amount of energy consumed with threshold value 0.5 is 1509 joules, and with threshold value 0.9 is 61 joules. For the density 1%, the amount of energy consumed with threshold value 0.5 is 146706 joules, and with threshold value 0.9 is 5956 joules. The amount of energy saving is 96% with these values.

6.2.3. Effects of Range and Threshold to Energy Consumption.

In SWR, energy consumption can be reduced by adjusting the threshold value as shown in Figure 12(d). As defined in Section 4.3, threshold value can be adjusted momentarily and independently in a distributed manner at each node for different reasons according to the current needs of the node. It depends on the current condition or the event a node experiencing, such as void recovery, requirement of higher reliability or guaranteed delivery, and urgent or real-time data transmissions. After the completion of the event, the threshold value can be readapted to the default value to save energy. These adaptations occur in a distributed manner independently at nodes which require these adaptations and without any administrative manipulation outside the network. Each node itself intrinsically decides to increase or decrease the threshold according to its current conditions. On the other hand, to satisfy some performance metrics, for example, reliability, the threshold value can be set to a new default desired value, which is based on the preknown information about the network-wide requirements. It can be reset again to the predefined default threshold value as needs according to the network-wide requirements as described above.

Another parameter which affects the energy consumption is the range of the transmissions. Increase in transmission range causes more nodes to receive transmissions, which cumulatively increases the system-wide energy consumption. On the other hand, range increase causes shorter path constructions, which causes a reduction on the number of transmissions and receptions. Therefore, the effect of range to the energy consumption is examined. Threshold value in SWR affects the reliability and energy consumption. Thus, effects of these two parameters range and threshold are examined. In WSNs, there are some approaches that adjust the transmission range of the transmitter according to the known distance of the receiver. In SWR, nodes do not have any information about the topology nor neighborhood nodes. Usage of adaptive transmitter is needless in SWR. Transmission range is fixed. What a node may need to change is only the threshold value in SWR.

Effects of range and threshold values are shown in Figures 13(a) and 13(b). Figure 13(a) shows the relation between relay nodes coverage, range, and threshold value. The x -axis shows the *applied threshold value*, and the y -axis shows the *relay node coverage reduction*. Relay node coverage defines the nodes are the candidates to be involved in relaying a packet between a source-destination pair. These relay nodes are located in an area shaped similar to the one in Figure 2(h).

Of the nodes in the transmission range, SWR protocol allows only those nodes which have lower weight values to relay the data. The number of these nodes is dependent on the applied threshold value. Therefore, change in threshold value changes the covered area, in other words, changes the number of relay nodes.

Figure 13(a) shows the relay node coverage relationship for a source-destination pair 100 meters away from each other. For 90 meters transmission range (tx range = 90), data is relayed in 2 hops. With a threshold value of 10%, the covered area is reduced 72% with respect to the area covered with threshold value 0%. For transmission range of 80 meters (tx range = 80), data is again relayed in 2 hops. With a threshold value 10%, the covered area is reduced 33% with respect to the area covered with threshold value 0%. Other transmission ranges (tx range = 70, 60, 50, 40, 30, 20, 10) show similar results with respect to the transmission range of 80 meters (tx range = 80) for threshold value 10%. Secondly, transmission ranges between 70 meters and 10 meters present close reduction values for the same threshold values. However, transmission ranges of 90 meters and 80 meters present a better reduction in area coverage. The reason is that with high transmission ranges in close distances between the source and the destination, some unnecessary part of the topology is covered. In other words, the data is relayed to some far way nodes from the sinks. Applying a threshold value prevents the far way nodes from the sink to be a relay node. When a smaller transmission range is used, the distance between the source and the destination is divided more equally. This is similar to occupy a square shape area with smaller square shape areas.

It is clear that increasing the threshold value reduces the covered area by relay nodes. It should be pointed out that there is a great coverage area reduction (between 77–93%) even with a 50% threshold value. Increasing the threshold value higher than 50% causes less reduction in coverage area. In simulations, it is found out that 50% threshold value provides a high reliability. Therefore, 50% threshold value is selected as default parameter.

The inference related with range and threshold value described above is seen clearly in Figure 13(b). In this figure, effects of range and threshold value are shown together. These are the results for 100 meters distance between the source and the destination. As seen, as the threshold value increases, the energy gain increases. However, energy gain gets higher as the transmission range increases. The reason of the erratic part for ranges 90 and 80 is as described above.

6.2.4. Energy Consumption per Data Delivery. We also measure the energy consumption on transportation of one data packet from source to destination node. Multiple receptions at the destination for the same data packet are counted as one successful delivery. Table 3 shows the comparative energy consumptions for different path lengths for Scenario 1. In flooding, equal amount of energy is consumed for each path length, since the packet is flooded within the network each time. However, for all other protocols, the number of transmissions and receptions varies according to the path length.

TABLE 3: Comparisons of the protocols with respect to energy consumption per data delivery.

Routing protocols	Energy consumption (joule)					
	1 hop	2 hop	3 hop	4 hop	Arithmetic average	Average by simulation
Flooding	1545.6	1545.6	1545.6	1545.6	1545.60	1043.01
GPSR 1 sec	1563.8	1582	1600.2	1618.4	1591.10	1306.99
GPSR 3 sec	533.4	551.6	569.8	588	560.70	519.71
GPSR 6 sec	275.8	294	312.2	330.4	303.10	259.70
SPEED	1582	1618.4	1654.8	1691.2	1636.60	1341.45
IGF	54.6	109.2	163.8	218.4	136.50	121.90
SWR	18.2	63.7	109.2	163.8	88.73	104.40
Optimal	18.2	36.4	54.6	72.8	45.50	36.52

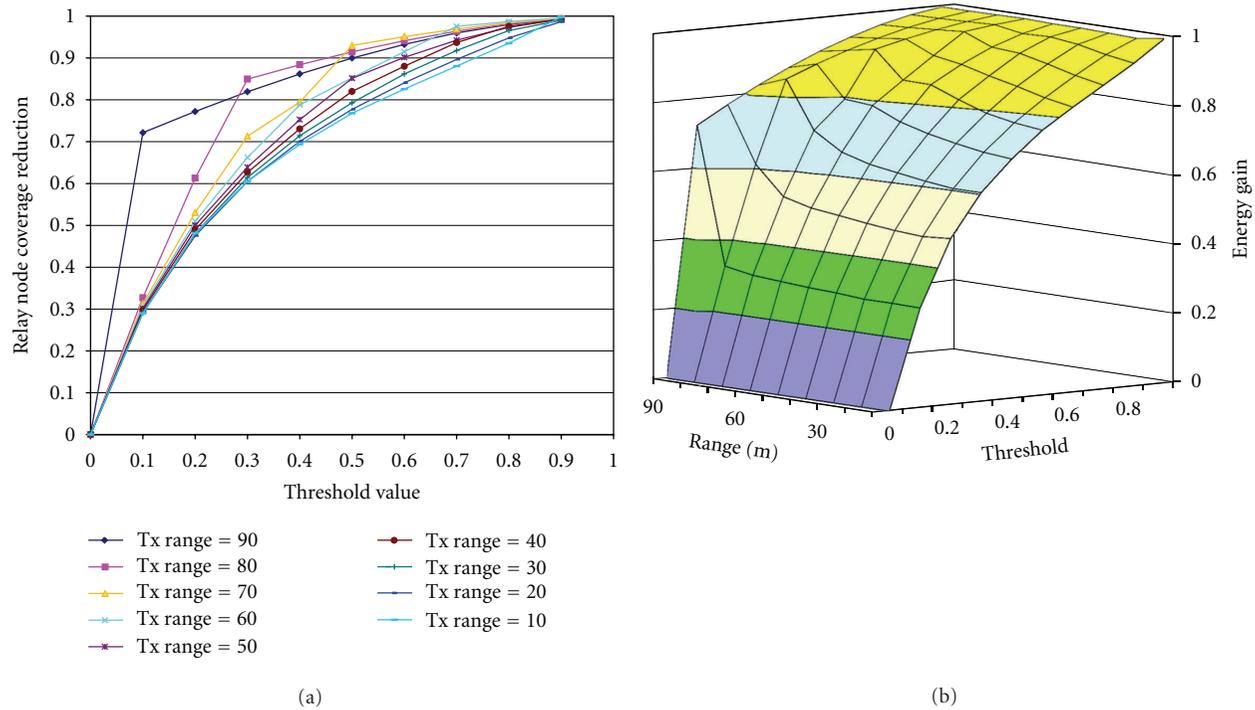


FIGURE 13: (a) Effects of range and threshold value on the relay nodes coverage in SWR. (b) Effects of threshold and range values on the energy consumption in SWR.

GPSR and SPEED use periodic beaconing to construct and update the neighborhood topology. They need this process to relay the data packets. Therefore, energy consumption for beaconing is considered and added to the energy consumption in GPSR and SPEED protocols. As expected, GPSR, SPEED, and flooding have higher energy consumptions. In GPSR and SPEED, beaconing consumes excessive energy. As the beaconing period extends, energy consumption reduces. For these protocols, packet forwarding consumes much less energy with respect to other protocols because shortest paths are constructed in GPSR and SPEED due to priori known topology. Beaconing generates the major energy consumption in these protocols.

Beaconless protocols consume less energy for routing. Compared to the energy consumption of these protocols, the order from high to low is IGF and SWR. Energy consumption in IGF for one data packet is higher than SWR because of

its probe packets before sending data packets. Energy consumption in SWR is higher than the Optimal Routing, but data is sent over multiple paths in SWR providing reliability. SWR provides reliability and minimum delay with shortest path by using multiple transmissions and utilizing multiple paths, while there is only one path prone to the failures in all other protocols except flooding. Energy consumption in SWR is very low and better than all protocols except the Optimal Routing.

Average during Simulation values are the averages of the results obtained in Scenario 1. There is difference between the *Average during Simulation* and *Arithmetic Average*. There are two reasons for this difference. Arithmetic Average is the average of energy consumptions in different path lengths. But the number of constructed paths for different path lengths cannot be the same during the simulations. This affects the average energy per path. Secondly, protocols continue to find

TABLE 4: Comparisons of the protocols with respect to transmissions and receptions per data delivery.

Routing protocols	Number of transmissions and receptions			
	1 hop	2 hops	3 hops	4 hops
Flooding	100 tx/1004 rx	100 tx/1004 rx	100 tx/1004 rx	100 tx/1004 rx
GPSR 1 sec	101 tx/1016 rx	102 tx/1028 rx	103 tx/1040 rx	104 tx/1052 rx
GPSR 3 sec	34 tx/347 rx	35 tx/359 rx	36 tx/371 rx	37 tx/383 rx
GPSR 6 sec	18 tx/179 rx	19 tx/191 rx	20 tx/203 rx	21 tx/215 rx
SPEED	102 tx/1028 rx	104 tx/1052 rx	106 tx/1076 rx	108 tx/1100 rx
IGF	3 tx/36 rx	6 tx/72 rx	9 tx/108 rx	12 tx/144 rx
SWR	1 tx/12 rx	3 tx/42 rx	6 tx/72 rx	9 tx/108 rx
Optimal	1 tx/12 rx	2 tx/24 rx	3 tx/36 rx	4 tx/48 rx

routes even if there are some terminated nodes. However, their responses to node terminations vary in terms of lifetime and ability to find routes. For example, flooding has the ability to find routes even if there are many node terminations. In Scenario 1, as shown previously in Table 2, flooding continues to find routes even when there are many node terminations. Supporting results are observed in Table 4. Flooding always makes the same amount of transmissions and receptions if there are not any node terminations. However, node terminations during the simulation reduce the number of transmissions and receptions in flooding. Similar results are observed for beacon-based protocols, GPSR and SPEED. Terminated nodes reduce the system-wide energy consumption in beaconing. Depending on the beaconing period, results change for GPSR 3 sec and GPSR 6 sec.

Scalability can be considered through multiple perspectives. It can be evaluated with regard to an increase in the sensor network area, the density of the nodes, the traffic, and so forth. The impact of all issues is minimized in the virtual optimal algorithm. Simulations indicate that the SWR is not only a better performing algorithm compared to the others in the literature but also quite close to the virtual optimal one. Main reason is that the SWR consumes the energy only on transmissions where and when events occur. Periodic data packet transmissions can also cause performance degradation as observed in other protocols. Considering the event-based data packet transmissions, the scalability of the SWR is very close to *virtual optimal routing* protocol as shown in Figures 10 and 11 and Tables 3 and 4.

The SWR is essential and imperative for Mission-Critical applications which require high reliability. The SWR is also suitable for event-based data acquisition networks rather than periodic data acquisition networks.

7. Conclusions

In this paper, a novel stateless routing algorithm for WSN, the SWR, is proposed. The SWR differs from other proposed protocols in the literature in many ways. It is a completely stateless routing protocol that does not require any topology knowledge on data transmissions. As it provides reliability by conveying data over multiple paths, the SWR reduces

transmissions and energy consumption drastically by avoiding transmissions on topology learning and control information exchange. Implicit features of the SWR provide nodes to adapt current conditions and to recover from voids. Moreover, an explicit void recovery is proposed to recover from voids in case of a node cannot recover a void by the implicit approach. Each node makes its own decision in a smart way according to its own conditions. In the SWR, data packets spontaneously flow over simultaneous multiple paths. This approach provides utilization of the shortest available path. Stateless property of the SWR helps to reduce the delay at nodes. These features are provided with a simple algorithm that does not require much resource such as CPU and memory. Assuring the use of cheap and disposable nodes for WSNs applications, the SWR can be used in instantly topology changing networks as well as with stationary ones. These features make the SWR a unique one considering the current literature. The measurement of delay is considered as a future work.

References

- [1] S. Giordano, I. Stojmenovic, and L. Blazevic, "Position based routing algorithms for Ad Hoc networks: a taxonomy," in *Ad Hoc Wireless Networking*, X. Cheng, X. Huang, and D. Z. Du, Eds., pp. 103–136, Kluwer Academic, Boston, Mass, USA, 2004.
- [2] M. Heissenbüttel, T. Braun, M. Wälchli, and T. Bernoulli, "Optimized stateless broadcasting in wireless multi-hop networks," in *Proceedings of 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, pp. 1–12, April 2006.
- [3] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile Ad Hoc networks," *IEEE Network*, vol. 15, no. 6, pp. 30–39, 2001.
- [4] F. Araujo and L. Rodrigues, "Survey on position-based routing," Tech. Rep. TR-01, University of Lisbon, 2006.
- [5] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.
- [6] M. Soyuturk and T. Altılar, "The challenges and the approaches for the geographic routing protocols in wireless sensor networks," in *Proceedings of the IEEE International Conference on Technologies for Homeland Security and Safety (IEEE TEHOSS '06)*, October 2006.

- [7] S. Dulman, T. Nieberg, J. Wu, and P. Havinga, "Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking (WCNC '03)*, pp. 1918–1922, March 2003.
- [8] U. Monaco, F. Cuomo, T. Melodia, F. Ricciato, and M. Borghini, "Understanding optimal data gathering in the energy and latency domains of a wireless sensor network," *Computer Networks*, vol. 50, no. 18, pp. 3564–3584, 2006.
- [9] H. Fuessler, J. Widmer, M. Kasemann, and M. Mauve, "Beaconless position-based routing for mobile Ad-Hoc networks," Tech. Rep., Department of Computer Science, University of Mannheim, 2003.
- [10] M. Heissenbuttel and T. Braun, "A novel position-based and beacon-less routing algorithm for mobile Ad Hoc networks," in *Proceedings of 3rd Workshop on Applications and Services in Wireless Networks (ASWN '03)*, pp. 197–210, Berlin, Germany, 2003.
- [11] M. Chawla, N. Goel, K. Kalaichelvan, A. Nayak, and I. Stojmenovic, "Beaconless position based routing with guaranteed delivery for wireless ad-hoc and sensor Networks," *IFIP International Federation for Information Processing*, vol. 212, pp. 61–70, 2006.
- [12] H. Xiaoyan, X. Kaixin, and M. Gerla, "Scalable routing protocols for mobile Ad Hoc networks," *IEEE Network*, vol. 16, no. 4, pp. 11–21, 2002.
- [13] Y. B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile Ad Hoc networks," *Wireless Networks*, vol. 6, no. 4, pp. 307–321, 2000.
- [14] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A Distance routing effect algorithm for mobility (DREAM)," in *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking of (MOBICOM '98)*, pp. 76–84, 1998.
- [15] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing in geometric graphs," in *Proceedings of 11th Canadian Conference on Computational Geometry (CCCG '99)*, pp. 51–54, 1999.
- [16] T. Melodia, D. Pompili, and I. F. Akyildiz, "On the interdependence of distributed topology control and geographical routing in Ad Hoc and sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 520–531, 2005.
- [17] L. Savidge, H. Lee, H. Aghajan, and A. Goldsmith, "Event-driven geographic routing for wireless image sensor networks," in *Proceedings of the Proceedings of Cognitive Systems and Interactive Sensors (COGIS '06)*, March 2006.
- [18] S. Lee, B. Bhattacharjee, and S. Banerjee, "Efficient geographic routing in multihop wireless networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '05)*, pp. 230–241, May 2005.
- [19] N. Ahmed, S. S. Kanhere, and S. Jha, "The holes problem in wireless sensor networks: a survey," in *ACM SIGMOBILE Mobile Computing and Communications Review (MCR '09)*, pp. 4–18, April 2005.
- [20] Q. Fang, J. Gao, and L. J. Guibas, "Locating and bypassing routing holes in sensor networks," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 2458–2468, March 2004.
- [21] S. Chen, G. Fan, and J.-H. Cui, "Avoid "Void" in geographic routing for data aggregation in sensor networks," *International Journal of Ad Hoc and Ubiquitous Computing Archive*, vol. 1, no. 4, pp. 169–178, 2006.
- [22] D. Son, A. Helmy, and B. Krishnamachari, "The effect of mobility-induced location errors on geographic routing in mobile Ad Hoc and sensor networks: analysis and improvement using mobility prediction," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 233–245, 2004.
- [23] B. Karp and H.T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00)*, pp. 243–254, Boston, Mass, USA, August 2000.
- [24] H. Tian, J. A. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: a stateless protocol for real-time communication in sensor networks," in *Proceedings of the 23rd International Conference on Distributed Computing Systems*, pp. 46–55, May 2003.
- [25] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, "Contention-based forwarding for mobile Ad Hoc networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, 2003.
- [26] B. Blum, T. He, S. Son, and J. Stankovic, "IGF: a state-free robust communication protocol for wireless sensor networks," Tech. Rep., University of Virginia, Charlottesville, Va, USA, 2003.
- [27] M. Soyuturk and D. T. Altılar, "Stateless data flow approach with void avoidance for wireless Ad Hoc and sensor networks," in *Proceedings of the 2nd International Symposium on Wireless Pervasive Computing (ISWPC '07)*, pp. 252–257, February 2007.
- [28] J. B. Schmitt, F. A. Zdarsky, and U. Roedig, "Sensor network calculus with multiple sinks," in *Proceedings of the Workshop on Performance Control in Wireless Sensor Networks (IFIP NETWORKING '06)*, Lecture Notes in Computer Science, pp. 6–13, Springer, Albacete, Spain, September 2006.
- [29] K. Yuen, B. Liang, and B. Li, "A distributed framework for correlated data gathering in sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 578–593, 2008.
- [30] N. Antunes, G. Jacinto, and A. Pacheco, "An analytical framework to infer multihop path reliability in MANETs," in *Proceedings of the International Conference on Measurement and Modeling of Computer Systems (ACM SIGMETRICS '10)*, pp. 323–332, 2010.
- [31] T. Korkmaz and K. Sarac, "Characterizing link and path reliability in large-scale wireless sensor networks," in *Proceedings of the 6th Annual IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '10)*, pp. 217–224, October 2010.
- [32] A. Tsigridis and Z. J. Haas, "Analysis of multipath routing—part I: the effect on the packet delivery ratio," *IEEE Transactions on Wireless Communications*, vol. 3, no. 1, pp. 138–146, 2004.
- [33] W. Yang, X. Yang, S. Yang, and D. Yang, "A greedy-based stable multi-path routing protocol in mobile Ad Hoc networks," *Ad Hoc Networks*, vol. 9, pp. 662–674, 2011.
- [34] E. I. Oyman and C. Ersoy, "Overhead energy considerations for efficient routing in wireless sensor networks," *Computer Networks*, vol. 46, no. 4, pp. 465–478, 2004.
- [35] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 364–369, April 2005.
- [36] A. Caracas et al., "Energy-efficiency through micro-managing communication and optimizing sleep," in *Proceedings of 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '11)*, 2011.

- [37] S. Tozlu, "Feasibility of Wi-Fi enabled sensors for internet of things," in *Proceedings of the 7th IEEE International Wireless Communications and Mobile Computing Conference (IWCMC '11)*, 2011.
- [38] S. Mueller, R. P. Tsang, and D. Ghosal, "Multipath routing in mobile Ad Hoc networks: issues and challenges," in *Proceedings of the International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS '03)*, vol. 2965 of *Lecture Notes in Computer Science*, pp. 209–234, 2003.
- [39] C. Avin, "Fast and efficient restricted delaunay triangulation in random geometric graphs," in *Proceedings of the Workshop on Combinatorial and Alg. Aspects of Networking (CAAN '05)*, 2005.
- [40] Y. Yu, B. Hong, and V. K. Prasanna, "On communication models for algorithm design in networked sensor systems: a case study," *Pervasive and Mobile Computing*, vol. 1, no. 1, pp. 95–121, 2005.

Research Article

An Experimental Study of WSN Power Efficiency: MICAz Networks with XMesh

Tyler W. Davis,¹ Xu Liang,¹ Miguel Navarro,² Diviyansh Bhatnagar,² and Yao Liang²

¹Department of Civil and Environmental Engineering, University of Pittsburgh, 3700 O'Hara Street, Benedum Hall Room 949, Pittsburgh, PA 15261, USA

²Department of Computer and Information Science, Indiana University Purdue University, 723 West Michigan Street, SL 280, Indianapolis, IN 46202, USA

Correspondence should be addressed to Xu Liang, xuliang@pitt.edu and Yao Liang, yliang@cs.iupui.edu

Received 16 July 2011; Revised 13 October 2011; Accepted 20 October 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Tyler W. Davis et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This is an investigation of Wireless Sensor Networks (WSNs) using Memsic's XMesh routing protocol on MICAz wireless motes. It focuses on the study of the practical aspects of WSNs' power efficiency and network characteristics, which play a critical role in real-world WSN deployments for environmental monitoring. Based on an experimental study and following a quantitative approach, this work examines XMesh's high power and low power operation modes and the data transmission intervals, among other factors. Route utilization was identified as a major contributor of the mote's battery use. A field study was conducted as a point of comparison and the results obtained were comparable to the laboratory tests with regards to the battery life and the mote's route utilization. The network reliability was found to be considerably lower in the field study. In addition, it was found that the original WSN gateway, used during the study, presented severe practical limitations regarding the system's robustness and reliability. To address these problems, we present a solution based on our integrated network and data management system, which successfully facilitates the deployment of a new WSN gateway and significantly improves the operational robustness and reliability of the WSN system.

1. Introduction

Wireless sensor networks (WSNs) have demonstrated their potential and promising application in different fields of science and engineering [1, 2], such as geophysical studies for volcanic activities [3], environmental monitoring in glacier regions [4], structural monitoring [5], and healthcare applications [6]. However, the severe resource constraints of wireless sensor nodes (e.g., battery power, memory size, processor capacity, and network bandwidth) in WSNs raise new theoretical and practical challenges, drawing great attention in the research community. This work focuses on the practical aspects of WSN power efficiency, which is critical in real-world WSN deployments for environmental monitoring. It is essential that domain scientists and engineers who foresee the potential benefits of including WSNs in their studies and experiments have a fundamental and comprehensive understanding about the WSN power

efficiency and characteristics for different applications under dynamic operational environments.

Multiple research works have attempted to identify key features in WSN deployments [7–9]. While good qualitative results and guidelines are available, the lack of more quantitative results and descriptions is often a major obstacle in today's WSN design, implementations and deployments. In practice, WSN deployments can lead to various important tradeoffs among quality of service, network performance, power consumption, and operational cost. Understanding these tradeoffs in a quantitative way is fundamental for any successful and smart WSN practice. This experimental study *quantitatively* investigates the power efficiency and battery savings for WSNs with various network characteristics, using the popular MICAz wireless motes for environmental monitoring.

WSNs often use individual node power efficiency as a key performance metric, because the battery power of individual

nodes can lead to multiple failure types within the network. For example, [7] examined fourteen environmental WSN field deployments from year 2002 to 2008, ranging in scale from 3 to 98 nodes. Their examination showed that battery power can affect every level of wireless networking problems which they had classified into four categories: node, link, path, and global failures. Node failures can occur when the battery power drops below the level at which the mote can still operate. Link failures can occur when low battery power reduces the range that a mote can transmit its data, effectively removing it from the network's communication topology. Path failures can result when an important node or nodes that route transmitting data have node or link failures forcing the network to use less efficient paths for transmitting data to the base station. Global failures can occur when a critical or bottleneck node experiences node or link failure cutting off the transmission of data from the whole or part of the network. It is important to identify these problems and their causes such that they can be avoided in field deployments.

In this study, we adopt the commercially available Memsic's WSN platform as a vehicle to carry out our investigation. It is one of today's most widely used WSN platforms (previously developed by Crossbow Technology), in which motes are programmed in nesC language [10], linked together with specific data acquisition boards, to form a mesh WSN for various applications. The application code, compiled and loaded to wireless motes, runs under the motes' TinyOS operating system [11]. Memsic's XMesh routing protocol [12], which features TrueMesh technology, is utilized as the networking foundation for these applications. It provides a mesh network which is self-healing and self-organizing, where each mote acts as both a sensor node and a router for its neighbor's data. The ad hoc formation of the network is based on link estimates made between node neighbors and routes data down a path of lowest transmission cost to the base station where data is stored. Parameter assignment in the nesC code along with some runtime argument passing during compilation allows users to have partial control over the mote's power efficiency.

Using Memsic's WSN application platform, motes can be programmed in either a high power (HP), low power (LP), or extended low power (ELP) operation mode. Note that the ELP operation mode does not support routing. Since we consider multihop networks, the ELP operation mode is not included in this study. LP and HP operation modes implement different power efficiencies and thus have different battery savings associated with them. In addition to the two operation modes, the transmission frequency of data packets can also be manually adjusted. The transmission rate also plays an important role on the battery life of the motes, because data transmission is the most power consuming operation, as shown in [7, Table 3], [13, Table 2], and [14, Table 6-2]. Adjustments to the operation mode and transmission frequency can reduce the power consumption of the motes which increases their operating life. The power consumption of wireless motes is an important consideration when deploying networks as it may affect the sensor readings [15] and network connections [7].

We investigate the battery savings by implementing the two operation modes over various data collection intervals with Memsic's XMesh WSN. Battery life is measured over the transmission life of the motes and is used to compare the actual battery savings of each configuration. In addition, we share our experiences and lessons learned from our experimental study regarding the WSN gateway Stargate Netbridge's operational robustness. It is found that Stargate Netbridge, a Linksys NSLU2 device specially modified by Crossbow, has a severe robustness and reliability issues for practical WSN deployments. To address this problem, we present how the solution of our general integrated network and data management system can facilitate the new deployment of a WSN gateway to successfully replace the Stargate Netbridge and significantly improve the operational robustness of WSN deployments.

The remainder of this paper is organized as follows: Section 2 includes an in depth description of the laboratory experiments performed and their results. Section 3 examines a prototype testbed of eleven nodes and compares the network and mote operations to the laboratory experiments. Section 4 presents the solution of our integrated network and data management system for the deployment and robust operation of the XMesh-based WSN in order to overcome the original Stargate Netbridge gateway limitations and to facilitate a more complete study in the future. The concluding remarks based on the findings of this study are given in Section 5.

2. Laboratory Methods

To determine the effect of different sampling frequencies on battery life, a series of experiments was conducted using the MICAz wireless mote, equipped with a MDA300 data acquisition board, both manufactured by Memsic Corporation (previously Crossbow Technology). The mote's operation mode, sampling, and transmission intervals were investigated in these experiments. The mote's operation mode is set using the XMesh high power (HP) or low power (LP) configuration. The difference between these two settings is mainly in the bandwidth consumption and latency of the transmissions. The LP mode, which utilizes a sleep function that powers off all unnecessary electronics between operations, has a high latency (i.e., long transmission delay) and low bandwidth consumption (i.e., low data capacity) compared to the HP mode.

The mote's onboard radio power is adjustable within a range of 0 to -25 dBm, the maximum and minimum power allocations, respectively. The MICAz CC2420 radio transceiver [16] uses IEEE 802.15.4 protocol [17] and transmits data in the 2.4 GHz frequency band with a maximum data rate of 250 kbits/s. Other investigations have already looked at the effects of the transmission power on mote connectivity and battery power [9]. The previous study was interested in understanding how the antenna power affected the transmission distance and how implementing various duty cycles increased the battery life. In contrast, this study is more focused on how the data collection rate

and multihop network functionality affects battery life. The default 0 dBm (1 mW) radio power setting in the 2.405–2.425 GHz frequency channel was used in this study. The data message interval (DMI) and the data transmission interval, which are designated to be the same value, can be set to any multiple of seconds. Intervals were chosen from 10 to 900 seconds.

2.1. Experiments and Analysis on Basic Battery Capacity. In an effort to reduce uncertainties in the sampling interval study that may be caused by the variability of the individual motes or batteries, eight random motes were selected and tested four times each (tests A–D) under the HP mode using a 1-second DMI. To test the effect of the battery capacity, half of the motes tested were powered with two 2500 mAh AA batteries and the other half were powered with two 2450 mAh AA batteries. The batteries used in this experiment were all rechargeable nickel-metal hydride (NiMH) AA batteries. In this experiment, motes were placed on desktops in close proximity to the base station. A summary of the results of these tests for the eight motes is given in Table 1.

The starting and ending battery voltage (based on the two AA batteries) for the experiment are shown in Table 1. The starting voltage is based on the earliest stable battery level recorded in the mote's data packets. This is typically within the first 5–10 transmissions. The ending voltage is the last recorded battery level received by the mote before transmissions ceased. Also included in Table 1 is the number of data packets that were received by the base station from each mote during its operation life. Each data packet represents one sample transmitted and collected from a mote to the base station.

Results, as shown in Table 1, indicate that the batteries from both series had variances in their starting voltages. There is a large amount of variability in the starting voltages of each test. This is due to the rechargeable nature of the batteries. While there are varied differences in the voltage drop for each mote, individual motes have a specific lower-end power requirement for operation as shown by the similar ending voltages in each mote's series of tests. The average battery life for the four motes with 2500 mAh batteries compared to the four motes with the 2450 mAh batteries is 4625 minutes to 2979 minutes (a 55% longer battery life in the 2500 mAh batteries). Given these results, it can be concluded that variability in the batteries, for example, differences in charging time, age of the battery, and so forth, is the major component in the variability of the battery life in the motes. This is not to say that the variability in the battery life is caused by the batteries alone. There is evidence of both individual variability (as seen in the ending voltage for each individual mote) and intermote variability (as seen in the differences in the ending voltages between groups of motes), albeit smaller than the influence of the batteries.

2.2. Experiments and Analysis on Data Message Intervals. The battery life for various DMIs in the two power modes was analyzed on twelve motes. The twelve motes that were selected were separated into four groups of three motes.

The battery life for a given DMI and power mode (HP or LP) pairing was tested on one of the four groups of motes. The three motes in each group were used to determine statistical parameters of the results. Each DMI and power mode pairing was tested three separate times. To reduce the battery variability, the motes were powered by two Panasonic Industrial (AM3) AA batteries, rated at 2870 mAh. The motes were once again tested using the default radio power level, 0 dBm (1 mW).

In HP mode, the DMIs tested were 10 s, 30 s, 60 s, and 900 s. The same DMIs were tested in LP mode except for 10 s. Due to the battery saving nature of the LP mode's sleep functionality, the 10 s DMI was deemed inappropriate (personal communication with Crossbow technician) and was intentionally excluded from the LP tests.

A summary of the battery tests is given in Table 2. The recommended operating voltage for motes is 3.6–2.7 V [14, Table 6-1], however it has been shown that the mote can continue to collect data down to 2.2 V and transmit messages down to 2.1 V [18]. Included for each DMI and power mode pairing is the average number of packets received and the average battery life over the total range of battery voltages. Next to each of the average values is the standard deviation based on the nine results (i.e., three groups of three motes per test).

Table 2 shows that for both the HP and LP modes, there is a small difference between the total battery life of the various DMIs. In the HP mode results, the 10 s and 30 s DMIs have a slightly longer battery life than the 60 s and 900 s sampling intervals. The battery life for all four of the HP tests are similar to the results obtained in Table 1 for the 2500 mAh batteries. The variability in the battery life results in Table 2 is also similar to the variability in the battery life of a single mote tested in Table 1. Therefore, it can be concluded based on these results that the sampling interval has no significant influence on the battery life, which is contrary to what was expected. These results would suggest that the sensed data transmissions regardless of sampling DMIs may not be a major cause of the battery energy consumption in the XMesh network. Further investigation was then conducted, as described in the next section, to determine the possible major causes for the battery life results shown in Table 2.

2.3. Experiments and Analysis on the Impacts of Health Messages. One possible cause for the battery life results shown in Table 2 would be the XMesh network's health messages. The network's health messages are sent to the base station periodically for updates regarding each individual mote's neighbor list and its own physical health statistics. The default configuration is to send the health messages every 60 s and 600 s for motes in HP and LP modes, respectively. The health message interval (HMI) is adjustable in the nesC application code. The mote alternates sending the neighbor information and its own statistics health messages at each transmission interval.

The transmission of the health messages could effectively reduce the mote's transmission interval to that of the HMI if the HMI is smaller than the mote's DMI. These health

TABLE 1: Summary of the mote and battery experiment of four tests on eight motes in HP mode at 1-second transmissions.

2500 mAh Batteries																
Mote	1				2				3				4			
Test	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
Start (volt)	2.55	2.79	2.72	2.85	2.46	2.71	2.64	2.70	2.63	2.87	2.80	2.97	2.47	2.69	2.59	2.76
End (volt)	1.97	1.97	1.97	1.97	1.87	1.65	1.87	1.87	1.99	1.99	1.99	1.99	1.86	1.88	1.87	1.86
Pkts. (thou)	102.9	156.1	147.1	143.6	100.7	87.9	145.1	143.5	99.7	154.4	147.1	149.9	98.4	150.1	135.5	142.3
Life (min)	3481	5342	5094	5090	3468	3038	5094	5095	3488	5383	5230	5381	3458	5296	4872	5191

2450 mAh Batteries																
Mote	5				6				7				8			
Test	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
Start (volt)	2.72	2.77	2.75	2.74	2.70	2.75	2.72	2.80	2.75	2.79	2.76	2.85	2.77	2.73	2.71	2.80
End (volt)	1.95	1.91	1.93	1.94	1.93	1.93	1.93	1.92	1.96	1.96	2.11	1.95	1.91	1.91	1.91	1.88
Pkts. (thou)	43.7	78.0	89.6	53.0	56.5	112.9	71.7	103.3	84.2	85.6	63.8	103.9	118.1	78.5	63.9	106.4
Life (min)	1570	2773	3182	1910	2043	4056	2576	3744	3072	3093	2318	3810	4351	2881	2346	3942

TABLE 2: Average battery life and transmission packets from nine motes tested in HP and LP modes at various sampling intervals with the standard deviations.

Power mode	Samp. Int.	Life (min.)	Data packets
HP	10 s	5545 (± 341)	29991 (± 1909)
	30 s	5618 (± 345)	10889 (± 645)
	60 s	4610 (± 217)	4499 (± 213)
	900 s	4597 (± 139)	308 (± 10)
	30 s	20502 (± 446)	21157 (± 3213)
LP	60 s	20859 (± 790)	14035 (± 2736)
	900 s	19952 (± 1942)	1179 (± 112)

messages sent to the base station can influence the speed in which the mote's battery power is exhausted. Thus, an experiment was conducted in attempt to quantify the effect of different HMIs on the mote battery power.

The same four groups of three motes that were used in the previous battery life measurements were once again used to test the effect of the HMIs on mote battery life. All four groups of motes were programmed to collect data in the LP mode and sample at a 900 s interval. The motes were powered with two Panasonic industrial 1.5 V AA batteries. The HMI for each of the four groups of motes was set to 120 s, 300 s, 600 s, and 900 s, respectively. The experiment took place indoors in a laboratory environment. In the laboratory, two groups (i.e., six motes) were positioned approximately 45 cm away on either side of the centrally located base station. Motes within each group were spaced a few centimeters apart.

The initial results from the health message testing revealed little insight on the effect of these message transmissions on battery life. The average battery life, shown in minutes, over the total range of battery voltages is shown in Table 3. Similar to the results shown in Table 2, there is again little difference between each group's average battery life. There is, however, a large difference between the average LP battery life between Tables 2 and 3. The battery life

TABLE 3: Battery life and transmission packets for various health message intervals at 900-second sampling interval in LP mode over the total battery life.

Health Int.	Life (min.)	Data packets
120 s	67304 (± 2174)	3655 (± 130)
300 s	67956 (± 641)	3672 (± 11)
600 s	68974 (± 1665)	3826 (± 63)
900 s	68112 (± 3697)	3818 (± 305)

results in Table 3 are over 200% longer than the LP results in Table 2. There was a single LP test completed, that was not included in the results in Table 2 because it was deemed an outlier but had an average battery life of 68000 minutes. It seems now that the outlying results from the DMI tests (as shown in Table 2) are in accordance with the results presented in Table 3 for the LP motes. This poses a new question concerning the reason for the low LP mote battery life presented in Table 2.

To see how the motes are performing over their battery life, the number of samples taken by each mote in the four groups, separated into voltage bins, is plotted in Figure 1. There is a distinct "double-hump" feature noticeable at around the 2.3 V and 2.65 V bins. The cumulative number of samples collected (as shown in Figure 2) shows that all the motes begin and end collecting approximately the same number of samples over their battery life. The variance present throughout the middle region may be explained by individual mote variability.

The similarities in battery life (e.g., Table 3) and samples collected (e.g., Figure 2) prompted further investigation on the mote's DMI over its battery life to understand these similarities. Figure 3 is an example of a typical mote's behavior over time. The plot of the battery voltage curve is based on the measurements from the data packets received by the base station. The DMIs are taken as the time difference, in minutes, between two successively received data packets. The number in parentheses indicates the count of data packets received at each given interval. It can be seen in Figure 3

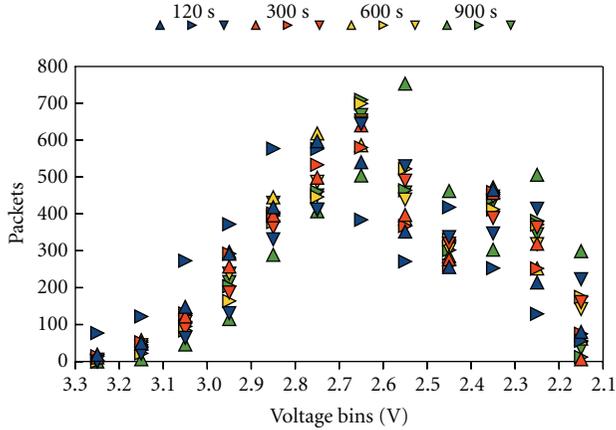


FIGURE 1: The number of data packets collected for notes in LP mode sampling at 900 seconds at various health message intervals. The four groups of three notes are plotted, each at their respective health update interval, that is, 120 s, 300 s, 600 s, and 900 s.

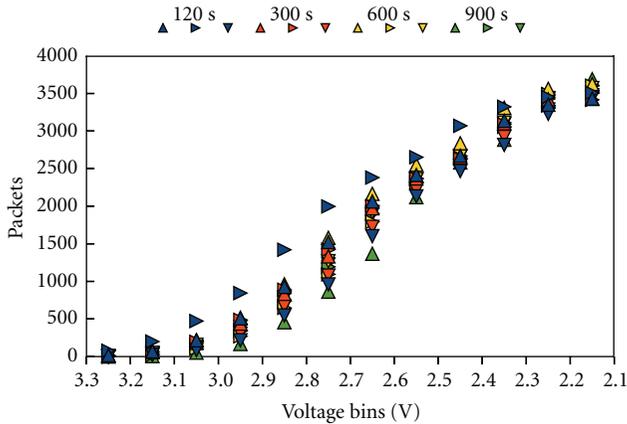


FIGURE 2: The cumulative number of packets collected over the battery life of notes in LP mode sampling at 900 seconds at various health message intervals. The four groups of three notes are plotted, each at their respective health update interval, that is, 120 s, 300 s, 600 s, and 900 s.

that the majority of the packets collected were made at the prescribed interval, that is, 15 min or 900 s. A large number of packets were also collected at twice the programmed interval, that is, 30 min. This indicates that the mote dropped one data packet between two successful transmissions to the base station. Fewer packets are shown to have been collected at intervals of 45, 60, 75, and 90 min, signifying that the mote dropped 2, 3, 4, and 5 data packets between successful transmissions to the base station. Using the time intervals between successfully received data packets, of the 3744 data packets received, there were an estimated 714 dropped packets (almost 20%). This is a considerably large number with respect to both the loss of data and the wasted power for transmissions. Therefore, the dropped packet rates of the wireless motes were more closely examined.

2.4. Experiments and Analysis on the Impacts of Dropped Packets. A dropped packet occurs when a mote’s data packet

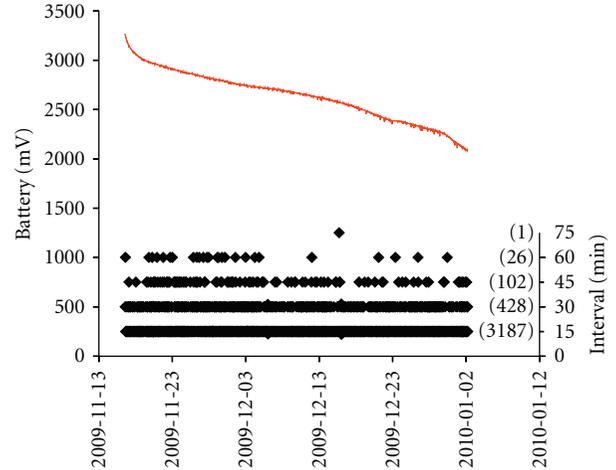


FIGURE 3: Mote battery voltage (in millivolts) over time (red line) in HP mode with a 15 minute DMI and the time interval (in minutes) between received data packets (black diamonds). The number of samples collected at each time interval is indicated in parentheses on the right side of the plot.

is unsuccessfully delivered through the network to the base station. This can occur at any individual node through the multihop network. The unsuccessful delivery of a data packet can be due to one of the four failure modes previously defined. In the transmission of a data packet over each consecutive hop, an acknowledgement message is returned to a mote by its parent to signify that the packet was received. If a mote does not receive an acknowledgement, it will attempt to resend the packet up to eight times until either an acknowledgement is received or the maximum number of retries is reached [19, Section 10.1.4]. If the maximum number of retries is reached without successful acknowledgment, the packet is “dropped” and results in lost data.

An analysis of the mote transmission performance is summarized in Table 4. The received packets in Table 4 are for the total transmission life of the motes as given in Table 3. Dropped packets were identified based on the time interval between successful receipts of data packets, as described above. Table 4 breaks down the number of data packets received into the number of packets received on time (no packet loss), the number of packets received with only one retransmission, the number of packets received after two retransmissions, the number of packets received after three or more retransmissions, and the number of asynchronous packets received.

Asynchronous packets occur when a mote does not receive an acknowledgement from its parent that its packet was received and therefore retransmits the same packet again. In some cases, the link quality from a mote to its parent is better than the link quality from the parent to the mote. This is also known as link quality asymmetry [9]. Under this circumstance, asynchronous packet delivery can occur and the base station will receive duplicate packets from a mote. The amount of this occurrence can be calculated by

TABLE 4: Transmission analysis of LP motes sampling at 900 seconds for various health message intervals.

Health Int.	120 s	300 s	600 s	900 s
Packets recv.	3655	3672	3826	3818
0 drop	84.3%	84.0%	84.9%	86.2%
1 drop	12.4%	12.8%	11.8%	11.2%
2 drop	2.4%	2.3%	2.4%	1.9%
2+ drop	0.6%	0.7%	0.7%	0.5%
Async.	0.3%	0.2%	0.2%	0.2%
Packets drop	713	728	735	638
Packets exp. ^a	4367	4399	4561	4456
Success rate	83.7%	83.5%	83.9%	85.7%

^aEstimates based on summation of data packets received and data packets dropped.

identifying two or more duplicate data packets received by the same node within a few seconds of each other.

The number of packets expected shown in Table 4 is an estimated value based on the sum of received and dropped data packets. The success rate, shown in the last row of Table 4 as a percentage, is the ratio of received to expected data packets. In each case, only about 80% of the packets expected were received. Furthermore, each of the four sets of motes had a similar number of dropped packets. To better understand the nature of packet drops, we further conducted an investigation into the multihop structure of the network.

2.5. Experiments and Analysis on Routing Usage. This WSN's architecture uses XMesh which features TrueMesh technology. This means that the network is self-healing and self-organizing and each mote acts as both a sensor node and a router for its neighbor's data. The ad hoc formation of the network is based on link estimates made between node neighbors in order to send data down a path of lowest transmission cost to the base station. As a consequence, certain motes may be exploited as a relay due to their low path cost.

A mote's parent ID is included in the data packet sent to the base station. The parent node is defined as a mote's neighbor with the lowest transmission cost [20, Section 4.2]. Thus, by analyzing the parent data, it can be determined if there are any motes being exploited and therefore having their batteries drained at a higher rate. Table 5 shows the results of this analysis. The number of health packets that each group sent is estimated based on the average battery life of each group and the HMI. The packets forwarded represent the additional transmissions motes make relaying neighbor data (i.e., data and health packets) through the network. To determine the multihop forwarding through the network, the distribution of each mote's connection with their neighbors was calculated based on the parenting information collected in the data packets. The same distribution was used to determine the routing of the data and health packets through the network. The number of packets generated is the summation of the

TABLE 5: Route-utilization analysis of the LP mote health message interval experiment (from Tables 3 and 4).

Health Msg. Int.	120 s	300 s	600 s	900 s
Data Pkts. Gen. ^a	4367	4399	4561	4456
Health Pkts. Gen. ^b	33652	13591	6897	4541
Data Pkts. Fwd.	1123	2790	2454	12714
Health Pkts. Fwd.	4575	11745	10191	52682
Route-util. ^c	15.0%	80.0%	110%	727%
Life (min.) ^d	77390	122859	145088	563209

^aExpected data packets based on the received and estimated dropped packets (see Table 4).

^bEstimates based on the battery life and health message interval.

^cRoute-utilization is the ratio of forwarded data and health packets to generated data and health packets.

^dTotal battery life, see Table 3, scaled based on the route-utilization percentage.

estimated number of data packets expected (see Table 4) and the estimated health packets (based on the battery life and HMI). The route-utilization is the percentage of additional forwarding transmissions compared to those generated by the mote (i.e., data and health packets) and is defined as the ratio of packets forwarded to packets generated as shown in the following equation:

$$\text{Route-utilization} = \frac{\text{Fwd}_{\text{data}} + \text{Fwd}_{\text{health}}}{\text{Gen}_{\text{data}} + \text{Gen}_{\text{health}}}, \quad (1)$$

where Gen_{data} and $\text{Gen}_{\text{health}}$ are the estimated total number of data and health packets generated by a mote and Fwd_{data} and $\text{Fwd}_{\text{health}}$ are the estimated number of data and health packets forwarded by a mote.

It can be seen that the route-utilization increases as the HMI increases. The motes with the 900 s HMI were utilized considerably more than those with the smaller HMIs. Table 5 shows the scaled battery life of each set of motes, as given in Table 3, based on the route-utilization percentage. The results of scaling the mote battery life show the expected trend in increasing battery life with decreasing the number of samples taken.

The MICAz mote is expected to have a battery life up to one year [12, Table 3-1]. This corresponds to the estimated battery life of the 900 s HMI results. Given the results in Tables 2 and 3 for LP motes, the scaling may be an exaggerated battery life adjustment. It is more likely that the increase in battery life compared to the default 600 s would be closer to 14% (logarithmic trend) or 29% (linear trend). Regardless, this test shows that decreasing the HMI from the default 600 s to either 300 s or 120 s will affect approximately 15–46% of a mote's battery life, respectively.

The analysis of scaling the battery life according to the mote's route-utilization can be applied to the results in Table 2. The adjusted HP results in Table 6 show no change in the battery life trend from the original results in Table 2. Given that the HMI for motes in HP mode is 60 s, it is expected that the data sampling rates of 60 s and 900 s HP results would be similar, that is, motes sending data every

900 s are also sending health packets every 60 s. The HMI, however, does not explain why with 10 s and 30 s sampling rates the HP motes' battery lives are about 25% longer than those with the 60 s and 900 s sampling rates. A more detailed analysis of the power consumption of HP motes of XMesh-based WSNs is necessary to understand this difference which is beyond the scope of this work. The adjusted battery life for the LP results in Table 6 show the expected trend in battery life with increasing transmission interval.

The route-utilization shown in Table 6 indicates that the LP motes with the largest data transmission interval, that is, 900 s, suffer the highest routing relays. This is similar to what was seen in the results of Table 5. For the HP motes, however, it can be seen that the smaller transmission interval (10 s) leads to the highest routing relays. From both tests, the HP motes have a consistently higher rate of successful transmissions (above 90%) while the LP motes have generally lower rate of successful transmissions over a varied range (78–92%).

While increasing the HMI can save some battery life, it reduces the amount of information collected regarding node link quality and path cost through the network. For purposes of monitoring the network mesh, the HMI may be more important than the battery savings. However, if network monitoring is not being considered, the health messages can be disabled completely to maximize battery savings.

3. Field Study

In addition to the laboratory tests, an experimental testbed consisting of eleven nodes was deployed in a residential backyard site (see Figure 4) located in western Pennsylvania (40.5436° N, 80.0638° W). The nodes range 6–60 m away from one another and are located mainly along the perimeter of the yard which is an approximately 60 m by 30 m rectangle. The wireless motes, data acquisitions boards, and battery packs were all housed inside polycarbonate high-impact enclosures (Bud Industries Inc.; part no. PN-1337). An external antenna (Pulse Electronics; part no. W1038) was attached to the outside of each enclosure. The enclosures were mounted on wooden stakes, placing the antenna approximately 0.3 m above the ground.

The network collected data starting from the summer of 2009 until the summer of 2010. All nodes were programmed in LP mode with the default radio transmission power (0 dBm) and health message interval (600 s). Motes were powered by two sets of two AA rechargeable batteries connected in parallel. The LED indicator lights on the mote were turned off due to their power consumption (6–8 mA) on the mote's batteries [12]. From 07/22–10/20 in 2009, all nodes sampled data at a 15 min interval. An analysis of the network's battery life and behavior was completed during this time. Figure 5 shows the received packets from each of the eleven nodes over the three month period.

Table 7 shows the battery life and packet analysis for each of the eleven nodes. The start and end dates correspond to the time that the transmission of data packets began and ended (see Figure 5). It should be noted that mote 5140 was

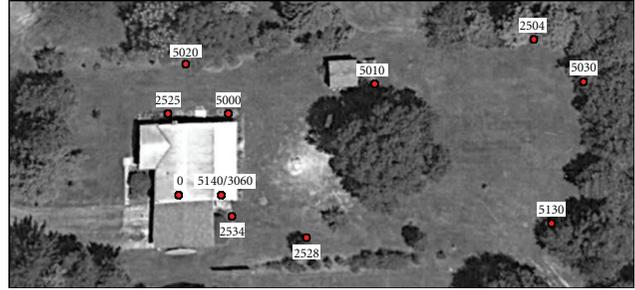


FIGURE 4: Residential backyard wireless sensor network testbed node locations during the late Summer and Autumn of 2009.

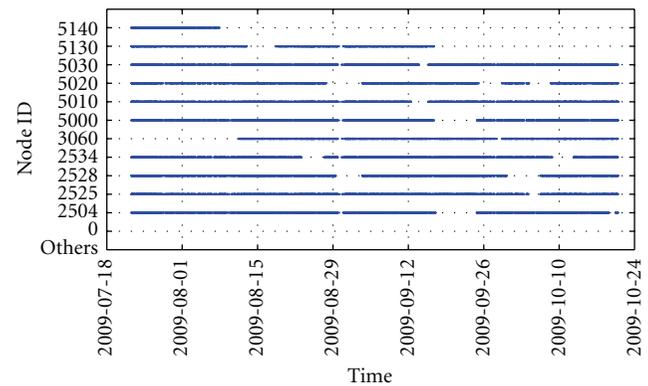


FIGURE 5: Received data packets by the base station for each node over the three-month monitoring period.

replaced by mote 3060 due to technical difficulties. Data was excluded from the analysis during periods where the motes did not complete a full battery cycle, that is, motes whose batteries were replaced before 10/20/2009 but were not fully depleted until after this time.

It can be seen from Table 7 that the outlier nodes (e.g., 2525, 2504, and 5030) have less forwarded packets and longer battery lives. While the outlier node 5130 did not forward many packets, its battery life was not as long as the other outlier nodes. This could be due to its isolated position not having a good connection to its neighbors. Nodes with the closest contact with the base station (e.g., 3060, 2534, and 2528) forwarded the most packets with battery lives on average less than the outlier nodes.

The success rate is approximately the same, around 40%, for all the nodes tested in the field. The high dropped packet rate may be attributable to the motes' close proximity to the ground. From the laboratory experiments the dropped packet rate for LP motes was found to be 10–20% compared to the 60% dropped packet rate in the field. The network's stability can be analyzed by examining the number of times a mote's parent changes. The high number of parent changes for the motes suggests that the link quality between the motes and their neighbors was low and fluctuated around the threshold for new parent selection. A large number of parent changes is representative of low network stability. This may have also attributed to the high dropped packet rate.

TABLE 6: Route-utilization analysis of the HP and LP mote sampling interval experiment (from Table 2).

Sampling Int.	HP				LP		
	10 s	30 s	60 s	900 s	30 s	60 s	900 s
Health Pkt. ^a	5545	5618	4610	4597	2050	2086	1995
Packets drop	156 (± 77)	12 (± 3)	2 (± 2)	0 (± 0)	5655 (± 4672)	1229 (± 472)	112 (± 41)
Packets Fwd. ^b	1855 (± 4099)	64 (± 157)	1 (± 1)	0 (± 0)	11525 (± 3962)	12294 (± 12498)	9531 (± 18069)
Packets Gen. ^c	35692	16519	9111	4905	28862	17350	3286
Route-util.	4.6%	0.5%	0.0%	0.0%	42.2%	66.5%	289.6%
Success rate	99.5%	99.9%	99.9%	99.9%	78.9%	91.9%	91.3%
Life (min.) ^d	5800	5646	4610	4597	29154	34730	77732

^a Estimates based on battery life and the default health message intervals, that is, 60 s for HP motes and 600 s for LP motes.

^b Forwarded packets are estimates of both data and health packets, taking into account the multi-hop network functionality, based on the statistical distribution of mote routing.

^c Packets generated is equal to the sum of a mote's total data and health packets.

^d Total battery life, see Table 2, scaled based on the route-utilization percentage.

TABLE 7: Battery life and transmission statistics of the eleven node prototype testbed network.

Node ID	Start date	End date	Life (min)	Packets rcvd.	Packets dropped	Packets Fwd.	Parent changes	Success
2504	07/22/2009	09/17/2009	81294.9	2152	3136	1080	1149	40.7%
	09/24/2009	10/19/2009	35322.6	895	1399	321	370	39.0%
2525	07/22/2009	10/04/2009	106103.0	2889	4018	371	495	41.8%
	07/22/2009	08/29/2009	54618.2	1432	2113	2029	686	40.4%
2528	09/03/2009	09/30/2009	38652.5	1121	1382	1946	477	44.8%
	10/06/2009	10/20/2009	20038.7	529	768	835	235	40.8%
2534	07/22/2009	08/23/2009	45425.7	1136	1802	1931	518	38.7%
	08/27/2009	10/08/2009	60846.7	1646	2287	2753	634	41.9%
3060	08/11/2009	09/28/2009	69075.5	1934	2567	3492	741	43.0%
5000	07/22/2009	09/16/2009	80909.1	2194	3116	2636	783	41.3%
	09/24/2009	10/20/2009	37022.7	945	1491	601	318	38.8%
5010	07/22/2009	09/12/2009	74773.6	1944	2947	1517	1007	39.7%
5020	07/22/2009	08/27/2009	52074.3	1312	2110	689	570	38.3%
	09/03/2009	09/25/2009	31147.3	923	1122	625	277	45.1%
5030	07/22/2009	09/13/2009	76867.8	2014	3013	663	938	40.1%
5130	07/22/2009	08/12/2009	30834.1	790	1281	363	410	38.1%
	08/18/2009	09/16/2009	42237.0	1214	1611	380	581	43.0%
5140	07/22/2009	08/07/2009	23470.4	752	807	674	213	48.2%

The field test results are comparable to the laboratory experiments for motes programmed in LP mode. In Tables 3 and 5, the results for motes with the smallest health message interval (120 s) are close to the field experiment results for nodes with higher traffic (e.g., more forwarded packets) such as motes 2504 and 5010. The comparable battery life (approximately 70000 min) and forwarded packets (approximately 1400 packets) show that the laboratory experiments were completed in a relatively high traffic condition compared to other motes in the field. It should be noted that under the same HMI, the scaled battery life from the laboratory experiment (Table 5) is about double the results in the field. The original battery life (Table 3) is similar to what was seen from the motes in the testbed. This would suggest that route utilization affects approximately 50% of the battery life. While there does not appear to be any correlation between mote location and the transmission

success rate, battery life does show some dependence on the network's topology.

An analysis on the route utilization in this field test was then performed. Based on the motes' parent IDs included in the data packets (see Figure 6), the link selection probability distribution for a packet to be forwarded by a mote through any of the mote's neighbors was calculated. Based on the conditional probabilities, a graph of the network topology was created where the vertices represent motes and each edge corresponds to a communication link associated with its selection probability. Considering the possibility of asymmetric links, the topology graph is a directed graph and its adjacency matrix is presented in Table 8. In this graph, the data for mote 5140 was included in mote 3060.

For this graph, each individual mote's link selection probability distribution is calculated based on the parent IDs included in its generated data packets, which represents

TABLE 8: Field test adjacency matrix for the network topology graph.

	0	2504	2525	2528	2534	3060	5000	5010	5020	5030	5130
0	0	0	0	0	0	0	0	0	0	0	0
2504	0	0	0	0.162	0.242	0.303	0	0.117	0.013	0.104	0.059
2525	0	0	0	0	0	0	0.631	0.002	0.367	0	0
2528	0.274	0.030	0	0	0.261	0.253	0.046	0.076	0.007	0.029	0.025
2534	0.478	0.025	0	0.17	0	0.221	0.008	0.046	0.019	0.022	0.011
3060	0.528	0.032	0	0.13	0.222	0	0.026	0.042	0.007	0.007	0.006
5000	0	0	0.038	0.348	0.134	0.188	0	0.191	0.094	0.001	0.007
5010	0	0.074	0.001	0.214	0.271	0.318	0.038	0	0.012	0.045	0.028
5020	0	0.001	0.107	0.129	0.168	0.310	0.230	0.041	0	0.003	0.011
5030	0	0.150	0	0.210	0.329	0.079	0.001	0.129	0.017	0	0.084
5130	0.005	0.185	0	0.228	0.168	0.135	0.021	0.119	0.015	0.124	0

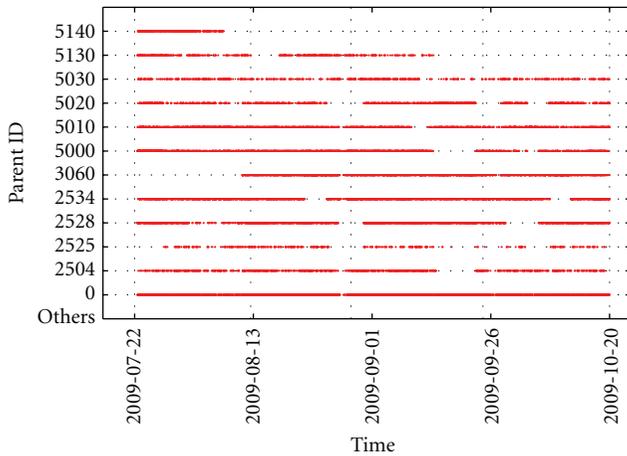


FIGURE 6: Parent ID for the received data packets for each node over the three-month monitoring period.

the behavior of the mote's first hop of its route towards the base station. In order to understand the overall routing behavior of the multihop network, it is reasonable to assume that the forwarded packets at each relay mote will follow the same link selection distribution of the generated packets at this mote for there is no parent ID information for relayed packets. Thus, in the following analysis, the link selection in routing for both originally generated packets and relayed packets at each individual mote follows the same link selection distribution.

The results of route utilization for each mote are presented in Table 9. The average route utilization for the field in this test is 105%. Given that the network is configured for a DMI of 900 s with a default HMI of 600 s, these results are comparable to the corresponding lab test with the same DMI and HMI (see Table 5). These results can be justified based on the higher traffic loads and route utilization in some specific motes. The analysis for the route utilization in each individual mote shows that motes 2528, 2534, and 3060 have the highest route utilization in the network followed by motes 5000 and 5020. The group with the smaller route-utilization consists of motes 5010, 2504, 5130, 5030, and

TABLE 9: Route-utilization percentage for each mote and network average.

Mote ID	Generated packets	Forwarded packets	Route utilization
2504	1524	742	48.7%
2525	2889	181	6.3%
2528	1027	2417	235%
2534	1391	2689	193%
3060	1343	2584	192%
5000	1570	2270	144%
5010	1944	1106	56.8%
5020	1118	1343	120%
5030	2014	444	22%
5130	1002	385	38.4%
Network avg.	1582	1416	105%

2525. These three groups of motes are highlighted in Figure 7 according to their route-utilization status.

Lastly, aiming to provide a major insight in the network dynamics, the highest probability routing links between nodes were selected based on the network topology graph (see Table 8). Based on these probability links, the most possible paths through the network for each node are presented in Table 10. From these results, it can be confirmed how motes with higher route utilization are used to forward packets from other motes. Since their main route is directly connected to the base station (mote 0), these motes are associated with a much higher route selection probability.

While the results from the field study show comparable results with experiments done in the laboratory setting, we note that there is still a considerable amount of missing information with respect to the effect of transmissions on the battery life of motes. The information transmitted via the health messages, which include a large portion of the missing information (e.g., retransmissions, dropped packets, forwarded packets, link quality, neighbor nodes, etc.), is not stored by default in the gateway for further investigation.

TABLE 10: Routes with highest probability for the different motes in the field test.

Mote ID	Route 1	Route 2
2504	3060–0 (16%)	2534–0 (11.5%)
2525	5000–2528–0 (6%)	5000–2528–2534–0 (2.7%)
2528	0 (27.4%)	2534–0 (12.4%)
2534	0 (47.8%)	—
3060	0 (52.8%)	—
5000	2528–0 (9.5%)	2528–2534–0 (4.3%)
5010	3060–0 (16.7%)	2534–0 (12.9%)
5020	3060–0 (16.3%)	5000–2528–0 (2.1%)
5030	2534–0 (15.6%)	—
5130	2528–0 (6.2%)	2528–2534–0 (2.8%)

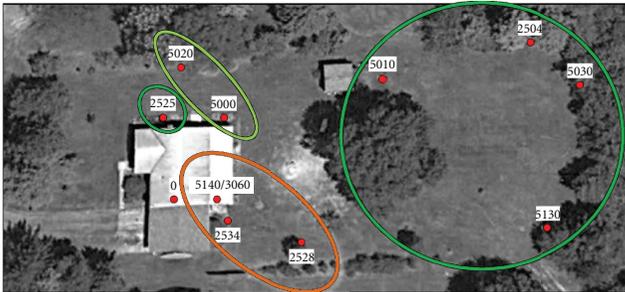


FIGURE 7: Locations of three mote groups in the prototype testbed according to their route utilization status. Groups highlighted in green correspond to the smaller route utilization, and orange corresponds to the highest.

One lesson learned in our experimental study is that the gateway in a WSN plays a critical role in the WSN reliability and robustness which could be easily overlooked at the beginning of WSN deployment. The gateway that was used in both the laboratory and field studies presented here was the Stargate Netbridge, a Linksys NSLU2 device specially modified by Crossbow. One attractive feature of the Stargate Netbridge is that it has a local web-based monitoring and management interface called MoteExplorer. MoteExplorer allows easy visualization of the network status, the current mesh topology, and a live stream of mote data collected by the base station, which made the Stargate Netbridge a favorable choice of the WSN gateway in deployment and operation. On the other hand, the Stargate Netbridge runs a Debian Linux operating system from an attached 4 GB flash drive. The deprecated version of Debian Linux came with limited and mostly outdated software. Due to the limited resources, in computational speed and memory storage, and an unconventional architecture (ARM), updates and upgrades to the device were not possible. With no direct access to the computer (monitor, keyboard, and mouse are not supported), all operations had to be performed through an SSH connection (for more information regarding the limitations on the Stargate Netbridge, see [21]). It has been discovered in our experimental study that the

Stargate Netbridge has severe reliability issues, and its strong limitations in hardware and extensibility make it an inconvenient solution for a practical WSN deployment. Following several unrecoverable crashes to the Stargate Netbridge, it was decided to move the gateway platform to a Linux x86 computer running Ubuntu Linux 10.04 operating system. While the new gateway platform would allow for faster, easier, and more reliable network operations, it raises an important challenge due to the lack of the convenient WSN management tool, MoteExplorer. In order to facilitate the change in the gateway platform to fundamentally improve the WSN system's robustness and reliability, we apply our general integrated network and data management solution [22] to the WSN testbed in this study. The following section describes the network and data management system and its application to the deployed outdoor WSN testbed. An example of its operation is also shown for the field study testbed described above.

4. Network and Data Management System

A web-based integrated network and data management system called INDAMS, presented in [22], has been applied to the WSN testbed to facilitate the management and monitoring needs of real-world WSN deployments. This management system employs the following fundamental features: (1) separates the WSN management functions from WSN applications, (2) utilizes an accessible web-based user interface for management functionalities, and (3) systematically supports multiple WSNs from multiple platforms and technologies. Multiple users cannot only independently in real-time remotely access the WSN testbed operations via this management system, but also retrieve and monitor sensing data and network management information with unified web-based management tools that may not be available at local gateway system(s). This eliminates the complexity involved with users dealing with the complex commands and configurations of the WSN gateway (e.g., Memsic's XServe). Instead, this management system enables users to access the important and critical WSN management data for conducting network analyses. Integrated with the new gateway platform in this study, INDAMS not only successfully addresses the issue of regular network management needs in the operations of the WSN testbed, but also collects and saves the network health statistics that were previously discarded, most likely due to storage requirements of the Stargate Netbridge, such that comprehensive power and routing analyses can be conducted. In this section, we highlight some key aspects of the web-based WSN management system INDAMS, with the focus on its interaction with Memsic's XServe gateway used in our study, and the partial deployment in the WSN testbed (for the detailed description of the general INDAMS development, see [22]).

4.1. Management System Architecture. The overall architecture of the management system is illustrated in Figure 8. For the management server to communicate with the WSN gateway (in this case XServe), a representative agent is

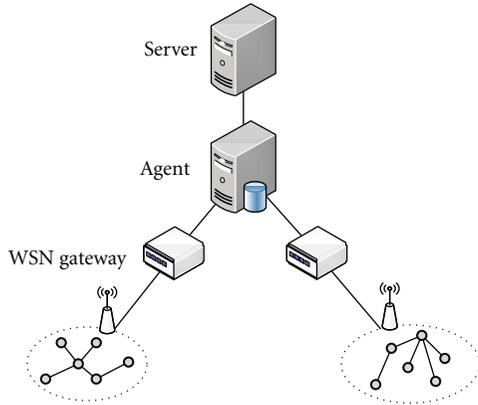


FIGURE 8: An illustration of the management system's overall architecture. The main server of the management system communicates with an on-site agent which interacts with the WSN gateway and network.

introduced which works directly with the WSN gateway. A management protocol, that is, the agent-server protocol, is designed to formally communicate between the management server and the agent. In this respect, the system's architecture is generic such that all management functions developed on the management server are independent to the WSN gateway platform. For example, in the event that the XServe gateway is replaced by another gateway platform only the agent's gateway interface needs to be modified. Thus, the agent-server protocol is the key component of the design and implementation of this management system.

The agent-server protocol is an application-layer protocol that is carried by TCP for reliable transmissions. In general, the functionalities of the management system are classified into two categories: control request/response functions and data functions. The agent-server protocol defines a specific control connection and a specific data connection for the control functions and data functions categories, respectively. The message exchange sequence of the agent-server protocol depends on each specific function request. An example of one of the most complex scenarios which involves message exchanges over both control and data connections between the management agent and the server is shown in Figure 9.

In Figure 9, the sequence of messages starts with messages m_1 and m_2 for the registration process. After m_2 is received, the agent waits for requests. A client request (e.g., a remote user's request through the internet) to start data collection is translated and forwarded by the server, that is, m_3 , which triggers a response by the agent, that is, m_4 and m_5 . After m_5 is transmitted by the agent and received and processed by the server, the agent and server resume waiting for requests.

The data function represented in Figure 9 begins with a monitoring request, that is, m_6 , from the server to the agent which responds with message m_7 . The difference between the data collection and the monitoring request is the continuous use of the data connection for sending and collecting data from the agent by the server. When m_6 is processed, the agent

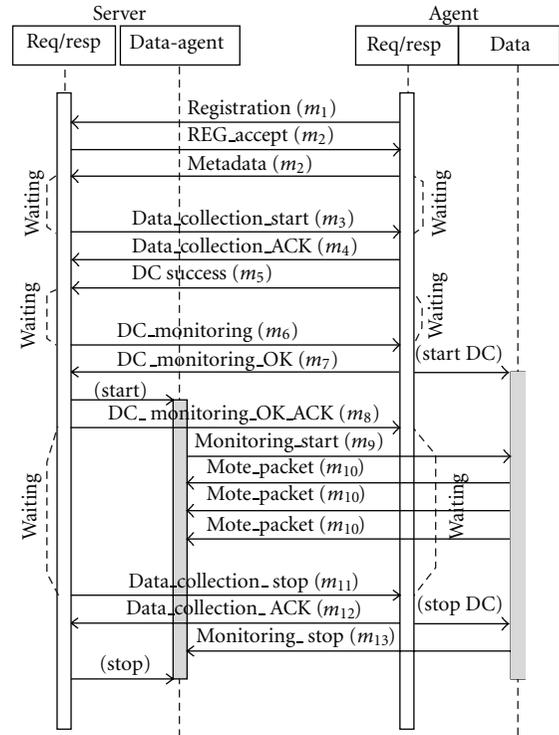


FIGURE 9: A complex data collection sequence which involves message exchanges over both control and data connections between the management agent and the server. The direction of each message's transmission is indicated by the message arrow, with the message's name and number shown above the arrow.

waits for requests on the data connection. After the server receives m_7 , the server starts the data connection for the agent and sends an acknowledgement message back to the agent, that is, m_8 . Following the transmission of m_8 , the server waits for client requests. When m_8 is received, the agent waits for requests to be forwarded by the server, while the message sequence continues on the data connection.

At this point, both the control and data connections are active. The server sends a monitoring data request, that is, m_9 , to the agent which then begins sending the data packets back to the server, that is, m_{10} . Finally, the sequence to stop the monitoring function is given, that is, m_{11} , m_{12} , and m_{13} . The data connection is finished with the closing of both sides of the protocol. Note that many acknowledgement messages are used to synchronize the execution at both sides of the protocol and to detect any possible errors at the other side.

In the management system, client requests are processed by the control and data handler component. The data handler subscribes clients as event listeners and assigns them parameters according to their request. After receiving data from the protocol server, the data handler decides which clients are going to receive the data and does so in the appropriate format. The data handler is capable of differentiating data sent from multiple agents and the data type. Moreover, it is a simple task to add new clients and new types of parameters. In this way, the management system

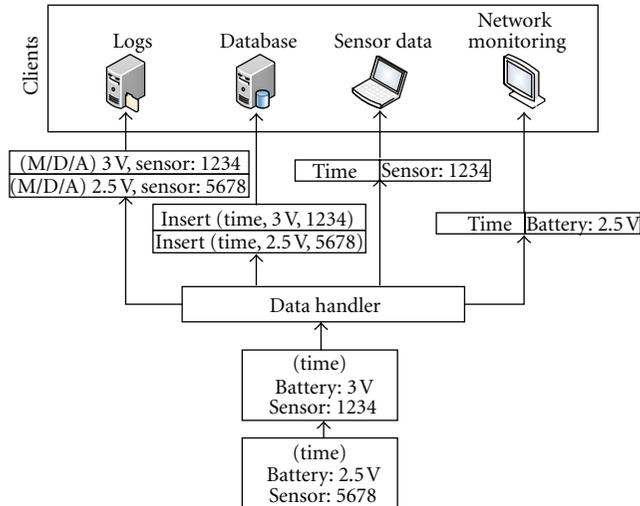


FIGURE 10: An illustration of operations of the data handler component when two packets are received. The data handler sends the notification information to each of the corresponding subscribed listeners separately.

provides flexible functionality to ensure the proper delivery of data to the appropriate clients. The process performed by the data handler in the control and data handler component is shown in Figure 10.

The agent for XServe WSN gateways was implemented in Java as a standalone application. The XServe agent implements the agent-server protocol with a communication interface to the management system. For this study, XServe's functions were classified into different types, such as data collection and network monitoring.

Xserve runs on a variety of platforms including Linux x86, which is the platform of the new gateway, and also uses a set of parameters to activate different functions of the protocol. A technology-specific section of the agent's metadata was assigned to store the parameters and values required for the agent to start XServe's application. XServe also provides an interface, that is, XServeTerm, to allow external applications to send commands, that is, XCommands, to the WSN or to the gateway itself.

The XCommands required by the agent are included in the agent's metadata for the purpose of mapping the function requests. Each request received by the agent can be mapped to a combination of parameters for XServe and XServeTerm. Therefore, when the agent receives a request, it checks the metadata for the appropriate mapping and syntax and communicates with XServe and XServeTerm applications via Java interprocess communication mechanisms. An example of the command mapping between the agent and the server for the data collection function, shown in Table 11, considers only the basic function parameters, that is, start and stop, which leaves all remaining values to default. In this case, the mapping for the request to start the data collection corresponds to a single command that executes XServe with a set of parameters. The mapping to stop the data collection corresponds to a single command that executes

TABLE 11: An example of agent/server command mapping.

Protocol server command	Agent command	
	XServe	XServeTerm
Data collection: Start	<code>Xserve-db-c-s=<COM/USB port>-xmlport=<xml port></code>	(not required)
Data collection: Stop	(running)	<code>xserve term xserve.shutdown</code>

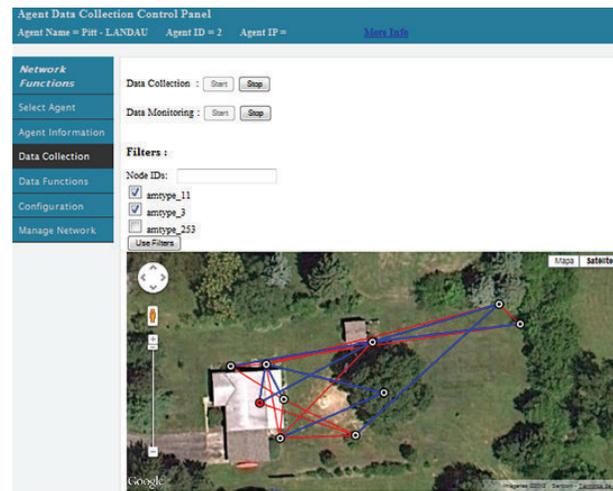


FIGURE 11: A recent screenshot from the web interface of the management system showing the topology monitoring feature for the prototype testbed. Blue links represent the link chosen by a mote to send a packet and red links represent the neighbors of each mote. Some adjustments have been made to the node locations since the network was originally deployed.

XServeTerm with the XCommand, that is, `xserve.shutdown`. More complex functions may require a combination of XServe and XServeTerm with different parameters and different sequences of execution.

4.2. Field Deployment. The developed management system is deployed and tested in the experimental testbed described in Section 3. The management system offers a geographical and topology monitoring feature as shown in Figure 11. This allows users to see a map of the mote locations in the WSN. Each node on the map shows the connections to each of mote's neighbors and highlights the mote's parent. The neighbor and parent information is received from the data and health packets transmitted by each mote. The map is updated continuously as the information is being received by the server. This provides a useful tool for a fast and updated view of the network's state and routing for each of the motes.

The management system also includes a live feed of the information received from the WSN gateway displayed in a rolling table as shown in Figure 12. All packets received from

Time Stamp	Packet Type	Node ID	Data
2012-01-31 13:09:55.172	3	5020	neighborID2 3039
			neighborID1 2534
			neighborID3 3060
			node_count 3
			old_type 15
			pathcost2 4
			pathcost3 9
			pathcost1 9
			rsid2 0
			rsid1 0
			rsid3 0
2012-01-31 13:09:53.378	253		
2012-01-31 13:09:48.496	253		
2012-01-31 13:09:45.614	253		
2012-01-31 13:09:28.55	11	2525	adc5 0.610352
			adc4 0.610352
			adc6 0.610352
			adc1 59.204102
			adc0 823.974609
			adc3 0.610352
			adc2 656.127930
			board_id 129
group 130			
humidity 47.833076			
humidity 13.780000			
parent 5020			
packet_id 137			
socketid 51			
voltage 3772			

FIGURE 12: A screenshot from the web interface of the management system presenting the live feed information from the prototype network.

the motes, that is, data and health, and base station, that is, heartbeats, are displayed. Filters were implemented to specify the types of packets, for example, data or health, or packet fields, for example, specific node IDs, to be displayed.

Figure 12 shows the three types of data packets that are collected at the testbed site. The first is the health packet, type 3, which is giving the current neighbor statistics for node 5020. The second packet type shown is the base station heartbeat, type 253. This shows the user that the base station is on and working. The third type is the data packet, type 11, which shows the mote's sensor data, current configuration and parent ID used to transmit the data. All of the data that is collected, with the exception of the base station heartbeats, is stored in a database on the gateway and is recorded with their received time stamp by the gateway.

5. Conclusions

This paper thoroughly examines the power efficiency and battery savings, as well as some key characteristics of wireless mote transmissions in both laboratory and field settings, based on experimental study and following a quantitative approach. The major contributions of this work are summarized as follows.

(1) This study shows that motes in LP mode drop between 10–20% of their sampled measurements (see Tables 4 and 6). This is one of the drawbacks of operating the Memsic MICAz motes in LP mode, as opposed to the HP mode. In contrast, in the HP mode, mote measurements at 15 min intervals have less than 1% packet loss occurrence. While the LP mode decreases the success rate of data transmission compared to using the HP mode, it supplies around four times the battery life in comparison with that in the HP mode (see Table 2).

(2) We show that by adjusting the health message interval, an additional 15–46% in battery life can be gained (see Table 5). In addition, the multihop capability of the network can exploit certain motes which have relatively low path cost to the base station. This leads to motes serving as relays to have a shorter battery life than those that do not. This investigation shows that route-utilization can result in over 50% reduction in the battery life in motes in LP mode in the laboratory setting (see Table 6). Little to no route utilization was found in HP motes.

(3) The field study shows that the battery life of the LP motes was comparable to those tested in the lab. Route utilization was identified in motes with good link quality with the base station. The battery life of these motes was variable but generally less than motes located on the outer edge of the network which forwarded less packets. The dropped packet rate was significantly higher in the field than that in the laboratory setting, about 60% compared to 10–20%, respectively. The higher rate of dropped packets may be due to the low clearance of the motes above the ground and the stability of the network which was found to be questionable due to the high occurrence of parent changes.

(4) Our study reveals the practical vulnerability of the original Stargate Netbridge gateway in WSN testbed deployment, when the deployed WSN moderately scales up. To address this critical issue, a new gateway platform integrated with our general web-based WSN management solution is presented. Our solution not only succeeds in replacing the Stargate Netbridge gateway for reliable WSN deployment and operations, but also enables the network management data collection of all the important network health and neighbor statistics information which was not possible before due to the limitations of the Stargate Netbridge gateway. Indeed, this will allow for more comprehensive studies of the power consumption of WSN deployments in the future. Our network management system also further improves upon the functionality of the original web-based monitor provided by Crossbow on their Stargate Netbridge. This testbed study demonstrates the merits of the general integrated WSN network and data management system INDAMS presented in [22].

In conclusion, our work provides new quantitative insights into the power characteristics of practical WSNs using the currently most popular and commercially available wireless sensor networking technology, which would be useful toward energy-efficient WSN developments for environmental monitoring in real-world practice.

Acknowledgment

This work was supported by NSF under CNS-0721474 and CNS-0758372 to the University of Pittsburgh and to IUPUI, respectively. Special thanks go to Mr. and Mrs. Lucas Landau for allowing us the use of their property for this field study. Thanks also go to Mr. Thomas Hare for his programming and analysis of the network data. Lastly, we thank all of the people whose time making comments and suggestions has helped to improve this work.

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] Y. Liu, Y. He, M. Li et al., "Does wireless sensor network scale? A measurement study on GreenOrbs," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '12)*, pp. 873–881, April 2011.
- [3] G. Werner-Allen, K. Lorincz, M. Ruiz et al., "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, 2006.
- [4] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "SensorScope: application-specific sensor network for environmental monitoring," *ACM Transactions on Sensor Networks*, vol. 6, no. 2, article 17, 2010.
- [5] N. Xu, S. Rangwala, K. K. Chintalapudi et al., "A wireless sensor network for structural monitoring," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys'04)*, pp. 13–24, ACM Press, New York, NY, USA, November 2004.
- [6] Z. Chaczko, A. Kale, and C. Chiu, "Intelligent health care—a motion analysis system for health practitioners," in *Proceedings of the 6th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 303–308, Brisbane, Australia, December 2010.
- [7] J. Beutel, K. Römer, M. Ringwald, and M. Woehrl, "Deployment techniques for sensor networks," in *Sensor Networks: Where Theory Meets Practice*, G. Ferrari, Ed., pp. 219–248, Springer, Berlin, Germany, 2010.
- [8] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, "The hitchhiker's guide to successful wireless sensor network deployments," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, pp. 43–56, ACM Press, New York, NY, USA, 2008.
- [9] A. Teo, G. Singh, and J. C. McEachen, "Evaluation of the XMesh routing protocol in wireless sensor networks," in *Proceedings of the 49th Midwest Symposium on Circuits and Systems (MWSCAS '06)*, pp. 113–117, IEEE, San Juan, Puerto Rico, August 2007.
- [10] D. Gay, P. Levis, R. V. Behren, M. Welsh, E. Brewer, and D. Culler, "The nesC language: a holistic approach to networked embedded systems," *ACM SIGPLAN Notices*, vol. 38, no. 5, pp. 1–11, 2003.
- [11] "TinyOS," 2010, <http://tinycos.net/>.
- [12] Crossbow Technology, Inc., *XMesh Users Manual (Doc.#7430-0108-01) Rev. D.*, San Jose, Calif, USA, 2007.
- [13] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, New York, NY, USA, September 2002.
- [14] Crossbow Technology, Inc., *MPR-MIB Wireless Module Users Manual (Doc.#7430-0021-08) Rev. A*, San Jose, Calif, USA, 2007.
- [15] T. Davis, X. Liang, C.-M. Kuo, and Y. Liang, "Analysis of power characteristics for sap flow, soil moisture and soil water potential sensors in wireless sensor networking systems," *IEEE Sensors Journal*. In press.
- [16] A. S. Chipcom, "SmartRF CC2420 Preliminary Datasheet," 2004, <http://inst.eecs.berkeley.edu/~cs150/Documents/CC2420.pdf>.
- [17] IEEE 802.15.4 Standard-2003, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE-SA Standards Board," 2003.
- [18] E. R. Musaloiu, A. Terzis, K. Szlavec, A. Szalay, J. Cogan, and J. Gray, "Life under your feet: a wireless soil ecology sensor network," in *Proceedings of the 3rd Workshop on Embedded Networked Sensors*, pp. 51–55, Cambridge, Mass, USA, 2006.
- [19] Crossbow Technology, Inc., *MoteView Users Manual (Doc.# 7430-0008-05) Rev. A.*, San Jose, Calif, USA, 2007.
- [20] Crossbow Technology, Inc., *XServe Users Manual (Doc.# 7430-0111-01) Rev. E.*, San Jose, Calif, USA, 2007.
- [21] F. Stanjo, D. Cvrcek, and M. Lewis, "Steel, cast iron and concrete: security engineering for real world wireless sensor networks," in *Proceedings of the 6th Applied Cryptography and Network Security Conference*, vol. 5037 of *Series Lecture Notes in Computer Science*, pp. 460–478, Springer, June 2008.
- [22] M. Navarro, D. Bhatnagar, and Y. Liang, "An integrated network and data management system for heterogeneous WSNs," in *Proceedings of the 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS '11)*, Valencia, Spain, October 2011.

Research Article

Wireless Sensor Network for Environmental Monitoring: Application in a Coffee Factory

J. Valverde,¹ V. Rosello,¹ G. Mujica,¹ J. Portilla,¹ A. Uriarte,² and T. Riesgo¹

¹Centro de Electronica Industrial, Universidad Politecnica de Madrid, Jose Gutierrez Abascal 2, 28006 Madrid, Spain

²INKOA Soluciones Agroalimentarias, Ribera de Axpe 11, 48950 Erandio, Spain

Correspondence should be addressed to J. Valverde, juan.valverde@upm.es

Received 15 July 2011; Revised 4 November 2011; Accepted 10 November 2011

Academic Editor: Yuhang Yang

Copyright © 2012 J. Valverde et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless sensor networks have been a big promise during the last few years, but a lack of real applications makes difficult the establishment of this technology. In this paper a real monitoring application in an instant coffee factory is presented. This application belongs to the group of environmental solutions based on wireless sensor networks, and it is focused on the impact of the instant coffee production processes in one of the largest instant coffee factories in Europe. The paper includes the entire application scenario, from the hardware of the WSN nodes to the software that will evaluate the impact and will close the loop.

1. Introduction

The environmental care has become one of the biggest concerns for almost every country in the last few years. Even though the industrialization level has been increasing without any control in the last decades, the current situation is clearly changing towards more environmentally friendly solutions.

Water and air quality are essential to maintain the equilibrium between human development and a healthy environment. It is also important to notice that by means of looking for a more efficient production in factories both pollution and consumption of natural resources can be decreased. Processes, such as boiling, drying, binding, and so forth, are being carried out by almost every kind of the current factories. Those processes are responsible of a great amount of gas emissions and polluted water discharges. Although the majority of the factories have their own sewage plants, it is crucial to measure the quality of the waste water that is being poured into the public sewer.

Due to the reasons above, the necessity of monitoring production processes and environmental parameters has become an essential task for the industrial community. The solution proposed in this text is an environmental monitoring system based on a wireless sensor network (WSN hereafter) platform called Cookies [1], to measure both gas

emissions and waste water quality in an instant coffee factory in Spain. Even though there are myriad other approaches that are now being used, WSNs can offer a cheaper solution while having data acquisition in real time, working in an unattended way.

Typically, the environmental data acquisition in factories is carried out manually and occasionally or using wired systems that are normally expensive and not flexible. This solution is not the best in terms of security, as it is necessary to hire workers to take measurements in dangerous places such as chimneys or waste water pipes. In this way, if a catastrophic discharge occurs, the factory will not notice it until next measurement, which can be several weeks later. That is one of the reasons why WSNs can offer a more reliable and safe measuring process.

In the state of the art, it is possible to find quite a few environmental solutions based on WSNs such as watershed monitoring systems [2], health monitoring in rivers [3], or energy management solutions to reduce both the amount of resources needed and the atmospheric emissions [4].

The work presented in this paper shows a closed-loop system which involves all the different steps to make the environmental impact assessment (EIA from now on) in a coffee factory. Each of these steps will be carried out by different companies and research centers. First of all, the environmental analysis will take place in an instant coffee

factory in Spain. As it was mentioned before, the WSN platform used to take the measurements is the one developed by the Universidad Politécnica de Madrid (UPM) called Cookies. The WSN will send all the data to a web tool designed by Inkoa, the company in charge of the project management. Finally, once the information is stored, a research center called Gaiker will make the EIA in order to report the results to the factory closing the loop.

This work is developed under the umbrella of the SustenTIC Project, which is a technology demonstration project, funded by the Spanish Government. It began in 2008 with the study of the processes and parameters to be measured in the factory and will finish with the deployment of the network and the evaluation of the environmental data in the summer of 2011. The project covers many different aspects related to WSN management for continuous monitoring, such as network communications quality, power consumption, state of the sensors, and so forth. But what makes it different from other applications in this field are the tools to evaluate the incoming data. Those tools will store and compare the data taken from the factory with different environmental data bases in order to make the EIA and being able to report the results directly to the factory.

This project will cover all the different steps to reach this very ambitious target, starting from the study of the processes in the coffee manufacturing and finishing with the results of the platform deployment.

The rest of the paper is organized as follows. Section 2 details the state of the art related to WSN applications for environmental monitoring. Section 3 describes the application scenario in order to obtain a better understanding of the main goals of the project. Section 4 gives a general outlook about food industry and its sustainability explaining the processes followed to carry out the EIA. Section 5 details the Cookies architecture and the new hardware and software needed to cover the requirements of this specific application. Section 6 shows how the measurements were done and the obtained results. Section 7 gives a summary of the different steps to close the loop. Section 8 shows the lessons learned during the project and concludes the paper.

2. State of the Art

In this section, the different solutions that already exist in the state of the art related to environmental monitoring will be detailed.

WSNs and environmental care have been always very close to each other, either because they have been evolving alongside one another during the last decade or because WSN features seem to suit very well into the environmental evaluation requirements.

One of the first references available of a WSN deployment for environmental issues is the habitat monitoring analysis made by UC at Berkeley University [5]. This is one of the first long-term and large-scale deployments in the state of the art with four-month duration and more than one hundred Mica2 motes using TinyOS. The main goal of this deployment was the observation of seabird nests and the effect of the microclimatic changes in its placements

along the Great Duck Island in Maine. By using passive infrared (PIR) sensors to measure heat and by testing both temperature and humidity variations in the environment, they were able to evaluate the effects of a prolonged occupancy in the habitat during a long period of time, even though the burrow-deployed nodes failed in a couple of weeks. The difficulties of this kind of deployments are clear due to the large areas that need to be covered. By contrast, in the case of the coffee factory, the problems will appear due to the signal attenuation between nodes caused by the big amount of metallic tanks and machines that are all around the factory and not because of large distances.

When talking about environmental applications for WSNs, there is a relevant work made by the CSIRO Center in Australia [6] which covers both a complete set of environmental and agricultural applications and a complete review of past and future opportunities in WSN applications.

Some of these applications are cattle and ground water quality monitoring, virtual fencing, rainforest and lake water quality monitoring, and so forth. As well as the previous example, the first deployments were done using the Berkeley Mica2 motes. Although after using the Berkeley motes for these first cases, they built their own motes, Flecks, and its different upgrades. It is important to emphasize the big reliability of these nodes in long-term applications since they were working for almost one year and a half with only two maintenance visits. Apart from that, they used a very well-known operating system, TinyOS, which can be very useful in these large-scale deployments when at the same time allows the programmer to build the application without taking care of the hardware behind. They even made a modification of the operating system called FOS. In the case of the SustenTIC project, the deployment consists of less than ten nodes, and the Cookies platform already has an application code based on custom libraries instead of a use of an operating system.

Since this paper is focused on a specific application deployment, there are other interesting references like the one called "The Game of Deployment" [7] and some others facing the differences between deploying a WSN in different scenarios. This first work shows interesting aspects in how the deployment should be faced by means of showing the problem as a game. In this game, the board is the map where the nodes have to be deployed and the counters are the nodes themselves. In this way, the map can be seen as a group of tiles with different placement restrictions where the way of placing them is a calculus optimization problem.

Even though through this kind of games the different scenarios can be generalized, it is possible to find other works showing that not all of them can be seen in the same way. The University of Trento has evaluated the huge differences between three different scenarios that at first glance seemed to be similar [8]. Those scenarios are a traffic tunnel, a mine tunnel, and a vineyard where they analyzed the communication problems caused by traffic, humidity, temperature, light, and how they affect the quality of the measurements in each of these different places. This is a very interesting study, since it shows both the problems and challenges of deploying a WSN and how these issues can change from one place to another. In the SustenTIC Project,

a very specific scenario is shown where these deployment troubles are faced and evaluated.

Apart from general environmental applications, it is important to analyze more in detail those applications focused in both water and air quality monitoring.

There are quite a few applications related to water monitoring. Most of them are being carried out in China, where pollution has become one of the biggest problems for the current government. For example, in [8], the authors propose a monitoring system to measure parameters such as pH, dissolved oxygen, conductivity, and temperature for aquaculture, river, and lakes monitoring, where the convenience of using this kind of systems in terms of price, flexibility, and real time processing is explained. Even though they talk about low price and low power, there is no reference neither of the power consumption of the node nor from the sensor prices which, in this case, can be very high. In [3], water parameters such as turbidity, pH, and so forth are measured in an artificial lake in the Hang-Zhou Dian-Zi University. Samples were taken every one hour during approximately one month, and the values were compared during night and day through five different points in order to compensate the measurement drifts. This is an other important difference. While taking measurements every one hour can be enough for a lake or a pond, in the case of water monitoring for waste water in a factory what is interesting is to have data every five minutes or so, to be able to generate alarms in case of a problem in the sewage plant occurs.

Another effort related to water monitoring applications is exposed in [2], with a study of a river watershed in North Carolina. They present an end-to-end hardware-software structure to monitor water parameters through a large number of highly distributed sensor networks. The sensor nodes used are also stackable, like the *Cookies* platform, which allows some kind of modularity. In this case they installed in the same PCB both the conditioning circuits for six different analog sensors and a communication module based on an XBee platform. In the *Cookies* platform, the functionalities of the node are separated in layers so that it is possible to combine different sensors with different communication modules with different frequency of operation. The second layer of their platform consists of a general purpose microcontroller (uC) (AVR) and a basic power regulator with a noise filter. In the platform used in this paper, by contrast, both power supply and processing functions are placed in different layers to allow different combinations of powering and processing as it will be explained in Section 5. They also have an in situ collection and uplink infrastructure and a representation tool for the data management. It is also important to notice that they combine different kind of motes making a heterogeneous network formed by commercial data loggers and custom boards as the ones explained before.

To finish with the water monitoring applications, in [9], some recommendations of water quality measurements are detailed. This application shows a water monitoring system at river basin level based on WSNs. They also compare the advantages of these wireless systems with the lab methods, highlighting factors like price or real-time

measurements. This study is part of a project called DEPLOY to demonstrate the use of a WSN in the river lee in Cork (Ireland). The main goal of this application is to comply with the regulation imposed by the European Water Framework Directive (WFD) by measuring the water quality in five different areas of the river.

Apart from testing the advantages of having continuous data from the river, they also want to demonstrate the reliability of the measurements by comparing the results of the multistation system when at the same time they also face problems such as fouling sensors, power management, or the study of available sensors for the application. These aspects will be also studied in this paper for this specific application.

While the majority of these applications are focused in natural environments, the deployment shown in the SustenTIC project is based on the requirements imposed by the government for industrial emissions and water discharges. This implies a very important difficulty due to the low amount of places with the same or similar features than the place of the final deployment, to make previous tests.

3. Application Scenario

In this section, all the details about the application presented in this paper are to be explained. This research work is part of a demonstration project funded by the Spanish Ministry of Industry, called SustenTIC, whose main goal is to evaluate the environmental impact in an instant coffee factory.

First of all, the features of the deployment scenario are shown. As it was mentioned before, the deployment will take place in an instant coffee factory in Spain. This factory can be considered as a medium size factory even though it is one of the most important instant coffee manufacturers in Europe. This kind of scenarios has a very important handicap when talking about wireless communications because of the problems caused by factory machines and the presence of metallic objects such as tanks, pipes, and intense trucks traffic.

The environmental measurements will take place in two different areas in order to cover two different kinds of quality parameters. The first place is the sewage treatment plant where both pH and water temperature need to be monitored. The main challenges in this measuring point are the corrosive environment and the sensor soiling, apart from the attenuations caused by metallic objects and the traffic around the sewage plant house which is placed in a separated building than the rest of the factory facilities.

In addition to the water quality measurements, it is also very interesting, from the environmental point of view, to monitor the gas emissions caused by the factory processes. The parameters chosen for this demonstration project are carbon monoxide, nitric oxide, and sulfur dioxide, since some of them can be very dangerous for human health and it is necessary for the factory to comply with the government regulations. There are three different chimneys where the measurements need to be done, these are

- (i) the boiler: carbon monoxide (limit value 500 ppm), sulfur dioxide (limit value 4300 mg/m³N), and nitric oxide;

- (ii) the drying tower: carbon monoxide (limit value 500 ppm), nitric oxide;
- (iii) the toaster: carbon monoxide (limit value 500 ppm), sulfur dioxide (limit value 4300 mg/m³N), and nitric oxide.

It is important to notice that, even though the factory has two different production lines, the demonstration will be done only in one of them. In contrast with the previous water measurements, the main challenges in this case are due to the gas conditions inside the chimneys. The first problem is caused by the output temperature of the emitted gas which can be more than 100°C when the temperature range of the sensors is only -30°C to 50°C. Therefore, to face this problem it is necessary to cool down the output gases. Another problem is the volume variation, since these gas sensors need approximately a constant flow to give a stable response. In order to solve this situation, the air needs to be sucked out from the chimney into a spiral tube and then into an isolated box where the sensor is located. Besides, the assembly must be protected from the atmospheric phenomenon.

Apart from the emissions and water quality measurements, it is crucial to know the temperature and humidity in different parts of the factory, either to compensate the gas measurements or to analyze the effect on the wireless communication quality. In this way, the router nodes will also act as sensor nodes in some cases.

In normal operation, the nodes are powered by lithium or AA batteries. However, due to the places where the nodes are located and the fact that the factory needs the information every four seconds, the power consumption becomes very high so that the batteries should be changed very often, and this is not affordable by the factory staff. Due to all of these reasons, some of the nodes are powered directly from the mains.

Taking into account the features of this scenario, it seems to be logical to think about powering some of the nodes by solar cells. Nevertheless, due to the big amount of coffee dust on the roofs, the cells would be covered needing a periodical maintenance to clean them up.

Once the peculiarities of the scenario are explained, it is important to know some information about the processes that are being carried out in the factory. The main processes involved in the coffee manufacturing are raw coffee recollection, roasting, blending, grinding, aroma extraction, drying, and binding [10]. All these processes are responsible of both gas emissions and polluted water discharges. In order to clarify the explanation, the coffee manufacturing processes are shown in Figure 1.

Even though raw coffee recollection is not one of the processes carried out on the factory, it is crucial to highlight that this first step is one of the most important when talking about the final coffee quality. Once the green grains have arrived to the factory, they have to be stored in different brown bags depending on the place of origin. This storage must be done keeping the temperature and humidity values in a very accurate range.

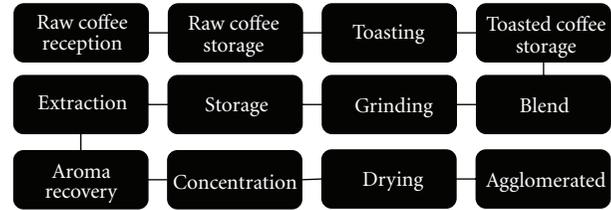


FIGURE 1: Coffee manufacturing main processes.

The first transformation process is roasting. The raw coffee is introduced in the roasters where the temperature varies from 200°C to 250°C during 200 to 600 s. After this process, the roasted coffee is weighted and the value is compared with the previous one in order to know the amount of water that has been lost.

The next step in the coffee manufacturing is blending. By mixing different types of coffee grains, the manufacturers can define what makes them different from the others. This part is only known by the company as part of their know-how so it is extremely important because it defines the final taste of the instant coffee.

Once the mixture is done, all the coffee grains are grinded together and stored in another warehouse. After being stored for a while, the next step is the aroma extraction, which can be done in a few different ways. The last three processes are aroma recovery, drying, and the binding of the coffee dust into bigger particles.

Due to all of these transformation processes, both gas emissions and water pollution must be measured to comply with regulation and to know as much as possible about the efficiency of the factory.

4. Environmental Sustainability in Food Industry

4.1. Sustainability Parameters. Today, the food and drink industry is the single largest manufacturing sector in the EU in turnover, value added, and employment terms, ahead of automobile, chemicals, and machinery industry. It is also the second leading manufacturing sector in the EU in terms of number of companies [11].

Despite the fact that food industry nowadays represents a crucial motor for the development of the economy and society, as an industrial process, the sector reports serious environmental, economic, and social impacts. Environmental degradation (water, soil, climate), competition for resources, agricultural subsidies, and trade barriers, unfair distribution of revenues to the different actors in the food production system and the integration of the primary sector into the international economy, threaten the sustainability of many food production systems.

With regards to the environmental sustainability, the food industry is a major energy user (food sector consumes about 10–15% of total energy in industrialized countries) and a major user of water (around 8% of all water used by the entire industrial sector). Moreover, the food and

drink manufacturing industry accounts for about 1.5% of total greenhouse gases emissions in the EU-15, where direct emissions account for 0.9%. This sector is also a significant source of waste generation (the food and drink manufacturing sector accounts for about 3.25% of overall waste generation in the EU) [12].

The whole production process of instant coffee is energy intensive (in the coffee factory 11.99 kW natural gas/kg product and 1.57 kW electricity/Kg product). Concerning the air emission, the outlet of both the roasting and drying processes are CO₂, SO₂, NO_x, and volatile organic compounds (VOCs). Solid wastes are mainly produced during the extraction stage, with around 40 tons of coffee wastes produced per day. Water usage is an issue in the production of instant coffee as water is used as a solvent in extraction processes and during the cleaning processes. This cleaning generates waste water containing soluble and insoluble organic material and SS [13].

For these reasons, nowadays, the idea that methods and techniques must be designed on the concept of sustainability has gained acceptance and considerations of sustainability will guide future developments in European industry and therefore in the food sector [14]. In that sense, the EU has placed increasing emphasis in the need of integrating sustainability in the sectoral policies, becoming the sustainable development, which “meets the needs of the present without compromising those of future generation,” a central objective of the European Commission. In this framework, the European Council in Lisbon (March 2000) launched the Lisbon Strategy the main pillars of which are economy, society, and environment.

For the assessment of progress towards sustainable development, several sustainability indicators have been employed since the 1992 Earth Summit recognized the important role that indicators can play in helping countries to make informed decisions concerning sustainable development. This recognition is articulated in Chapter 40 of Agenda 21 which calls on countries at the national level, as well as international, governmental, and nongovernmental organizations to develop and identify indicators of sustainable development that can provide a solid basis for decision-making at all levels [15]. In recent years, a multitude of indicators have been proposed and used in the context of sustainable development: Index of Environmental Friendliness (Statistic Finland, 2003), Eco-Indicator 99 (Pré Consultants, 2001), Well Being Index (IUCN, 2001; Prescott-Allen 2001), Living Planet Index (WWF, 2004), Internal Market Index (JRC, 2004), Composite Leading Indicators (OECD, 2002), and Dow Jones Sustainability Index (Dow Jones Indexes et al. 2005) [16].

4.2. Impact Analysis Methodology Tools. Regarding the environmental protection, the EU has a set of common rules for permitting and controlling industrial installations in the Directive 96/61/EC on Integrated Pollution Prevention and Control (IPPC) whose main objective is to secure a high level of protection of the environment taken as a whole [17]. One of the sectors where the implementation of this directive is nowadays more crucial is the agrofood sector,

since environmental impacts occur in all the processes within the food supply chain.

Nowadays, a large number of tools for the assessment of environmental impacts are available, such as life cycle assessment (LCA), material flow analysis (MFA), environmental impact assessment (EIA), and system of economic, environmental auditing and environmental accounting [18]. LCA is the most widely tool employed for the quantification of the environmental impacts nowadays.

LCA is a “compilation and evaluation of the inputs and outputs and the potential environmental impacts of a product throughout its life cycle” [19]. Therefore, LCA as analytical tool allows the assessment of the environmental impacts and resources employed throughout the whole life cycle of a product. The International Organization for Standardization (ISO) in the ISO 14040 series has standardized a framework for LCA.

LCA, while increasingly accepted and more facile to perform due the databases and software systems, faces some important challenges over the coming decades including evaluations and comparisons of the results obtained through different methodology’s variants, uncertainty and data quality, LCA sophistication. Methods, operational procedures, and concepts for the implementation into business processes need more attention and research, in order to enable the wide-scale exploitation of LCA’s potential [20].

For about 20 years now, the Life Cycle Assessment (LCA) method has successfully been used to analyze agricultural production systems and food chains. Although progress in terms of methodological robustness and data availability and reliability has been widely demonstrated [21], the application of LCA methodology to the food and drink sector has several weaknesses. An effort must be done to explore the suitable functional units, system boundaries and allocation procedures for LCA in food production to facilitate a valid comparison between different products [22].

4.3. Dynamic System for Sustainable Transformation Processes.

According to the Reference Document on Best Available Techniques in the Food, Drink and Milk Industries of the European Commission, (August 2006), environmental management tools, identified as a general best available technique for the food and drink sector, require checking systems in place to evaluate the effectiveness of measures implemented to minimize consumption and emission levels and to monitor and review their effects periodically. By ongoing monitoring, the effectiveness of the chosen measure can be periodically checked to see whether it is meeting the set targets, for example, consumption and emission performance levels. Underperformances can thereby be detected and rectified. Also, monitoring shows trends and can identify priority areas for improvement.

The main environmental impact categories that are nowadays monitored in the food sector are natural resources depletion, global warming, ecological toxicity, solid waste, embodied energy, acidification, human toxicity, smog formation, indoor air quality eutrophication and ozone depletion. the following indicators are normally measured to monitor the impact categories previously mentioned.

- (i) Resources Consumption, The main resources consumed are water, electricity, and fossil fuels. Their measurement and control by the company tends to be simple because industries typically acquire the consumable resources from other companies and transaction data are documented and recorded.
- (ii) Wastewater An adequate monitoring of the wastewater can control both the maximum concentration values of chemical parameters such as the quantification of the quantities discharged. The measurement and control of physical and chemical parameters of the wastewater is often achieved by direct methods. In the case of pH and EC, there are standardized publications according to the UNE standards related to the implementation of monitoring tools extent of these two parameters, namely, UNE 77078 : 2002 [23] and UNE 77079 : 2002 [24]. The rest of parameters are usually controlled in batch, taking a representative sample of water and analyzed later insitu using an appropriate kit or in the laboratory.
- (iii) Air emissions. The analysis of the combustion gases is usually done by measuring systems in situ, on a continuous or discontinuous way. These equipments may be mobile and analyze different parameters at the same time (O₂, CO₂, draft of chimney, CO, NO, SO₂, etc.). There are several standard procedures UNE-related sampling in continuous and more specifically with the measurement of flow rates, as the UNE 77227 : 2001 [25]. There is also a standard that establishes standard procedures for the measurement of the characteristics of the flow of gases: UNE 77225 : 2000 [26].
- (iv) Subproducts/wastes. The main wastes generated by the food industry are organic waste, waste oils, batteries, laboratory waste, solvents, glass, plastics, paper, carton, and so on.

Several approaches have been made for the development of technologies and methodologies for the monitoring of emissions and consumption levels of energy, water, and wastes throughout the whole supply chain as follows.

- (i) In case of energy consumption, data are normally obtained from the monthly utility bills which are irregular and cannot quantify any on-site energy consumption, so researchers in this field are focused nowadays in the development of real-time web-based monitoring systems including a data logger that monitors sensors and sorts their values.
- (ii) Regarding environmental impacts generated by wastes, researchers are mainly focused in monitoring wastewater quality. Sampling and laboratory analysis are not well adapted to wastewater quality monitoring in a process control or hazards prevention context, for which on-line/on-site measurements are preferable [27]. Some of the rapid toxicity tests available today require certain conditions to function properly or their results do not always correlate with other methods.

- (iii) Systems and methods for monitoring and controlling fluid consumption in a fluid-supply system are disclosed using one or more sensors for generating signals indicative of the operation thereof. Systems and methods herein involve one or more sensors in a fluid-based system for generating signals indicative of the operation thereof.

As a summary, existing systems for the monitoring and reporting of environmental parameters are based in the collection of data obtained from irregular sources such as monthly utility bills or in robust and dependable sensors. Therefore, those systems are irregular, cannot quantify any on-site parameter, and do not assure continuous functionality and data transmission.

5. Cookies Platform Description

The Cookies WSN platform [1] consists in four different layers that can be changed depending on the application making the modularity its best feature. Each of these layers matches with a different functionality: power supply, communication, sensing/actuating, and processing.

5.1. General. As it has been said before, a Cookie is a HW platform for WSNs. Every Cookie is a node of the network and is composed of four main layers. This node has been used already in a security application, which can be seen in [28].

Depending on the requirements of the application, the platform can be adapted by changing these layers. In this way, it is very easy to obtain an optimum design in a very short time. It is not the purpose of the authors to deeply describe the Cookies platform, as they were analyzed in other works like [1] and [29].

In the application studied in the present work, different modifications have been carried out, demonstrating the advantages of such a solution, specifically in sensing and power supply layers. These developments are detailed in the following subsections.

In Table 1, a comparison between different WSN platforms can be seen.

The platform Cookies is based on an 8051 uC from Analog Devices and a Spartan 3 FPGA from Xilinx, which make it one of the most powerful platforms in the state of the art. Moreover, its modularity allows to use it in prototyping stage and to proof different concepts before deployment in a short time. Due to its modularity, only part of the platform was redesigned to make it suitable for the application presented in this work.

5.2. Hardware Developed for This Application. In order to comply with the requirements imposed by this specific application, it is necessary to design two new layers of the platform. First of all, a sensor layer needs to be adapted to face the water quality measurements, and, on the other hand, another sensor layer has to be developed to measure the gas emissions. As it was mentioned before, these new measurements are

TABLE 1: Comparison between WSN platforms.

Platform name	Marketed	No. of Layers	Processing	Communications
TelosB (Berkley)	YES	2	MSP430F1611 16 bit	CC2420 IEEE 802.15.4
Intel iMote 2	YES	2	XScale PXA271 32 bit	CC2420 IEEE 802.15.4
Sun SPOT	YES	2	ARM920T 32 bit	CC2420 IEEE 802.15.4
Libelium Wasp mote	YES	3	ATMega1281 8bit	XBee module, ZigBee compliant
Platform MIT	NO	4	C8051F206 8 bit	TDMA protocol
mPlatform Microsoft	NO	Not specified (>4)	2 processors in each layer and 1 CPLD XC2C512 CoolRunner	CC2420 IEEE 802.15.4
Hitachi ZN1	NO	3	H8S/2218 16 bit	CC2420 IEEE 802.15.4
Cookies	NO	4	Analog Devices ADuC841/MSP430 and Spartan-3 FPGA/Igloo	ZigBee Telegesis and Meshnetics modules (2.4 GHz and 868/916.5 MHz) and Bluetooth

- (i) pH and temperature of the waste water,
- (ii) gas emissions: CO, SO₂, NO.

In this application, even though the nodes are normally powered by batteries, it will be necessary to mix the batteries with nodes powered directly from the mains, since it is required that the measurements are taken every four seconds and the power consumption is quite high (about 70 mA).

5.2.1. Water Quality Measurements (pH and Temperature).

The pH value expresses the proton concentration in a water solution given by the drop voltage between two electrodes. This drop voltage appears when two different liquids with two different pH values are separated by a crystal membrane. Due to this reason, the pH sensor is made of two different electrodes: a reference electrode immersed in a known solution and the measurement electrode in contact with the liquid sample.

In order to obtain the pH value it is necessary to measure the drop voltage between both electrodes but it is crucial to take into account that this drop voltage depends on the temperature of the liquid following the expression shown below:

$$\text{pH}(x) = \text{pH}(S) + \frac{(E_s - E_x) * F}{R * T * \ln(10)}, \quad (1)$$

where $\text{pH}(x)$ is the target value, $\text{pH}(S)$ is the pH value, of the reference solution ($\text{pH} = 7$), $(E_s - E_x)$ is the drop voltage, F is the Faraday constant ($F = 9.6485309 \cdot 10^4 \text{ C} \cdot \text{mol}^{-1}$), R is the ideal gas constant ($R = 8.314510 \text{ J K}^{-1} \text{ mol}^{-1}$), and T is the temperature of the liquid sample. In this way, the sensor must be capable of measuring both the drop voltage between the electrodes, and the temperature of the liquid.

The sensor chosen is the *InPro 4260/120* manufactured by Mettler. It is a passive sensor since it does not need to be powered externally. The voltage response can be either positive or negative, so that it is considered as a bipolar sensor. Another important feature of this kind of sensors is

the huge impedance that can reach values of 10 M Ω or even 1000 M Ω . Due to this fact, it will be necessary to measure the voltage response through a very high impedance operational amplifier. The sensor includes a resistance temperature detector (RTD hereafter) *Pt100* to measure the temperature and to compensate the value of the drop voltage.

Since this is an analog sensor, the response signal needs to be adapted to the voltage range understood by the ADC (integrated in the microcontroller of processing layer) which is 0 V–2.5 V. As it was said before, the response signal can be either positive or negative which obliges to have negative power supply in the conditioning circuit. The power supply of the platform is given by the power supply layer, but it is only positive (1.2 V, 2.5 V, 3.3 V), so, in order to be able to power the circuit with a symmetric voltage of –3.3 V to 3.3 V, it was necessary to set up a charge pump.

The sensor output consists of seven different wires: crystal electrode, reference electrode, ground, cable shield and three cables for the pt100 sensor.

Both electrodes have to be connected to a voltage follower made by a very precise operational amplifier due to the very high impedance of the sensor. After that, the drop voltage between them is adapted to the desired range by an amplifier stage with variable gain and offset.

At the same time, the RTD response has to be measured. Since it is a three-wire RTD connection, the measurement circuit is very simple (Wheatstone bridge). Besides, there is a cable shield to protect the signal against noise and, of course, the circuit ground.

The voltage response of the bridge will be also amplified and corrected with an offset in order to adapt it to the ADC range. In Figure 2 the PCB for pH and water temperature measurements is shown.

Once the circuit is designed and validated, the next step is calibration. First of all, it is necessary to calibrate the temperature measurement. To do so, a buffer solution with a known pH value was introduced in a special chamber (in the *Laboratorio Central Oficial de Electrotecnia*, LCOE), increasing the temperature and measuring the sensor response.



FIGURE 2: Cookie node with pH and temperature layer.

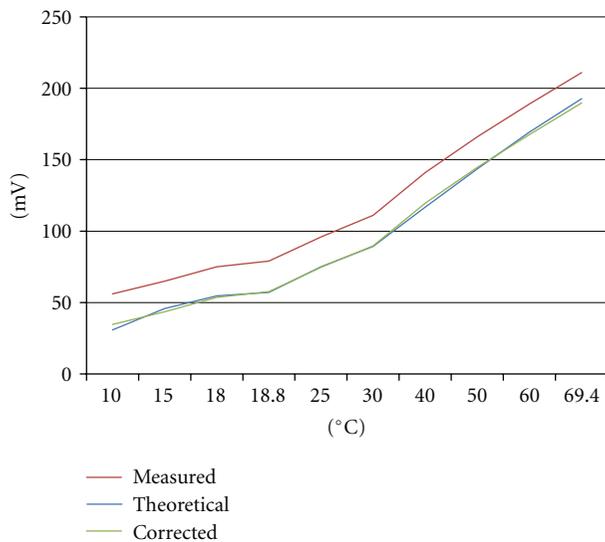


FIGURE 3: Temperature calibration curve.

After that, the theoretical values were compared with the experimental results and the measurement was corrected as shown in Figure 3.

After temperature calibration, the sensor was immersed in different buffer solutions with known temperature in order to obtain similar curves and to be able to make the correction of the values (Figure 4).

The admissible pH values in the sewage plant can vary from 5 to 8, 6 being the optimum value. The temperature will oscillate between 30°C and 40°C.

5.2.2. Air Quality Measurements

Gas Emissions. The gases that are to be measured in the factory are

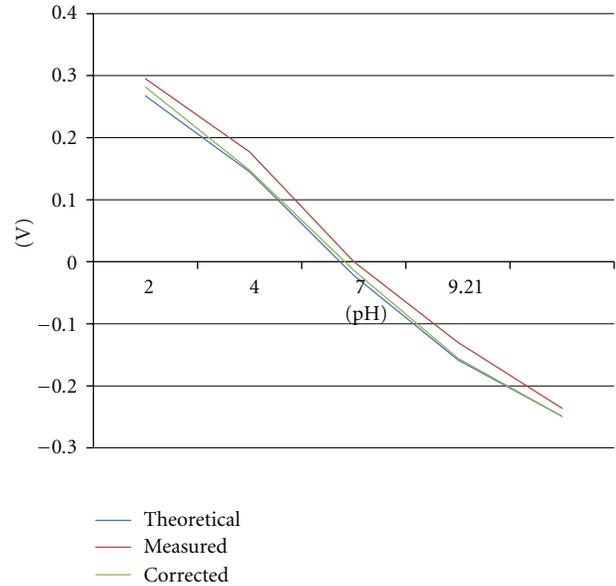


FIGURE 4: pH calibration curves.

- (i) carbon monoxide (CO),
- (ii) sulfur dioxide (SO₂),
- (iii) nitric oxide (NO).

Carbon monoxide is a poisonous, colorless, odorless, insipid, and very toxic gas. It is usually emitted because of incomplete combustion processes, and it is very dangerous since it replaces the oxygen in the hemoglobin causing what is commonly called the “sweet death.” The limit value to comply with regulation in the factory is 500 ppm.

Sulfur dioxide is a colorless, irritant, toxic gas with a very particular suffocating smell. It is the main producer of the acid rain because it is transformed into sulfuric acid in the atmosphere. The maximum value to comply with regulation is 4300 mg/m³N.

Nitric oxide is also an irritant and toxic gas produced during combustion processes. Normally it can be very dangerous when oxidized into nitric dioxide.

The sensors that have been chosen for these measurements are manufactured by a British company called *Alphasense*:

- (i) carbon monoxide (CO-BF),
- (ii) sulfur dioxide (SO₂),
- (iii) nitric oxide (NO).

All of them are electrochemical sensors. Those sensors consist in three different electrodes with a thin screen of electrolyte between them.

The first one, the working electrode or sensing electrode, is the one in contact with the outside gas, so that it is the real surface where the reduction or the oxidation occurs. In this way, a current which is linearly proportional to the volume of the toxic gas is generated. The second electrode is called the counter electrode, and it is in charge of balancing

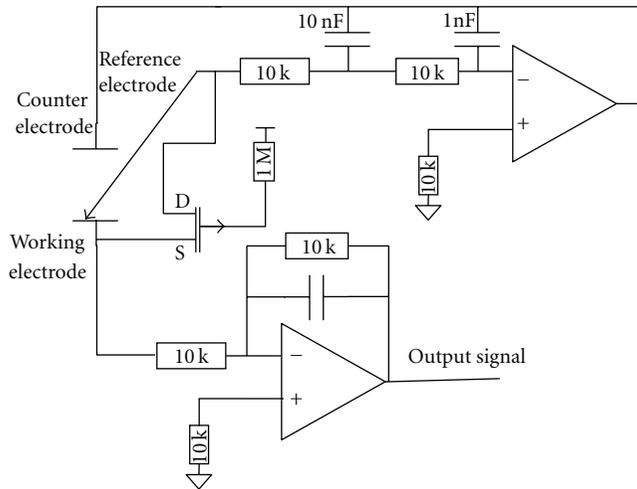


FIGURE 5: Potentiostatic circuit.

the reactions on the working electrode generating the same current but in the opposite sense.

The last one is the reference electrode. This one is to ensure that the working electrode is always working in the correct region of the current-voltage curve. By keeping the potential of the working electrode as stable as possible, the spoilage of the sensor can be dramatically reduced when at the same time both linearity and sensitivity are increased.

Those are also analog sensors, so that it is also necessary to adapt the signal to make it fit into the ADC voltage range. The typical conditioning circuit used for this kind of sensors is known as potentiostatic circuit, shown in Figure 5.

This potentiostatic circuit can be divided in three different parts.

- (i) Control. This is the upper part of the circuit. Since it is the part connected both to the reference electrode and to the counter electrode, it will be in charge of giving the necessary current to maintain the equilibrium between the working electrode and the reference one. In this way, when the circuit is turned on, the FET is set in high impedance, and both electrodes are fixed at the same voltage. In order to be able to maintain good linearity and sensitivity, the amplifier must be a very precise operational amplifier with a very low value of bias current.
- (ii) Current Measurement. This corresponds with the lower part of the circuit. When the working electrode is exposed to the toxic gas, an oxidation reaction occurs creating a current response. This current will be converted into a voltage response which is proportional to the toxic gas concentration in the transimpedance amplifier.
- (iii) FET. The transistor works just to avoid the polarization of the sensor when the circuit is not powered. When the circuit is turned off, the JFET creates a shortcut between the working and the reference electrodes to guarantee that they have the same

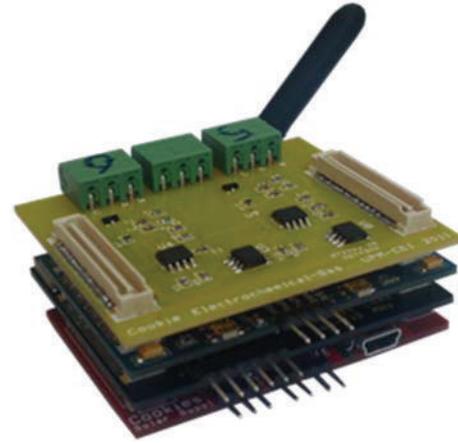


FIGURE 6: Cookie node with the electrochemical sensor board.

voltage. In Figure 6, a node with the potentiostatic circuit can be seen.

Once the circuit board is finished, the sensors must be calibrated. The calibration process consists in doing two different tests to find two different working points. Two points will be enough since the sensor response is linear.

The goal of the first test is to find the *zero* value that corresponds with the sensor response when there is no target gas. This test was done in the lab, considering that the gas concentration is negligible compared with the target values. On the other hand, the second test was done to find the sensor response using a gas bottle with a known concentration value. With the information given by these two points, the response line can be drawn and compared with expected one in order to be corrected by software. Another important issue is how the temperature affects the measurements. Due to this fact, it is crucial to measure the temperature and to correct the value according to the curves given by the manufacturer.

In order to clarify the process, a test example is shown in Figure 7.

According to the response curve shown in Figure 8, the voltage value takes about 2 minutes to be stable after the circuit is turned on. It is important to notice that the older the sensor is the slower the response becomes. With a new sensor, the response time should be much lower, but it also depends on the volume and test conditions. As it can be seen, the voltage value for 300 ppm is 2 V, which corresponds with the expected value.

Air Temperature and Humidity. The effect of the air temperature and humidity on wireless communications can be very notable. As it was explained above, the Cookies communication modules are based on the ZigBee protocol which is highly affected by these parameters. By monitoring both temperature and humidity in some points of the deployment, it is possible to compare how the communication quality indicators such as the link quality indicator (LQI) and the receive signal strength indicator (RSSI), varies over the changes in the environment. Besides, the gas temperature on



FIGURE 7: Gas calibration test conditions. Target gas: carbon monoxide. Sensor: CO-BF (Alphasense). Concentration: 300 ppm. Volume: 5 mL/min. Temperature: 25°C.

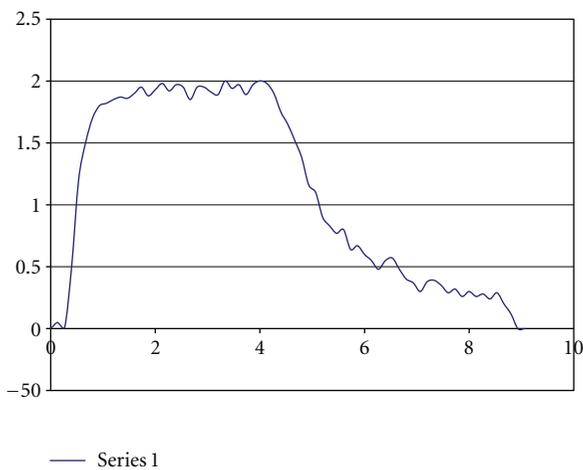


FIGURE 8: CO Sensor response: Test 1.

the chimneys is necessary to compensate the measurements taken by the gas sensors. The sensor used to measure both air temperature and humidity is the SHT11 manufactured by *Sensirion* (included in the Cookie platform; see Figure 9). This is a digital, low-power, fully calibrated sensor so it is very convenient for this WSN application. The output signal is processed by the FPGA in the processing layer. Since it is a prototype version of the platform, the processing layer includes an FPGA which is not the best solution in terms of power consumption, at least in the current platform.

5.3. Software Description. The application software is composed of two different groups. A WSN distributed application, running on the nodes of the network, and a management application, running on a PC. Working as an interface between the network and the PC, a special kind of node called sink acts as a gateway. This node processes and translates all the information sent by the sensor nodes to make it compatible with the application running on the computer. Figure 10 shows the basic structure of the application.

The WSN distributed application includes three different elements: sensor nodes, router nodes, and the sink node.



FIGURE 9: Sensor node. SHT11 based.

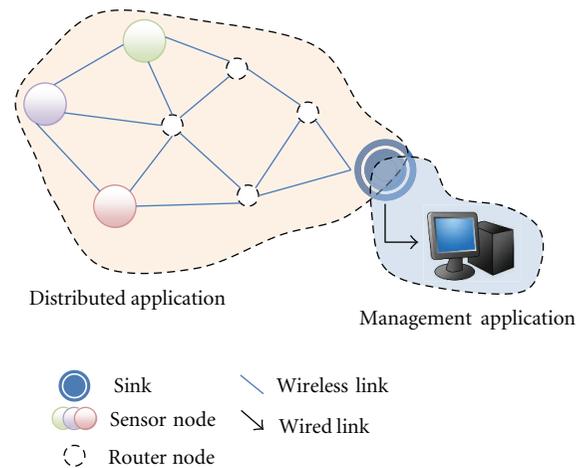


FIGURE 10: Application structure.

As it was reported before, the sink node processes and translates the information received from the network. This means not only to transform the binary information received to their corresponding physical magnitude but also to process the messages related to network management and sending the relevant information to the management application. Every time the sink sends a message to the management application, it also attaches a timestamp. This timestamp refers to the start time of the network which is different from the GMT time.

Apart from the processing tasks, the sink node acts like the coordinator of the network. This implies that the application of the sink node needs to handle the startup of the network and in case of a network collapse it has to create a new one.

On the other hand, the router nodes send the received messages to the sink node. This is done on the network level so that the application does not have to handle the routing of the messages.

TABLE 2: Application message.

Header (8 bits)			Data		
Type	Descriptor	Data length	Data 1	...	Data N
x x	x x x	x x x	16 bits	...	16 bits

Sensor nodes application main tasks are to control the activation of the processing unit, reading from the analog and/or digital sensors and, once all the measurements have been made, sending the data to the coordinator. Another task will be processing all the messages received from the communication layer while the node is active.

Router and sensor nodes application share some common tasks on the commissioning side. These are controlling the status of the node, monitoring batteries, warranting the presence of the node in the network and notifying the node type to the sink node.

In order to accomplish with all the tasks, a structured application message mechanism has been designed. This mechanism is based on 4 basic messages.

- (i) Commissioning messages in order to keep trace of the quality of the network, battery life, and so forth.
- (ii) Error message used to communicate to the coordinator any unexpected situation, that is, malfunctions in sensors.
- (iii) Node identification. These types of messages are used by the monitoring application to assign the appropriate indicator in the visualization interface. This message is sent when the node joins the network. It only contains the sensor node descriptor.
- (iv) Sensor measurement. These messages contain the measure values of all the sensor of a unique node, and no separate values for different sensor of the same node can be sent; that is if a node has sensors of pH and temperature a message has to contain values for both magnitudes, ph and temperature cannot be separate in two different sensor measurement messages.

In order to allow a simple decoding mechanism, a 1 byte application message header is used in all the messages. The structure of the header is shown in Table 2.

These different fields are explained below.

- (i) Type. This section indicates the message type; it corresponds to one of the four messages described previously.
- (ii) Descriptor. This field informs about the data type contained in the message. In the case of messages containing measurements, it contains the type of the node. This attribute can change its meaning depending on the type of nodes used in each application. For commissioning messages, it indicates what information is inside, communications quality, battery life, and so forth.
- (iii) Data length. This is the number of data packets included in the message. Every data packet is 2 bytes

length. The data is sent in binary format, without conversion to native units (Celsius degrees, pH, etc.). This conversion will be done in the sink node before sending the messages to the management application.

- (iv) Data. This is the information given by the sensors, such as communications quality, battery life, and so forth.

In the application presented on this document, five different types of nodes are used, including router nodes and four different types of sensor nodes (pH and water temperature, NO SO₂, and CO, and ambient humidity and temperature). This implies the use of five different descriptors.

The WSN distributed application works as follows.

- (i) The Sink node creates the network.
- (ii) Sensor nodes and routers join the network, and then they send an identification message to the sink.
- (iii) Periodically (depending on the sensor type), sensor nodes activate their processing unit, read the sensor data, and send the acquired values to the sink. Once this process has finished, the processing unit goes back to standby mode.
- (iv) Periodically, router and sensor nodes send commissioning messages to the sink node.
- (v) When a message is received in the sink node, it processes the information, attaches a timestamp, and sends the result to the management application.

The management application is used as a visual interface between the user and the sensor network. It is also used as a data logger to create log files with the data sent by the sink.

Figure 11 shows the interface of the management tool. The main characteristics of the interface will be presented below.

On the top left part, the configuration interface for the serial port is placed, which communicates the management application with the sink node.

The left-side bottom includes the network status table. This status table shows the information about the nodes that have joined the network, the node identifier (node ID), type (translation of the binary descriptors to its matching names), timestamp of the last message received from the node, location of the node (it has to be filled by the user; cookie nodes do not have geolocation capabilities yet), and the status. The status is used to shown if the node still connected to the network (status OK) or if the node is disconnected, that is, by a temporary fail of communications, or completely out of batteries, which will be shown as status ERROR.

On the right side, the indicators of each different node used in the application are placed; when an identification message arrives to the sink, it extracts the node identification from the message and the descriptor enclosed and sends both to the management application in order to allow it to assign an appropriate indicator depending on the sensor type. Each indicator has only a node attached; in this way all the data

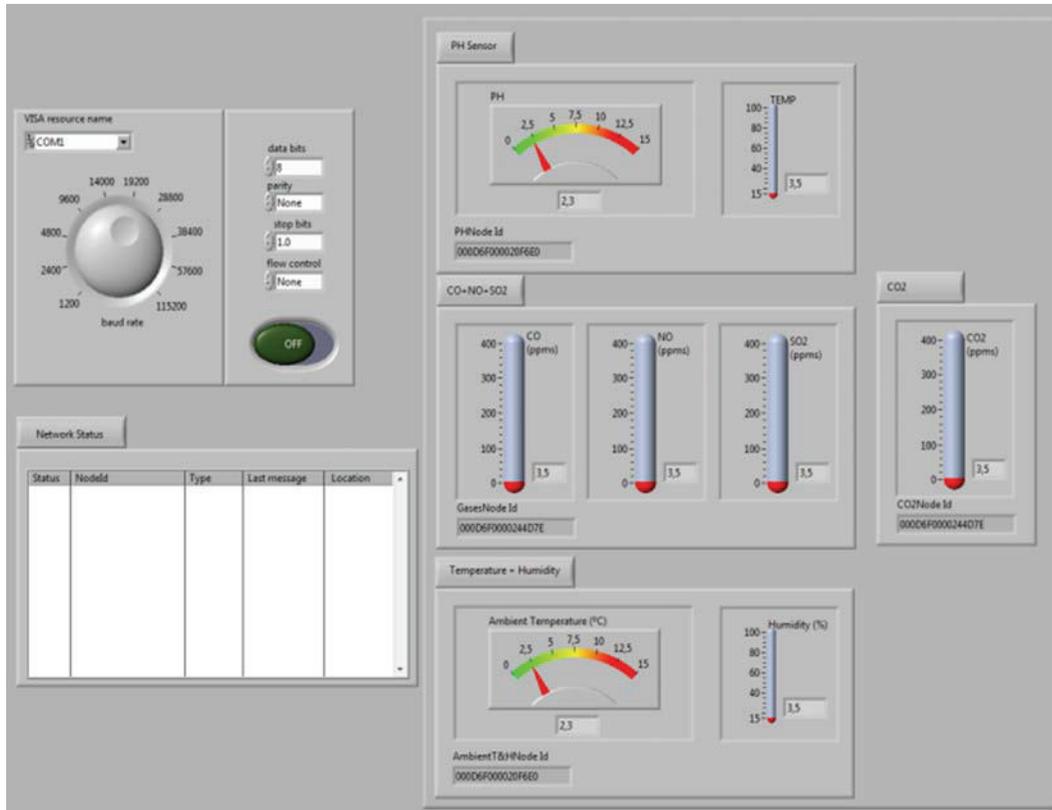


FIGURE 11: Application interface.

received from a unique node will be shown in the same display.

As it was explained before, the management application has another important task, storing all the information sent by the sink to the computer. There are mainly two different groups of messages, the ones related to sensor measurements and those related to the network commissioning, so the application stores the information in two separate folders depending on which group the message belongs to. The application handles the creation of new files for both types of messages with a time interval set by the user.

To conclude, the main features of the software application are summarized as follows.

- (1) There are two groups in the software side, the application running in the nodes of the network and the management application running on the computer.
- (2) The distributed application main task is to measure environment parameters, and not only sending the information to the sink but also keeping the communication structure alive.
- (3) The sink centralizes the information processing and sends the data to the management application.
- (4) The management application shows the information about the status of the network and the instant measurements of the sensor nodes in a visual interface, but it also stores all the messages received separating

the data related to sensor measurements and the data related to the commissioning of the network in different files.

6. Deployment, Measurements, and Results

In this section, the results of the measurements taken in the factory are detailed. In order to test different parameters of the WSN platform, the deployment was carried out in two different tests.

The first experiment was done to cover the perimeter of the factory using the fewer amount of nodes possible. On the contrary, the second test was a complete deployment where both air quality and waste water quality were measured while all the data was sent to a sink node placed on the factory offices.

6.1. Test 1. In this first test, 4 nodes were deployed as seen in Figure 12. The features of each one of these nodes are

- (1) sensor node, measuring pH, temperature of the waste water, and power consumption,
- (2) sensor/router node, measuring air temperature, humidity, power consumption, RSSI, and LQI to study the quality of the communications,
- (3) sensor/router node, measuring air temperature, humidity, power consumption, RSSI, and LQI to study the quality of the communications,

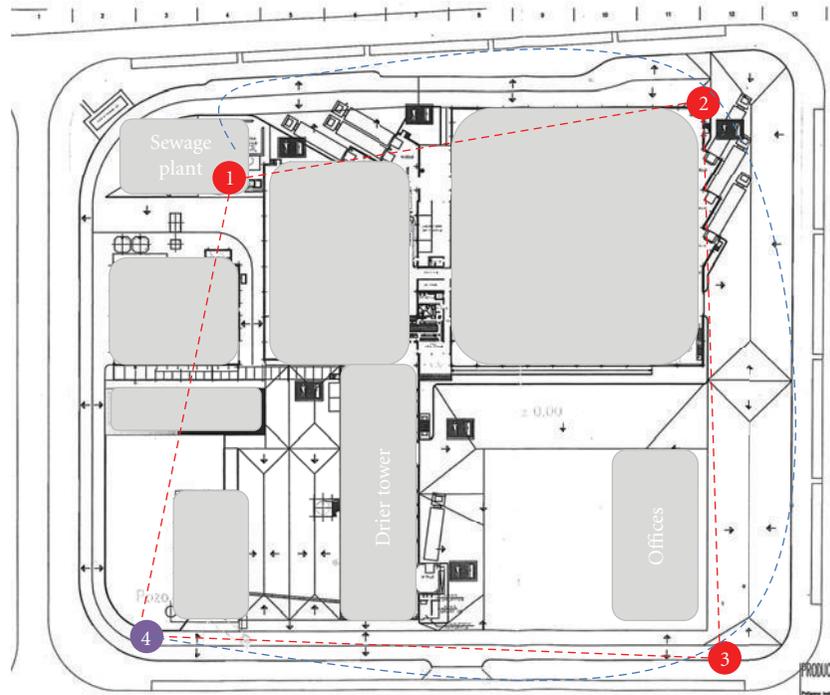


FIGURE 12: Test 1 in the factory.

- (4) sink node to harvest all the information from the rest of the nodes.

During this test, node no. 1 was left on the sewage plant as the sink node was getting further following the path shown in Figure 12. Once the sink node reached the north-east corner of the factory, communication was lost due to signal attenuation caused by the buildings and trucks in this side of the factory. Then, node no. 2 was left on this corner acting as a router node between the one on the treatment plant and the sink node (the distance between node no.1 and node no. 2 is 128 m). The sink node was then carried until next corner where the communication was lost again due to interferences (distance between node no. 2 and node no. 3 is 160 m). Finally, node n no. 3 was left on the next corner while the sink node was carried to the last one as seen in Figure 12.

In Figure 13, some measurement results are shown. The pH value oscillates from 5.5 to 6 pH units while the temperature of the water varies from 30°C to 32°C. The power consumption changes depending on whether the node is transmitting or receiving data. It is important to take into account that 4 nodes are enough for covering the perimeter of the factory because there are only a few sources of signal attenuation between them. When the path has to cross the factory, there are a lot of buildings and metallic tanks where the signal can be lost so that a bigger amount of nodes is necessary.

In Figure 14 and Figure 15, the results for the RSSI value can be seen. In Figure 14, the received signal strength

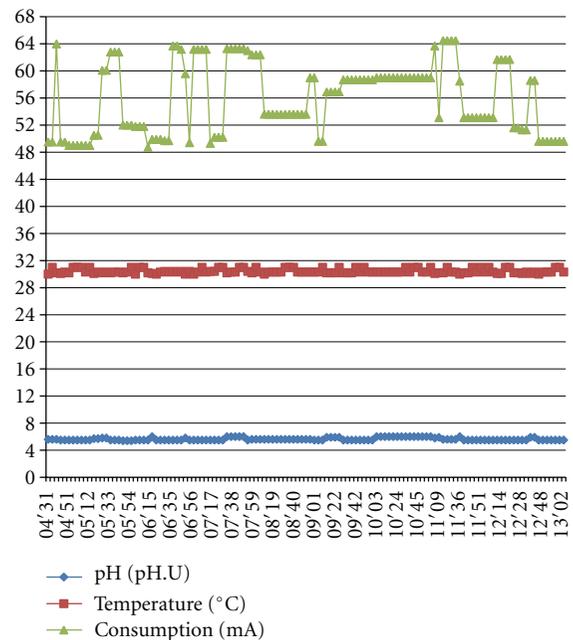


FIGURE 13: Measurement results. pH node. Test 1.

indication tends to decrease due to the interferences caused by the trucks that were working while the test was done. In the case of Figure 15, the interferences were only caused by one or two trucks passing by, because there was not a lot of activity in this place in this specific moment.

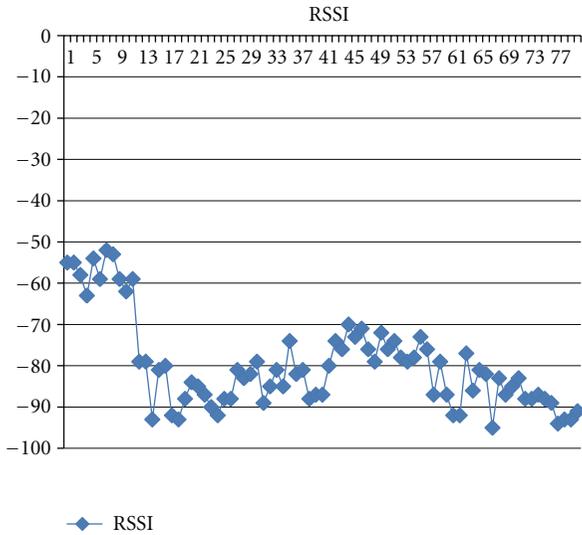


FIGURE 14: RSSI values between nodes no. 2 and no. 3.

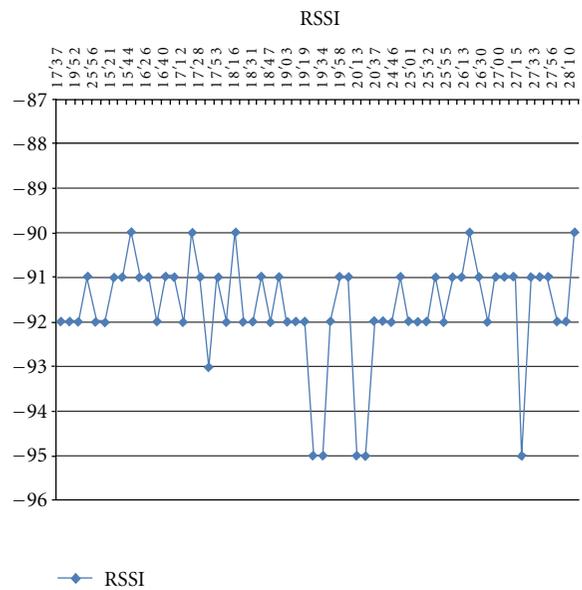


FIGURE 15: RSSI values between nodes no. 1 and no. 2.

Another important question is that, even though the node no. 3 was placed in the south-east corner, at the end of the experiment the node no. 2 was directly connected to the sink node. In this way it is possible to notice that the range of the nodes can be quite high even facing the attenuation caused by buildings and the factory machines. Nevertheless, this connection was not very reliable so it was considered as a better option to include an alternative path for the cases where the interferences were bigger.

6.2. Test 2. The second experiment consisted of 5 nodes (Figure 17), with the main goal of monitoring water quality, emissions on the drier chimney (Figure 16), air temperature, humidity, and the communication quality in different places



FIGURE 16: Drier chimney.

of the factory. As it was explained before, some of the nodes were to be connected directly to the mains. In this case, only nodes no. 1 and no. 2 were connected in this way.

The sink node was placed in the office buildings in order to collect all the data harvested by the rest of the nodes. It is important to highlight that, once the node no. 1 and node no. 2 were deployed, when node no. 3 was going to be placed on the roof, the communication between the sink node and node no. 2 was lost. This happened because of the attenuation inside the building. Once the node was on the roof, it rejoined the network automatically.

In Figure 18, the measurement results on the drier chimney are shown. Even though the experiment time was quite short, due to the temperature of the gas, it is possible to see the time necessary for the sensor to give a stable measurement after being powered. The experiment time could not be longer because the system needed to cool down the gas is not installed yet. This time is, in this case, 3 min. although it can be reduced using a new sensor with a nonworn membrane. It is also possible to notice that after 5 minutes the gas concentration started to decrease as the smoke of the chimney did the same.

Another important aspect that it is necessary to take into account is the sensor fouling (Figure 19). After being immersed inside the waste water in the output of the sewage plant, the membrane of the pH sensor was covered by coffee grounds. This caused a drift on the sensor measurement so the membrane needed to be cleaned up. After having the sensor immersed in the waste water for an hour, the membrane was partially covered so the measurements were not trustworthy. The way to solve that is setting up a shield around the sensor to filter the floating matter, but this is still future work. In this first deployment, the sensor was cleaned manually after one hour, since the drifts at this time were more than 40%. The correction of this kind of drifts caused by the dirtiness is quite difficult since the sensor fouling is hard to foresee. The only solution seems to be preventing the sensor from the dirtiness by using the shield mentioned above.

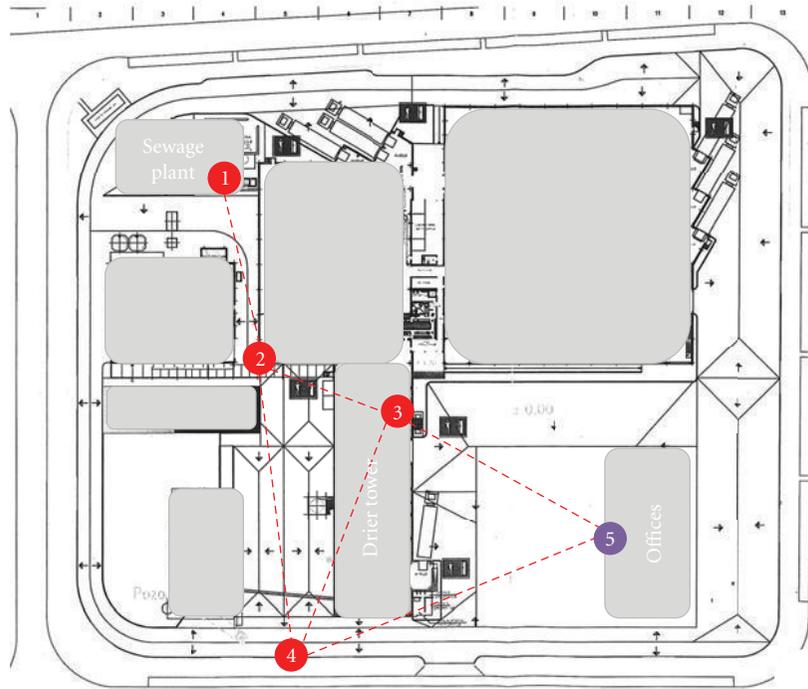


FIGURE 17: Test 2 in the factory.

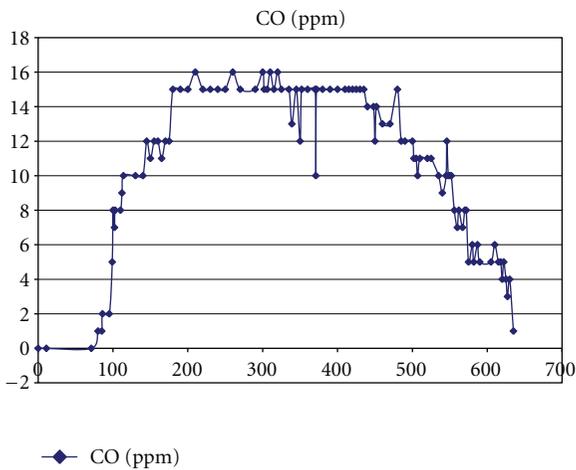


FIGURE 18: CO measurement in the drier tower chimney.



FIGURE 19: Sewage plant. pH sensor.

6.3. *Challenges of the Deployment.* When facing this kind of real applications, a lot of challenges appear along the way. First of all, it is fair to say that it is the first time the *Cookies* have to deal with this kind of environmental measurements. At the beginning, the adaptation of these chemical measurements was really tough, since the nature of the sensors was completely unknown for the team. Due to this reason, a big effort in order to understand how the sensors worked was necessary. Another challenge related to these sensors, is that the majority of the pH ones always include their own signal conditioning systems which, of course, are not compliant with the *Cookies* platform. The documents and guides are not prepared for using only the sensor so it was difficult to adapt the signal given by the sensor to the platform needs.

A factory is not an ideal scenario. What can be really easy in a laboratory can become a real problem when deploying the network in such an unfriendly environment. In a factory like this one, the production line cannot be stopped. This fact can be really uncomfortable when trying to set the nodes in the test points.

An example of how difficult the scenario can become is the signal attenuation caused by the metallic objects all around the place including the trucks in charge of delivering and rubbish recollection in the factory. In the initial planning, most of the router nodes were placed inside the buildings. Nevertheless, the team realized that placing the nodes outside the coverage was improved, so fewer devices were needed. Even with these trucks moving around the nodes, the attenuation was important but not enough for

interrupting the communication. In the rare cases where the communication was lost, the nodes rejoined the network in a few seconds so no important information was missed.

The best way to solve this situation is to characterize the deployment environment before the network is set up. This is one of the biggest challenges today in WSNs, and several works are being carried out by the scientific community to find a suitable solution, in terms of planning, simulation, and maintenance tools for WSNs.

Temperature conditions on the test points were also a big deal when trying to organize the deployment. In the case of the gas emissions, the temperature inside the chimneys was really high (around 100°C), so the sensors could not work without cooling the air first. That is why the results were taken only during a short period of time. This problem needs still to be solved because the platform to cool the air down is being studied by the factory.

Other important problem was related to power consumption, since the measurements had to be taken every three minutes. Taking into account that the normal way to power the nodes is using either AA or Lithium batteries, with such a big amount of data transfer, the batteries would need to be changed every two days. Due to this reason, the nodes with the biggest power consumption were directly connected to the mains.

Regarding the web tool developed for the data storage and evaluation, it was necessary to install some programs in the PC in charge of receiving the data in the factory. The main purpose of this software is sending the information automatically from this PC to the Inkoa server where the data is uploaded to be used by the web tool. In order to do so, some ports of the private network of the factory had to be opened. This was a problem, since it opened a path for possible illegal users.

7. Closing the Loop

The data obtained from the WSN will serve as input to a software application (Figure 20) whose main functionality is the development of environmental impact studies at unitary process and company level in vegetable processing facilities. The software application will enable the customer the assessment of the environmental sustainability of their production processes under a life cycle approach and the identification of processes' impacts on the environment. The software application is structured into 2 different modules.

- (i) *Monitoring module* to compile and record data about the end users' consumption and emission levels and that will be supported by the WSN enabling the measurement of specific parameters.
- (ii) *Assessment module* to perform the life cycle assessment of a product based on ISO 14040 and 14044 methodology and enabling the identification of environmentally unfriendly production processes as well as prevention and minimization options. For the life cycle impact assessment, the software includes the method "Eco-Indicator 99" that enables the assessment of the damage caused by a product system



1. Gráfica de medidas del sensor mA del nodo EDAR

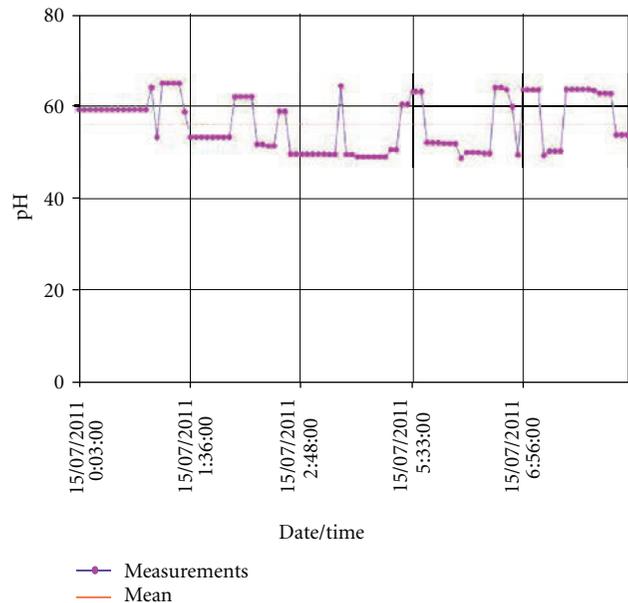


FIGURE 20: Application preview.

to human's health and the ecosystem quality and resources.

To allow communication between the sensor network and the software platform, it is necessary to dispose of a dedicated application invisible to the user, which runs through a scheduled daily task. At the time the application is running, it accesses to the monitoring files, reads them one by one, and connects to the database software application to store them in a table. In this table, for each record,

the application saves the date and time of measurement, the sensor, the node, and the measurement made.

For the development of the software application, a web server has been employed, performing bidirectional and/or unidirectional and synchronous or asynchronous connections with the client side. The services offered by this web server are FTP, SMTP, NNTP, and HTTP/HTTPS. The web pages that collect the software application have been published in the same computer where the web server has been set up. The web pages are accessible both local and remotely. The server services have provided the tools and features necessary to easily manage the web server.

For the database management, a system based on entity-relationship model has been used. Its languages for query are T-SQL and ANSI SQL. The version of the database used includes increased security, integration with power-shell, transparent data encryption, data auditing, data compression, and reviewer of Transact-SQL and IntelliSense languages. It also includes new data and functions types (spatial data, new data on time (date time 2 and date time offset), types of hierarchical data, etc.).

8. Conclusion and Lessons Learned

This paper has shown a real application of a WSN. Many contributions are presented about the use of a custom platform, such as the adaptation to a heterogeneous network, the study of an interesting industrial field which is applicable worldwide in many installations, the integration of management and control tools with WSNs, and so forth.

This paper may serve as a starting point for many replicas of environmental control applications. A closed loop has been presented where three different partners have joined efforts to achieve a complete monitoring system based on a nonintrusive and unattended technology. Thanks both to the chemical nature of the measurements and to this very specific industrial application, this project has been very didactic for the team since it was necessary to learn about very different topics to face all the requirements. Apart from that, the application has served as well in the development of the Cookies platform which is now capable of facing very demanding applications in terms of reliability and adaptation to the measurement of new parameters. Besides, deploying the network in a very hard place in terms of communication reliability has been very important in order to know more about the limitations of the platform and, in this way, being able to improve or change some aspects.

This application opens a wide opportunity in the environmental control of food industry which is one of the strongest industries around Europe while it continues with a very interesting research line in the WSN field.

Acknowledgment

This project was partially founded by the Spanish Ministry of Industry under the Avanza I+D Program, Projects numbers TSI-020100-2008-172 and TSI-020100-2010-570.

References

- [1] J. Portilla, A. de Castro, E. de La Torre, and T. Riesgo, "A modular architecture for nodes in wireless sensor networks," *Journal of Universal Computer Science*, vol. 12, no. 3, pp. 328–339, 2006.
- [2] G. W. Eidson, S. T. Esswein, J. B. Gemmill et al., "The South Carolina digital Watershed: end-to-end support for real-time management of water resources," in *Proceedings of the 4th International Symposium on Innovations and Real-time Applications of Distributed Sensor Networks (IRADSN '09)*, vol. 2010, Los Alamitos, Calif, USA, May 2009.
- [3] H.-B. Xia, P. Jiang, and K.-H. Wu, "Design of water environment data monitoring node based on ZigBee technology," in *Proceedings of the International Conference on Computational Intelligence and Software Engineering (CiSE '09)*, pp. 1–4, Wuhan, China, December 2009.
- [4] N.-H. Nguyen, Q.-T. Tran, J.-M. Leger, and T.-P. Vuong, "A real-time control using wireless sensor network for intelligent energy management system in buildings," in *Proceedings of the IEEE Workshop on Environmental Energy and Structural Monitoring Systems (EESMS '10)*, pp. 87–92, Taranto, Italy, September 2010.
- [5] R. Szcwyczyk, 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)*, New York, NY, USA, November 2004.
- [6] P. Corke, T. Wark, R. Jurdak, H. Wen, P. Valencia, and D. Moore, "Environmental wireless sensor networks," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1903–1917, 2010.
- [7] B. Carter and R. Ragade, "The game of deployment," in *Proceedings of the IEEE Sensors Applications Symposium (SAS '11)*, San Antonio, Tex, USA, February 2011.
- [8] L. Mottola, G. P. Picco, M. Ceriotti, Ş. Gună, and A. L. Murphy, "Not all wireless sensor networks are created equal: a comparative study on tunnels," *ACM Transactions on Sensor Networks*, vol. 7, no. 2, article 15, 2010.
- [9] B. O'Flynn, F. Regan, A. Lawlor, J. Wallace, J. Torres, and C. O'Mathuna, "Experiences and recommendations in deploying a real-time, water quality monitoring system," *Measurement Science and Technology*, vol. 21, no. 12, Article ID 124004, 10 pages, 2010.
- [10] "Documento de identificación y análisis de los principales parámetros que afectan a la sostenibilidad," Project Deliverable no. 1. SustenTIC Project TSI-02011-2008-172.
- [11] CIAA (Confederation of Food and Drink Industries of the EU), Data and trends of the European Food and Drink Industry, CIAA, 24 pages, 2010.
- [12] CIAA (Confederation of Food and Drink Industries of the EU), Managing Environmental Sustainability in the Food and Drink Industries 2008, CIAA, 64 pages, 2008, http://envi.ciaa.eu/documents/brochure_CIAA_envi.pdf.
- [13] EU (European Commission), Integrated Pollution Prevention and Control (IPPC). Reference Document on Best Available Techniques in Food, Drink and Milk Industries, EU 629 pages, August 2006, <ftp://ftp.jrc.es/pub/eippcb/doc/fdm.bref.0806.pdf>.
- [14] CIAA (Confederation of Food and Drink Industries of the EU), Vision and Strategic Research Agenda, European Technology Platform Food for Life, 12 pages.
- [15] United Nations (UN), Indicators of Sustainable Development: Framework and Methodologies, UN, 94 pages, April 2001, http://www.un.org/esa/sustdev/csd/csd9_indi_bp3.pdf.

- [16] P. Burghher and P. Scherrer, Survey of Criteria and Indicators, 64 pages, 2005, <http://www.needs-project.org/docs/2bReportExperience.pdf/>.
- [17] EU (European Commission), Directive 2008/1/EC of the European Parliament and of the Council of concerning integrated pollution prevention and control, EU, 22 pages, January 2008.
- [18] G. Finnveden and Å. Moberg, "Environmental accounts and material flow analysis and other environmental systems analysis tools," in *Proceedings of the Workshop on Economic Growth, Material Flows and Environmental Pressure*, Stockholm, Sweden, April 2001.
- [19] G. ISO (International Standard Organisation), International Standard 14040, 1997E. Environmental management—Life Cycle assessment—Principles and framework, ISO, 12 pages, 2004.
- [20] D. Hunkeler and G. Rebitzer, "The future of life cycle assessment," *International Journal of Life Cycle Assessment*, vol. 10, no. 5, pp. 305–308, 2005.
- [21] CIAA (Confederation of Food and Drink Industries of the EU). Food and Drink Industry– Sustainability report. Industry as a partner for sustainable development.
- [22] E. M. Schau and A. M. Fet, "LCA studies of food products as background for environmental product declarations," *International Journal of Life Cycle Assessment*, vol. 13, no. 3, pp. 255–264, 2008.
- [23] Asociación Española de Normalización y Certificación (AENOR), UNE 77078:2002. Water quality. Technical specifications of general character for continuous pH measurements instruments in industrial wastes, AENOR, 8 pages, November 2010.
- [24] Asociación Española de Normalización y Certificación (AENOR), UNE 77079:2002. Water quality. Technical specifications of general character for continuous conductivity measurements instruments in industrial wastes, AENOR, 10 pages, November 2010.
- [25] Asociación Española de Normalización y Certificación (AENOR), UNE 77227:2001. Stationary source emissions. Determination of the volume flowrate of gas stream in ducts. Automated method, AENOR, 18 pages, December 2007.
- [26] Asociación Española de Normalización y Certificación (AENOR), UNE 77225:2000. Stationary source emissions. Measurement of velocity and volume flow rate of streams in ducts, AENOR, 26 pages, December 2007.
- [27] O. and Thomas and M.-F. Pouet, "Wastewater quality monitoring: on-line/on-site measurement," in *The Handbook of Environmental Chemistry, Water Pollution*, pp. 211–226, Springer, Berlin, Germany, 2005.
- [28] J. Portilla, A. Otero, E. de la Torre et al., "Adaptable security in wireless sensor networks by Using reconfigurable ECC hardware coprocessors," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 740823, 12 pages, 2011.
- [29] Y. Krasteva, J. Portilla, E. de la Torre, and T. Riesgo, "Embedded runtime reconfigurable nodes for wireless sensor networks applications," *IEEE Sensors Journal*, vol. 11, no. 9, pp. 1800–1810, 2011.

Research Article

A Mutual Algorithm for Optimizing Distributed Source Coding in Wireless Sensor Networks

Nashat Abughalieh,¹ Kris Steenhaut,^{1,2} Bart Lemmens,^{1,2} and Ann Nowé¹

¹Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium

²Erasmushogeschool Brussel—IWT, Nijverheidskaai 170, 1070 Brussel, Belgium

Correspondence should be addressed to Nashat Abughalieh, nashatg@gmail.com

Received 15 July 2011; Revised 10 November 2011; Accepted 16 November 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Nashat Abughalieh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless Sensor Networks (WSNs) are composed of small wireless nodes equipped with sensors, a processor, and a radio communication unit, all normally powered by batteries. For most WSN applications, the network is expected to function for several months or years. In the common monitoring application scenario, adjacent nodes in a WSN often sense spatially correlated data. Suppressing this correlation can significantly improve the lifetime of the network. The maximum possible network data compression can be achieved using distributed source coding (DSC) techniques when nodes encode at Slepian-Wolf rates. This paper presents contributions to the lifetime optimization problem of WSNs in the form of two algorithms: the Updated-CMAX (UCMAX) power-aware routing algorithm to optimize the routing tree and the Rate Optimization (RO) algorithm to optimize the encoding rates of the nodes. The two algorithms combined offer a solution that maximizes the lifetime of a WSN measuring spatially correlated data. Simulations show that our proposed approach may significantly extend the lifetime of multihop WSNs with nodes that are observing correlated data.

1. Introduction

Wireless Sensor Networks have a wide range of possible applications like environmental monitoring, home automation, military, industrial, and medical applications [1]. Network designers must consider factors such as the environment, cost, and hardware, while engineering a particular WSN. Different applications will prompt different architectural constraints and requirements [2]. For some applications, network designers will need to focus on bounded delivery time of sensed events to the Base Station (BS), for example, Tsunami warning systems. Other applications require the WSN to function for several months or years before being replaced, and designers are thus more concerned about the lifetime of the network, such as in environmental monitoring systems [3]. Since the network lifetime has been the main challenge in the design of many WSN applications [4], we address the problem of maximizing the network lifetime in this paper.

The nodes in WSNs are mostly powered by batteries. The energy of the batteries is utilized by the main building

blocks of a node: the sensors, the processor, and the radio unit. The radio is known to be the most energy consuming component of the node [5, 6]. For most WSNs, the radio unit has four functional modes: sleep, active, transmit and receive. The transmit, and receive modes have the highest power consumption while the sleep mode has the lowest power consumption. To improve the lifetime of the network (the lifetime of a WSN has many definitions [7]: some consider it to be the timespan from network startup to the death of a certain percentage of the nodes, others define it as the timespan from startup to the loss of coverage of a certain percentage of the monitored area), the node's radio has to be switched to sleep mode as much as possible. In this paper, we accomplish this by reducing the size of the packets through the application of lossless compression techniques which remove the redundancy in the spatially correlated sensed data. Lossless compression can be achieved by Source Coding techniques. Two lossless Source Coding methodologies for WSNs are described in the literature: Explicit Communications (EC) [8–10] and distributed source coding (DSC) [11–14].

The EC coding technique eliminates the redundancy in the spatially correlated sensed data while routing all data to the Base Station. Each node compresses its own sensed data according to the data flow passing through it from other nodes in the routing tree. The problem of finding the optimum routing path that achieves maximum compression with minimum network power consumption using EC is proven to be NP-hard [9]. The EC encoding process requires intensive processing at each sensor node to compress its sensed data, especially when the node has to forward compressed data from other nodes: the routing node needs to uncompress the data flow before being able to encode its own data in order to remove the redundancy due to spatial correlation.

The other methodology used for applying lossless source coding in WSNs is DSC. The concept of DSC was introduced by Slepian and Wolf in [15]. They derived the admissible rate region of two correlated sources and proved that two source encoders can compress their input data to a total rate which equals their joint entropy, without communication between the encoders, on condition that they are jointly decoded. Since, many authors have proposed source encoding systems that almost achieve the Slepian-Wolf theoretical limit. In [16], the authors used Coset Coding for compressing one of the two sources, while using the second source's data at the decoder to predict the data of the first source. In [17, 18], the authors used turbo codes and reached near the Slepian-Wolf theoretical limit. Their coding techniques are based on sending the data of the first node to the base station without coding, while encoding the second node's output with a turbo encoder and then send some parity bits of the encoder's output to the Base Station. The decoder at the BS uses the parity bits together with the data from the first node to estimate the second node's data. In [19–21], the authors implemented DSC using Low-Density Parity-Check (LDPC) codes. Turbo codes and LDPC codes enable DSC implementations which almost reach the Slepian-Wolf theoretical limit [17, 21]. Using the DSC theory, Cristescu et al. [22] studied how optimizing the rates of the nodes can minimize the network's total power consumption, but they did not address optimizing the lifetime of the network.

We propose contributions to the lifetime maximization problem of WSNs through the formulation of the problem's optimization equations and the development of algorithms which assign data rates and routing paths to the network nodes. The paper is organized as follows: we review the prior work on DSC for WSN in Section 2. We derive a system of equations for the optimal DSC rates and introduce the routing and rate assignment algorithms in Section 3. In Section 4, simulations results of the optimization algorithms are shown and discussed. We conclude the paper in Section 5.

2. Prior Work on DSC for WSNs

For a random source X , a rate $R \geq H(X)$ is sufficient to transmit X over a reliable channel to the BS. If we have two *independent and identically distributed (i.i.d.)* sources (X_1, X_2) , as shown in Figure 1, and they are encoded

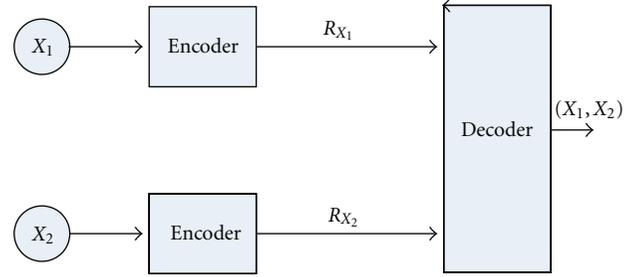


FIGURE 1: Slepian-Wolf coding for two correlated sources.

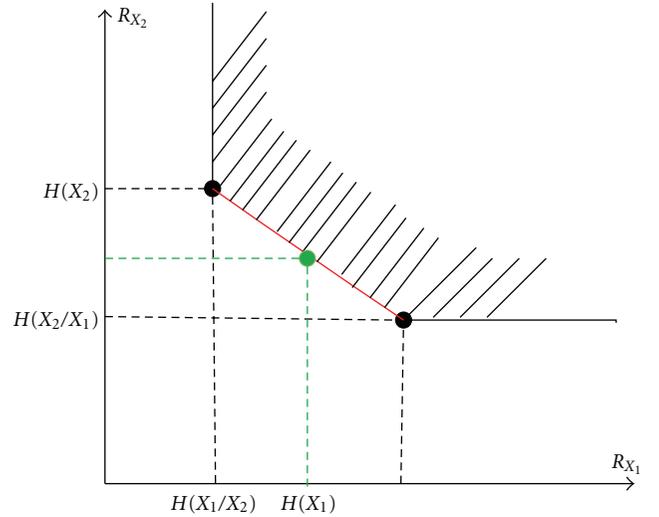


FIGURE 2: Rate region for Slepian-Wolf encoding.

separately, a total rate $R = R_{X1} + R_{X2} \geq H(X1) + H(X2)$ is required. However, as Slepian and Wolf have shown in their seminal paper [15], when both sources are correlated, a total rate $R = R_{X1} + R_{X2} \geq H(X1, X2)$ is sufficient, even when the two sources are encoded separately, as long as they are decoded jointly.

The achievable rate region according to the Slepian-Wolf coding theory is determined by the following equations:

$$\begin{aligned} R_{X1} &\geq H\left(\frac{X1}{X2}\right), \\ R_{X2} &\geq H\left(\frac{X2}{X1}\right), \end{aligned} \quad (1)$$

$$R_{X1} + R_{X2} \geq H(X1, X2).$$

The solution of this system of equations is shown in Figure 2. The minimum theoretical rate for the two correlated sources scenario is shown in Figure 2 with a red line. The two black dots at the corners of the optimum rate region correspond to the following rates:

$$\begin{aligned} R_{X1} &= H\left(\frac{X1}{X2}\right), & R_{X2} &= H(X2), \text{ or} \\ R_{X1} &= H(X1), & R_{X2} &= H\left(\frac{X2}{X1}\right). \end{aligned} \quad (2)$$

The Slepian-Wolf theory can be generalized to many sources [23]. If X_1, X_2, \dots, X_n are *i.i.d.*, but spatially correlated sources, then the set of rate vectors achievable using distributed source coding with separate encoders and joint decoder is defined by

$$R(S) \geq H\left(\frac{X(S)}{X(S^c)}\right) \quad (3)$$

for all $S \subseteq \{1, 2, 3, \dots, n\}$, where

$$R(S) = \sum_{i \in S} R_i. \quad (4)$$

Cristescu et al. [22] used the generalized Slepian-Wolf theory of multiple sources to optimize the encoding rates of WSN nodes. They set out to find an optimal transmission structure, that is, routing tree, and rate allocation for the nodes of a multihop WSN with multiple sensors X_i , $\{i = 1, 2, 3, \dots, n\}$ and one BS that minimizes a certain cost function which reflects the network's total power consumption:

$$\{R_i, d_i\}_{i=1}^N = \arg \min_{\{R_i, d_i\}_{i=1}^N} \sum_{i=1}^N F(R_i) \cdot d_i, \quad (5)$$

under the constraint of Slepian-Wolf encoding rates

$$\sum_{i \in S} R_i \geq H\left(\frac{X(S)}{X(S^c)}\right), \quad (6)$$

where R_i and d_i are the transmission rate of node X_i and the total weight (cost) associated to the routing links from node X_i to the Base Station (BS), respectively. They showed that for this specific type of cost function, the optimization problem can be decomposed into two separate problems: routing and rate optimization. Substituting the Shortest Path Tree (SPT) as the optimum routing tree, the optimization problem of (5) is reduced to

$$R_i = \arg \min_{R_i} \sum_{i=1}^N R_i d_{\text{SPT}}. \quad (7)$$

When numbering nodes in increasing order by weight of the routing paths from each node to the BS, so that nodes X_1, X_2, \dots, X_n have SPT weights $d_{\text{SPT}}(X_1) \leq d_{\text{SPT}}(X_2) \leq \dots \leq d_{\text{SPT}}(X_N)$, the solution to (7) under the constraint of Slepian-Wolf encoding is [22]

$$\begin{aligned} R_1 &\geq H(X_1), \\ R_2 &\geq H\left(\frac{X_2}{X_1}\right), \\ R_3 &\geq H\left(\frac{X_3}{X_1}, X_2\right) \\ &\vdots \\ R_N &\geq H\left(\frac{X_N}{X_{N-1}}, X_{N-2}, \dots, X\right). \end{aligned} \quad (8)$$

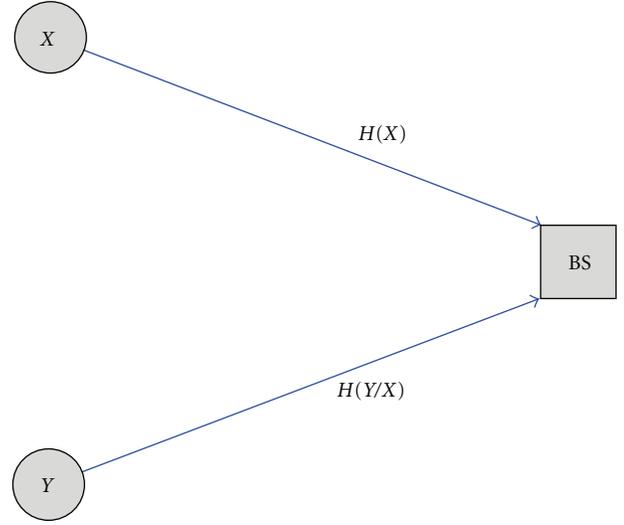


FIGURE 3: Two nodes WSN.

The node nearest to the BS has to encode at the rate of its entropy, while the second nearest node has to encode at the rate of the conditional entropy of itself given that the first source X_1 is available at the BS, and so on. Let us call the solution of (8) the Minimum Total Power (MTP) optimization. This optimization does however not maximize the network lifetime. To illustrate this, consider the example depicted in Figure 3, with two correlated sources placed at an equal distance from the BS. The MTP solution will assign encoding rates according to (2). While this rate assignment minimizes the network's total power consumption, the network lifetime is not maximized since the nodes are encoding at different rates. In [24, 25], the authors searched to optimize the lifetime of the WSN using distributed source coding on cluster level with one-hop communication between cluster nodes and the cluster head. They consider the decoder to run at the cluster head.

3. Lifetime Optimization of WSN with Correlated Sensors

The above-mentioned code design, where side information is assumed available at the decoder, is called *asymmetric* Slepian-Wolf coding. For the network in Figure 3, the maximum network lifetime can be achieved by encoding at the *symmetric* Slepian-Wolf coding point, which is the green point shown in Figure 2. Most DSC designs can reach almost the theoretical limit by implementing *asymmetric* Slepian-Wolf coding [17–19, 26]. When using *asymmetric* Slepian-Wolf coding, the lifetime of the network shown in Figure 3 can be maximized by periodically switching coding rates between nodes every time interval T . If T is fixed, the lifetime of the network can be considered as the maximum multiple of this interval, mT , $m = \{1, 2, \dots, M\}$, until the network is unable to perform its functionality successfully. Our goal is to maximize the lifetime of the network, which we can do by maximizing the value of M .

We model the WSN as a directed graph $G = (X, E)$, where the set $X = \{X_1, X_2, \dots, X_N\}$ is the wireless nodes and t is the BS that collects and decodes the network data. E is the set of links connecting the nodes of set X . The edge (X_i, X_j) is an element of E if X_j is in the transmission range of X_i . The total number of links, that is, the size of E , is K . We further denote the initial energy and the current energy of node X_i by $\text{IE}(X_i)$ and $\text{CE}(X_i)$, respectively. We allow the channel quality between nodes to change over time, so that for a link $(X_i, X_j) \in E$, X_i requires $e^m(X_i, X_j)$ energy to transmit one bit to node X_j at time index m . All nodes require a constant energy e_{rx} to receive one bit from any node. Node X_i is encoding its data using DSC at rate r_i^m . Since nodes also function as routers, we denote the rate of the data flow originally generated at node X_k forwarded from X_i to X_j in time slot m as $r_{i,j}^{m,k}$. Our goal is to maximize the number of time slots until the network is unable to deliver all nodes' data by optimizing the rate assignments:

$$\max_{r_i^m, r_{i,j}^{m,k}} M, \quad (9)$$

under the constraint of Slepian-Wolf encoding

$$\sum_{X_i \in X_s} r_i^m \geq H\left(\frac{X_s}{X_s^s}\right), \quad \forall X_s \subseteq X, \quad m = 1, 2, \dots, M. \quad (10)$$

The total flow at each node can be formulated as the difference between the output and the input data flows. If a node is a routing node, then the difference between its output and its input flows is zero. If the node is a source node, the difference between output and input flows is the rate of the node's encoded data:

$$\sum_{j \in X \setminus \{i\}} r_{i,j}^{m,k} - \sum_{j \in X \setminus \{i\}} r_{j,i}^{m,k} = \begin{cases} 0 & X_i \neq X_k, \\ r_i^k & X_i = X_k, \end{cases} \quad (11)$$

$$\forall X_i \in X, \quad X_k \in X, \quad \forall m = \{1, 2, \dots, M\}.$$

The relation between the data rate r_i^k and the transmission energy $e^m(X_i, X_j)$ has to be found in order to formulate the energy consumption as a function to be optimized. Starting from Shannon's point-to-point wireless communication theorem [27], which states that the maximum transmission rate at which a transmitter can communicate its data to a receiver through an AWGN channel is bounded by the capacity of that channel, we have

$$R \leq B \log(1 + \text{SNR}), \quad (12)$$

where B is the channel bandwidth and SNR is the signal power to the noise power ratio at the receiver antenna. Without loss of generality, we ignored any interference effects from concurrent communications as well as other channel characteristics like fading and shadowing. (12) can be reformulated into

$$e^{R/B} - 1 \leq \text{SNR}. \quad (13)$$

Since WSN nodes have low data rates, the left-hand part of (13) can be approximated into

$$e^{R/B} (1 - e^{-R/B}) \approx e^{R/B} \left(\frac{R}{B}\right) \approx \frac{R}{B}. \quad (14)$$

From (14) and (13), we derive that the data rate r_i^k and the transmission power of X_i have a linear relation if the bandwidth B and the noise power at the receiving node are constant. From this linear relation, we deduce that the power consumed by a sensor node for transmission on a particular link is the unit power consumption for transmission on that link multiplied by the data rate of the link. Likewise, the power consumption for reception on a link is the product of the unit reception power consumption and the rate of incoming data on that link. Thus, the total energy consumption of the radio units of all nodes in the network is expressed by the following relation:

$$\sum_{m=1}^M \sum_{X_j \in X \setminus \{X_i\}} \sum_{X_k \in X \setminus \{X_i\}} (e_{Tx}^m(X_i, X_j) r_{i,j}^{m,k} + e_{Rx} r_{j,i}^{m,k}) \leq \text{IE}(X_i), \quad \forall X_i \in X. \quad (15)$$

From (9), (10), (11), and (15), we can see that lifetime optimization using DSC comprises two optimization problems: the rate optimization problem and the route optimization problem. This optimization problem is NP-hard since the routing optimization problem itself is NP-hard [28]. It is generally difficult to construct a routing tree that maximizes the network lifetime due to the involvement of two optimization objectives: maximizing the residual energy of each node and minimizing the network's total energy consumption. These two objectives are not necessarily complementary and might even conflict: a routing tree could, for example, minimize the network's total energy consumption by placing a high burden on a particular node. However, a routing algorithm, which uses link weights based on an exponential function of the network's resource utilization, has been shown to cope very efficiently with this optimization problem [29]. The authors of [29] assign to each edge a cost that is exponential in the currently occupied link capacity in order to optimize the throughput of the network. Furthermore, they derived bounds on the competitive ratio of their routing algorithm and proved that no other online routing algorithm can achieve a better competitive ratio. In [30], this routing algorithm was adapted to optimize the lifetime of Wireless Sensor Networks by updating the links' weights with the energy utilization of the nodes.

The authors of [31] use the same optimization criteria as in [30], and links are assigned cost functions which are exponential in the transmitter's energy utilization. In [32], algorithms based on the same exponential penalization are proposed to optimize the routing tree of heterogeneous networks, in which nodes differ in energy capacity. In all aforementioned related work, the energy consumed for the reception of data is neglected.

We developed two algorithms for optimizing the routes and the rates used in a WSN. The routing algorithm, which is an improvement of the CMAX algorithm described in [30], penalizes the network links according to an exponential function of the energy consumption for transmission and reception. Regarding the Slepian-Wolf rates optimization problem, it is known that for an optimal *symmetric* rate assignment in a network with more than 3 nodes the complexity of the decoder is difficult to implement practically [21]. The realization of the decoder becomes feasible if we allow the nodes to encode at *asymmetric* rates, that is, one node is decoded separately and its output is used as side information to decode the second node, then these two outputs are used as side information to decode the third node and so on. We developed the Rate Optimization (RO) algorithm which assigns asymmetric rates to the nodes. The algorithm first assigns the rates using MTP to minimize the network's total energy consumption, after which it performs a tradeoff between minimizing network energy and maximizing nodes' residual energy by swapping the rates of the nodes. The RO algorithm requires global knowledge of all nodes' rates and residual energies. Thus, the RO algorithm is centralized and is running at the BS, which broadcasts the updated rate assignment every periodic interval T . The routing algorithm is also executed at the BS every T seconds, and the new routes are broadcast together with the rates assignment.

3.1. Routing and Rates Optimization Algorithms. In order to explain our route optimization algorithm, we first describe how CMAX [30] works, on which our extension UCMAX is built. After the BS collected the nodes' residual energies, the CMAX algorithm runs the following three steps.

Step 1. If all nodes have full energy (i.e., $CE(X_i) = IE(X_i)$), jump to Step 2 without modifying the graph G . Else, eliminate from G every edge $e(X_i, X_j)$ for which $CE(X_i) < e^m(X_i, X_j)$, then change the weight of every remaining edge $e^m(X_i, X_j)$ to $e^m(X_i, X_j) \times (\lambda^{\alpha(X_i)} - 1)$, where $\alpha(X_i)$ is the energy utilization ratio of node X_i :

$$\alpha(X_i) = \frac{IE(X_i) - CE(X_i)}{IE(X_i)} = 1 - \frac{CE(X_i)}{IE(X_i)}, \quad (16)$$

where λ is a constant that quantifies the penalty of using a link.

Step 2. Find the shortest path between each node and the BS using Dijkstra's algorithm in the modified graph.

Step 3. Let β be the length of the shortest path found in Step 2 ($\beta = \infty$ if no path was found). If $\beta \leq \sigma$, route the data along the shortest path, otherwise reject it.

The computational complexity of the CMAX algorithm is dominated by the shortest path computation (Step 2) and is $O(K + N \log N)$. The authors of [30] derived the competitive ratio of CMAX by comparing it to an optimal off-line routing algorithm. The competitive ratio of CMAX is found to be $O(\log N \rho)$, where ρ is the ratio of the edge with maximum

transmission energy to the edge with minimum transmission energy

$$\rho = \frac{\max_{i,j \in X} e(X_i, X_j)}{\min_{i,j \in X} e(X_i, X_j)}. \quad (17)$$

To find the competitive ratio, λ and σ are set to $\lambda = 2(N\rho + 1)$ and $\sigma = N \cdot \max_{i,j \in X} e(X_i, X_j)$, respectively. Setting $\sigma < \infty$ implies that packets may be rejected even if there is sufficient energy available to route the packet. Since our objective is to maximize the total number of packets delivered to the Base Station, we omit Step 3 in our modified routing algorithm, so that the route is not to be rejected if there is enough energy to deliver a packet over it.

Most WSN nodes, that are available on the market today, consume more energy while in receive mode rather than in transmit mode, even when the node is transmitting at the maximum power. The widely used transceiver chip CC2420 [33] consumes 18.8 mA in the receive mode, while it consumes 17.4 mA in the transmit mode at maximum transmission power. The authors of [30–32] do not take into account the energy spent in the receive mode while optimizing the routing tree. We updated CMAX to include the reception costs by modifying the weights of the graph's edges. The Updated-CMAX (UCMAX) runs the following steps.

Step 1. If all nodes have full energy (i.e., $CE(X_i) = IE(X_i)$), jump to Step 2 without modifying the graph G . Else, eliminate from G every edge $e(X_i, X_j)$ for which $CE(X_i) < e^m(X_i, X_j)$, then change the weight of every remaining edge to

$$e(X_i, X_j) \rightarrow e(X_i, X_j) \times (\lambda^{\alpha(X_i)} - 1) + e_{Rx} \times (\gamma^{\alpha(X_j)} - 1), \quad (18)$$

where λ and γ are constant parameters that quantify the penalty of using the link $e(X_i, X_j)$ based on the energy utilization of the transmitting node X_i and the receiving node X_j .

Step 2. Find the shortest path between each sensor node and the BS using Dijkstra's algorithm in the modified graph.

UCMAX avoids to route network data through nodes with low residual energy. The Rate Optimization algorithm (RO) runs on top of the optimized routing tree found by UCMAX as follows.

Step 1. Assign the rates to the nodes according to MTP using the routing tree found by UCMAX.

Step 2. Calculate the total energy consumption of the network

$$P_L = \sum_{i=1}^N w_i^m r_i^m, \quad (19)$$

where w_i^m is the total energy required to route one bit from node X_i to the BS, and r_i^m is the rate of node X_i during time slot m .

Step 3. Find the node with the minimum residual energy, let it be $X(\min)$.

Step 4. Search for another node with lower data rate and higher residual energy and name it $X(tmp)$. The two nodes should not be on the same routing path to the BS.

Step 5. Swap the rates between $X(\min)$ and $X(tmp)$, so that if $X(\min)$ is encoding at rate $H(X_1)$ and $X(tmp)$ at rate $H(X_2/X_1)$, $X(\min)$'s rate should become $H(X_1/X_2)$ while $X(tmp)$'s rate should become $H(X_2)$.

Step 6. Calculate the total network power with (19) and store it in a vector P . Go back to Step 4 and repeat for all possible rate switches. Choose the rate swap with the minimum total power in P and let us name the new total power P_L^{new} .

Step 7. If P_L^{new} is less than zP_L , where z is a constant parameter, accept the new rates. Else, do not switch the rates.

Step 8. Go back to Step 3 and repeat until all possible rate switches are tested for all nodes.

Parameter z allows our optimization to balance between minimum total network energy and maximum per node energy. At $z = 1$, the network retains the minimum total network energy achieved by MTP while trying to maximize per node energy. For $z < 1$, the RO algorithm is simply the MTP algorithm.

4. Simulations and Results

4.1. Experimental Setup. MatLab simulations are used to evaluate the performance of the optimization algorithm. We simulate an environment area of $100 \times 100 \text{ m}^2$ and consider two network configurations: in the first one nine nodes are located on a 3×3 grid, while in the second one the set of nodes is extended to twenty-five nodes arranged in a 5×5 grid. The BS is located at the center in both grids. The two grids allow us to compare the performance of the optimization algorithms with different node densities. The energy consumption of the nodes is approximated by the energy consumed by the radio transceiver. The energy for reception, e_{Rx} , is considered the same for all nodes. Without loss of generality, the energy for transmission $e_{Tx}^m(X_i, X_j)$ is assumed to depend on the distance between the nodes X_i and X_j . We use the following model:

$$e_{Tx}(X_i, X_j) = [\beta d(X_i, X_j)^\kappa + \rho] \times T_{Tx}, \quad (20)$$

where β , κ , and ρ are constants and their values depend on the radio chip's characteristics and the environmental conditions. T_{Tx} is the packet transmission time. The channel parameters used in our simulations are shown in Table 1 and are calculated according to the work of [34]. The correlation between the sensed data at the nodes is represented by a Gaussian model [22]

$$f(x) = \frac{1}{\sqrt{2\pi} \det(K)^{1/2}} e^{-((1/2)(X-\mu)K^{-1}(X-\mu))}, \quad (21)$$

TABLE 1: Experiment parameters.

$\beta = 5.219 \times 10^{-4}$	$\sigma = 1$
$\kappa = 3.5$	$c = 0.001$
$\rho = 1.2 \times 10^{-5}$	Bit rate = 250 kb/s
Transmission range = 100 m	Maximum packet size = 128 bytes
$IE(X_i) = 10 \text{ Joule}$	Maximum $T_{Tx} = 4.1 \text{ ms}$
$e_{Rx} = 59.1 \times 10^{-3} \times T_{Tx} \text{ Joule}$	

where K is the covariance matrix which represents the spatial correlation between the measurements. K is created by assuming that these correlations are changing according to the distance between the nodes. More precisely, the following model $K_{i,j} = \sigma^2 \exp(-c|d(X_i, X_j)|^2)$ is used to define this relationship, where $K_{i,j}$ and $d(X_i, X_j)$ represent the correlation and the distance between the nodes X_i and X_j , respectively. σ^2 is the variance of the nodes' sensed data (we consider all nodes to have the same variance) and c is the attenuation factor of the correlation between the nodes.

4.2. Simulation Results. The role of λ in the CMAX algorithm is studied in [30], where it acts as a penalty factor for using the links between nodes with low residual energy in the WSN. For $\lambda = 1$, the routing structure is the same as the Shortest Path Tree. Increasing λ improves the per node residual energy at the expense of a higher total network energy consumption. It is shown in [30] through experiments that at $\sigma = \infty$ and for large values of λ , the number of total delivered messages is maximized and becomes insensitive to increasing values of λ . In our simulation, we set $\sigma = \infty$ and $\lambda = 10,000$ for all experiments to analyze the effect of other factors on the lifetime of the network.

4.2.1. UCMA versus CMAX. For both the 3×3 and 5×5 grids, we execute several runs of the routing algorithm while changing the value of γ at each run. The rates of the nodes are assigned with MTP. The update period (the total number of measurement collection cycles to the Base Station before the algorithm calculates a new routing tree) T is set to 1000. Figure 5 presents the network lifetime (total number of route updates until there is no possible route to deliver all network data) on the y -axis and γ on the x -axis. The network lifetime of CMAX is constant since λ is constant and the CMAX algorithm is not affected by changes in γ . In the simulations, we include the reception energy utilization in the calculations of the edges' new weights. As shown in Figure 5, the 3×3 network has a longer lifetime than the 5×5 network with the CMAX routing optimization, even though the 5×5 network is denser. The 5×5 network has a shorter lifetime because the CMAX algorithm does not take into account energy consumed during receive mode, which can be much higher than the energy used in transmit mode in a dense network. As we pointed out before, for most WSN nodes' transceivers, the power consumption in receive mode is higher than the transmission energy utilization, even at the maximum transmission power. In the 3×3 network, the nodes consume more energy in the transmit mode compared

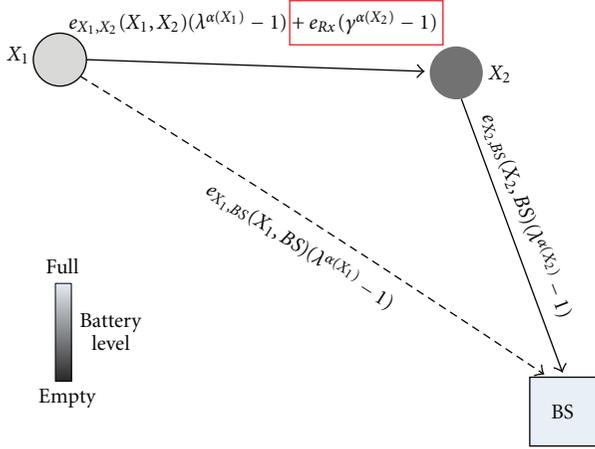


FIGURE 4: A two-node network with a Base Station.

to the nodes in the 5×5 network which results in a better optimization of the routing tree for the 3×3 network compared to the 5×5 network when using CMAX. The large gap between the estimated power consumption of CMAX and the calculated power consumption using (20) leads to a shorter lifetime for the 5×5 network.

The lifetime of the network varies with γ in the UCMAX algorithm. At $\gamma = 1$, UCMAX performs the same as CMAX. However by increasing γ , the lifetime of the network improves and reaches a maximum at $\gamma = 10$ for both the 3×3 and the 5×5 networks. The optimum value of γ depends on the parameters used in the energy consumption model of (20).

To describe the advantage of UCMAX over CMAX, let us consider the network in Figure 4. Using the CMAX optimization algorithm, node X_1 may choose to route its data through X_2 if the sum of the weights of the links (X_1, X_2) and (X_2, BS) is less than the weight of link (X_1, BS) . Recall that CMAX neglects the energy consumption for reception at X_2 . By including this reception energy utilization of X_2 in the weight of the edge (X_1, X_2) , UCMAX can decide to avoid routing through X_2 , since X_2 pays a double price in terms of energy, that is, for receiving and transmitting.

With UCMAX, the 5×5 network, which is denser than the 3×3 network, has a longer lifetime, since in dense networks the distance between nodes is shorter and thus the nodes require less transmission power.

4.2.2. RO Algorithm Evaluation. To compare the RO algorithm to MTP, we applied the RO algorithm on top of UCMAX. λ and γ are set to $\lambda = 10,000$ and $\gamma = 10$, respectively. Figure 6 depicts the variations in the network lifetime while changing the parameter z . For $z < 1$, there is no improvement in the lifetime when compared to MTP. For $z \geq 1$, the RO algorithm shows large improvements in network lifetime for both the 5×5 and 3×3 networks. The improvement in the lifetime at $z = 1$ is due to the the grid structure of the networks with the BS positioned at the center. In a grid network, some nodes are at an equal distance from the BS. When assigning different rates to these

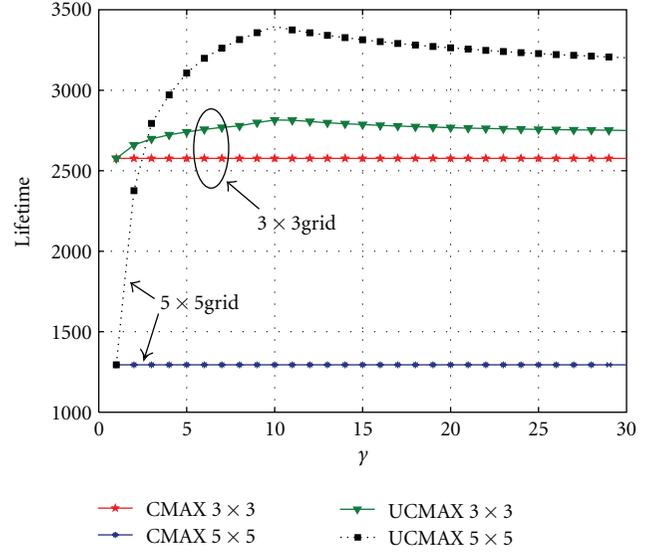


FIGURE 5: Routing optimization for maximizing the network lifetime.

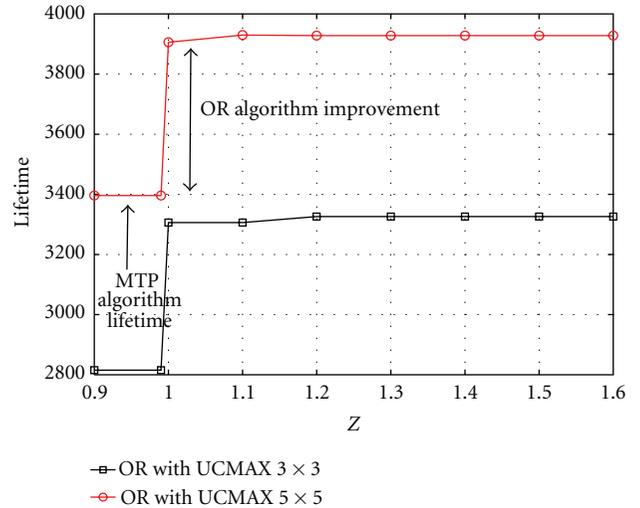


FIGURE 6: Network lifetime with optimum rate algorithm.

nodes, the total network power P_L will remain the same. After each update period T , the RO algorithm switches the rates between the nodes with equal distance to the BS. Low rates are assigned to nodes with minimum residual energy while healthier nodes are penalized with higher rates. The network lifetime is insensitive to high values of z because assigning low rates to nodes with low residual energy close to the BS and high rates to more distant nodes is not improving the lifetime of the network, since the nodes close to the BS always need to route the data from the farther nodes.

4.2.3. Update Period T . In Figure 7 we show the effect of the update frequency of the routing tree and the encoding rates on the lifetime of the network. Before each update, the

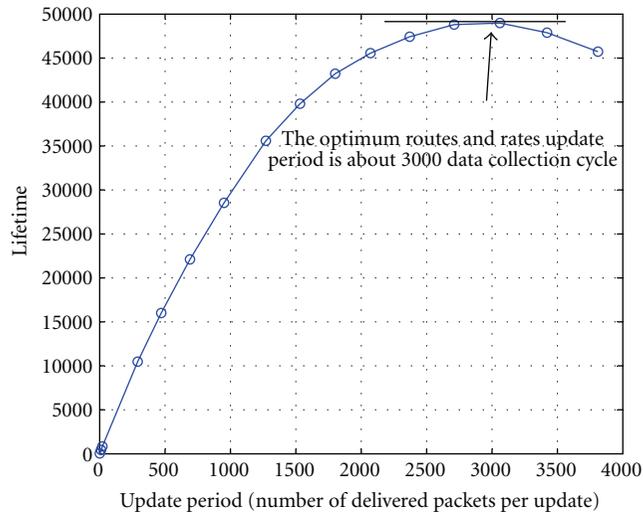


FIGURE 7: Effect of update period T on the network lifetime.

nodes need to communicate their residual energy to the BS, and the BS subsequently broadcasts the optimized routing tree and rates back to the nodes. We assume that the BS can encapsulate the routing tree and the rates of the nodes in one broadcast packet. When the BS broadcasts the packet, the nearest nodes receive the packet and then forward it to the nodes further in the network and so on. We consider that each node consumes energy equal to the reception and the transmission of one packet during the broadcast process and we neglect the energy spent in collecting the nodes' residual energies. By running the UCMA and RO algorithms on the 5×5 network with different update periods T , we found that at $T = 3000$ the network has the longest lifetime. At fast updating rates, the communication overhead caused by the broadcasts reduces the network lifetime, while at slow updating rates, the algorithm does not track accurately the depletion rate of the batteries of the nodes.

5. Conclusion

In this paper we considered Wireless Sensor Networks placed in a specific geographical area, gathering correlated information from multiple nodes that forward their data to a Base Station. We addressed the maximization of network lifetime through the application of distributed source coding for the compression of the spatially correlated data. The motivation for using DSC instead of Explicit Communication is the possibility of decomposing the optimization problem in two separate problems: optimizing the routes and the rates independently. The paper presents two algorithms: the first one is a routing optimization algorithm and the second one is a rate optimization algorithm. The first algorithm, the Updated-CMAX algorithm (UCMA), improves the CMAX algorithm, as presented in the literature, by taking into account the energy utilization of nodes in receive mode. The second algorithm, the Rate Optimization algorithm (RO), balances between minimizing total network energy consumption and the per node energy consumption while

assigning Slepian-Wolf encoding rates. The RO algorithm assigns low rates to nodes with low residual energy and higher rates to nodes with excessive energy.

Our experiments show that UCMA provides a significant improvement in terms of network lifetime in comparison to CMAX. With respect to the minimum total network Slepian-Wolf rates, our RO algorithm improved lifetime by 17%. The two algorithms combined provide a full-optimized solution that maximizes the lifetime of networks collecting correlated data. The optimal update period between successive rate and route optimizations is also derived by taking into account the broadcasting energy consumption needed to send the optimized rate and route assignments from the Base Station.

Acknowledgments

The work presented in this paper was supported by EMECW (Erasmus Mundus External Corporation Window lot3) and the FWO (Fonds Wetenschappelijk Onderzoek), project G.0219.09N.

References

- [1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [2] 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.
- [3] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "SensorScope: out-of-the-box environmental monitoring," in *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN '08)*, pp. 332–343, April 2008.
- [4] C. Hua and T. S. P. Yum, "Optimal routing and data aggregation for maximizing lifetime of wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 892–903, 2008.
- [5] Q. Gao, K. J. Blow, D. J. Holding, and I. Marshall, "Analysis of energy conservation in sensor networks," *Wireless Networks*, vol. 11, no. 6, pp. 787–794, 2005.
- [6] G. Xing, C. Lu, Y. Zhang, Q. Huang, and R. Pless, "Minimum power configuration for wireless communication in sensor networks," *ACM Transactions on Sensor Networks*, vol. 3, no. 2, Article ID 1240231, 2007.
- [7] K. Sha and W. Shi, "Modeling the lifetime of wireless sensor networks," *Sensor Letters*, vol. 3, no. 2, pp. 126–135, 2005.
- [8] S. Patten, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 4, pp. 1–33, 2008.
- [9] R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer, "Network correlated data gathering with explicit communication: NP-completeness and algorithms," *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, pp. 41–54, 2006.
- [10] H. Luo and G. J. Pottie, "Designing routes for source coding with explicit side information in sensor networks," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1401–1413, 2007.

- [11] H. Wang, D. Peng, W. Wang, and H. Sharif, "Optimal rate-oriented routing for distributed source coding in wireless sensor network," in *Proceedings of the 2nd ACM International Workshop on Quality of Service and Security in Wireless and Mobile Networks (Q2SWinet '06)*, pp. 55–58, ACM Press, New York, NY, USA, 2006.
- [12] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codes for distributed source coding," *Signal Processing*, vol. 86, no. 11, pp. 3123–3130, 2006.
- [13] A. Kashyap, L. A. Lastras-Montano, C. Xia, and Z. Liu, "Distributed source coding in dense sensor networks," in *Proceedings of the Data Compression Conference*, no. 2, pp. 13–22, IEEE, 2005.
- [14] D. Marco and D. L. Neuhoff, "Reliability vs. efficiency in distributed source coding for field-gathering sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 161–168, New York, NY, USA, ACM Press, 2004.
- [15] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.
- [16] S. S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 51–60, 2002.
- [17] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '01)*, vol. 2, pp. 1400–1404, IEEE, 2001.
- [18] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proceedings of the Data Compression Conference*, 2002.
- [19] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, 2002.
- [20] C. F. Lan, A. D. Liveris, K. Narayanan, Z. Xiong, and C. Georghiades, "Slepian-Wolf coding of multiple M-ary sources using LDPC codes," in *Proceedings of the Data Compression Conference (DCC '04)*, p. 549, March 2004.
- [21] A. D. Liveris, C. F. Lan, F. Narayanan, Z. Xiong, and C. N. Georghiades, "Slepian-Wolf coding of three binary sources using LDPC codes," in *Proceedings of the International Symposium on Turbo Codes and Related Topics*, pp. 1–4, 2003.
- [22] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "Networked Slepian-Wolf: theory, algorithms, and scaling laws," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4057–4073, 2005.
- [23] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, vol. 306 of *Wiley Series in Telecommunications*, Wiley-Interscience, New York, NY, USA, 1991.
- [24] H. Arjmandi, M. Taki, and F. Lahouti, "Lifetime maximized data gathering in wireless sensor networks using limited-order distributed source coding," *Signal Processing*, vol. 91, no. 11, pp. 2661–2666, 2011.
- [25] H. Arjmandi and F. Lahouti, "Resource optimized distributed source coding for complexity constrained data gathering wireless sensor networks," *IEEE Sensors Journal*, vol. 11, no. 9, Article ID 5705538, pp. 2094–2101, 2011.
- [26] D.S. Lun, M. Médard, T. Ho, and R. Koetter, "Network coding with a cost criterion," in *Proceedings of the International Symposium on Information Theory and its Applications (ISITA '04)*, pp. 1232–1237, Citeseer, April 2004.
- [27] C. E. Shannon, "The mathematical theory of communication," *M.D. Computing : Computers in Medical Practice*, vol. 14, no. 4, pp. 306–317, 1948.
- [28] Q. Li, J. Aslam, and D. Rus, "Online power-aware routing in wireless Ad-hoc networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MOBICOM '01)*, pp. 97–107, 2001.
- [29] B. Awerbuch, Y. Azar, and S. Plotkin, "Throughput-competitive on-line routing," in *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*, pp. 32–40, IEEE Computer Society, Washington, DC, USA, 1993.
- [30] K. Kar, M. Kodialam, T. V. Lakshman, and L. Tassiulas, "Routing for network capacity maximization in energy-constrained ad-hoc networks," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications (IEEE INFOCOM '03)*, vol. 1, pp. 673–681, IEEE Societies, 2003.
- [31] W. Liang and Y. Liu, "Online data gathering for maximizing network lifetime in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 1, pp. 2–11, 2007.
- [32] L. Lin, N. B. Shroff, and R. Srikant, "Asymptotically optimal energy-aware routing for multihop wireless networks with renewable energy sources," *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1021–1034, 2007.
- [33] "CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF transceiver," Tech. Rep., Texas Instruments, 2007.
- [34] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2349–2360, 2005.

Research Article

Interference-Free Wakeup Scheduling with Consecutive Constraints in Wireless Sensor Networks

Junchao Ma and Wei Lou

Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong

Correspondence should be addressed to Wei Lou, csweilou@comp.polyu.edu.hk

Received 13 June 2011; Revised 30 September 2011; Accepted 13 October 2011

Academic Editor: Yuhang Yang

Copyright © 2012 J. Ma and W. Lou. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wakeup scheduling has been widely used in wireless sensor networks (WSNs), for it can reduce the energy wastage caused by the idle listening state. In a traditional wakeup scheduling, sensor nodes start up numerous times in a period, thus consuming extra energy due to state transitions (e.g., from the sleep state to the active state). In this paper, we address a novel interference-free wakeup scheduling problem called compact wakeup scheduling, in which a node needs to wake up only once to communicate bidirectionally with all its neighbors. However, not all communication graphs have valid compact wakeup schedulings, and it is NP-complete to decide whether a valid compact wakeup scheduling exists for an arbitrary graph. In particular, tree and grid topologies, which are commonly used in WSNs, have valid compact wakeup schedulings. We propose polynomial-time algorithms using the optimum number of time slots in a period for trees and grid graphs. Simulations further validate our theoretical results.

1. Introduction

Wireless sensor networks (WSNs) consist of hundreds to thousands of tiny, inexpensive, and battery-powered wireless sensing devices which organize themselves into multihop radio networks. As the batteries of most sensor nodes are nonrechargeable, one key challenging issue is to schedule the activities of nodes to minimize the energy consumption. The major source of energy wastage [1–3] in WSNs is the idle listening state in the radio modules, which in fact consumes almost as much energy as receiving. Therefore, nodes are generally scheduled to sleep when the radio is not in use [4, 5] and wake up when necessary. By using wakeup scheduling, nodes could operate in a low-duty-cycle mode, and periodically start up to check the channel for activity.

In wireless networks, the packets transmitted by a node may be received by all the nodes within its transmission range due to the broadcast nature of the wireless medium. Therefore, the transmission of one link may interfere with the reception of another link. To avoid the interferences among the communication links, we adopt the time division multiple access (TDMA) MAC protocols, such as TRAMA [6], DCQS [7], and DRAND [8]. TDMA protocols have the natural advantages of having no contention-introduced

overhead or collisions [1]. In such protocols, the time is divided into equal intervals referred to as *time slots*. Correspondingly, nodes turn on the radio during the assigned time slots and turn off the radio when they are not transmitting or receiving in the wakeup scheduling. For multiple transmission links can communicate at the same time in wireless networks, several nodes can wake up to transmit their packets simultaneously when they do not interfere with each other. Therefore, we attempt to minimize the number of time slots assigned to each node while guaranteeing interference-free among the communication links.

The previous studies [9, 10] in the wakeup scheduling did not, however, consider all possible energy consumption, especially the energy consumed in the *state transitions*, for example, from the sleep state to the listening state or transmitting state. After such a scheduling, a node may start up numerous times in a period to communicate with its neighbors. Note that the typical startup time is on the order of milliseconds, while the transmission time may be less than the startup time if the packets are small [11]. Take Tmote Sky [12] as an example, the time and energy consumption to activate a node is about 1.4 ms and 17 μ J, respectively, whereas the time and energy consumption to transmit 1 byte

is about 0.032 ms and 1.7 μ J, respectively. If a sensor node starts up too frequently, it not only needs extra time, but also consumes extra energy for state transitions. Moreover, it reduces the battery capacity due to the current surges in the state transitions.

Figure 1 shows the battery voltage of Tmote Sky sensors with different startup frequencies but with the same duty cycle (50%): one starts up every 20 ms, stays in the receive state for 10 ms, and turns to the sleep state for the rest period; the other one starts up every 100 ms, stays in the receive state for 50 ms, and turns to the sleep state for the rest period. We can see that about 8% battery voltage can be saved by reducing the startup frequency from five times to once in every 100 ms. To minimize the energy cost, the state transitions should be considered in the wakeup scheduling design. Unlike the previous work, we are interested in the scheduling with *consecutive constraints*, where all the links incident to a node are assigned consecutive time slots so that each node needs to wake up only once to communicate bidirectionally with its neighbors.

In [13], energy-efficient centralized and distributed algorithms are proposed to reduce the frequency of state transitions of each node to twice in a data-gathering tree: once for receiving data from its children, and once for sending data to its parent. If the network topology is a directed acyclic graph (DAG) where each node v_i has k_i parents, the scheduling in [13] would require v_i to wake up $k_i + 1$ times as the parent nodes are not scheduled together. Moreover, the *two-way* (or bidirectional) communication is not taken into consideration. An interesting problem is to design an efficient scheduling where a node could wake up *only once* and finish all communication tasks with its neighbors consecutively and bidirectionally.

In this paper, we propose *compact wakeup scheduling*, a novel time division multiple access (TDMA) approach to the wakeup scheduling problem, to minimize the frequency of state transitions. Compact wakeup scheduling assigns consecutive time slots to all the links incident to a node v_i so that v_i can start up only once to communicate bidirectionally with all its neighbors in one scheduling period T .

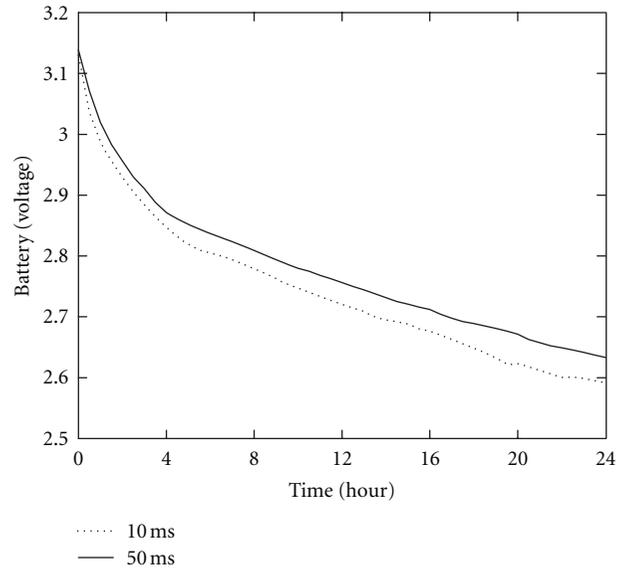
Apart from reducing the transient time and energy cost in the state transitions, compact wakeup scheduling also has other benefits. The network delay, which is a major concern in time-critical monitoring systems like that in [14], can be reduced. For instance, a sensor may need to wait until all its neighbors wake up so that it can collect the real-time data from these neighbors to make the local computation on these data. Note that compact wakeup scheduling cannot only reduce the state transitions of transceivers, but also reduce the state transitions of other components in the nodes, such as external memory and sensing devices.

The main contributions of this paper are summarized as follows.

- (i) We formulate the compact wakeup scheduling problem in WSNs to minimize the frequency of state transitions, and prove it to be NP-complete.
- (ii) We present polynomial-time algorithms using the optimum number of time slots in a period for trees



(a) Tmote Sky



(b) Battery

FIGURE 1: (a) Tmote Sky sensor. (b) The battery voltage of Tmotes with different startup frequency. AA Carbon-Zinc batteries are used in the experiment. 10 ms and 50 ms are active time in each period.

and grid graphs. In grid graphs, we point out all the possible coloring patterns and give the lower bound as well as the upper bound of the compact wakeup scheduling.

- (iii) We develop simulations to show the efficiency of compact wakeup scheduling.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 describes the system model and formulates the compact wakeup scheduling problem. Section 4 presents polynomial-time algorithms for trees and grid graphs. Section 5 shows the performance evaluation. Section 6 concludes the paper and provides directions for future research.

2. Related Work

Wakeup scheduling has attracted a lot of interest in WSNs in virtue of its energy efficiency. S-MAC [1] is a contention-based MAC scheme. In S-MAC, nodes periodically sleep

and wake up, and each active period is of a fixed size with a variable sleep time. T-MAC [2] improves S-MAC by adopting a dynamic duty cycle, that is, transmitting all messages in bursts and ending the listening period when nothing is heard within a limited time. DW-MAC [4] allows nodes to wake up on demand during the sleep period and ensures that data transmissions do not collide at their intended receivers. TreeMAC [15] is a localized TDMA MAC protocol, which is designed to achieve high throughput and low congestion with low overhead. PW-MAC [16] minimizes the energy consumption by enabling senders to predict the wakeup times of receivers based on asynchronous duty cycling.

Link scheduling is time slot assignments to communication links in TDMA MAC protocols. Ramanathan and Lloyd [17] consider both the tree networks and arbitrary networks, and the performance of the proposed algorithms is bounded by the thickness of a network. In [18], Gandham et al. propose a link scheduling algorithm involving two phases. In the first phase, a valid edge coloring is obtained in a distributed fashion. In the second phase, each color is mapped to a unique time slot, and the hidden terminal problem as well as the exposed terminal problem is avoided by assigning each edge a direction of transmission. The overall scheduling requires at most $2(\Delta + 1)$ time slots when the topologies are acyclic, where Δ is the maximum degree of a graph. In [10], Wang et al. propose a degree-based heuristic algorithm with performance guarantee to obtain a good interference-free link scheduling to maximize the throughput of the network. In the algorithm, the sensors are scheduled individually in a predefined order without consecutive assignment of time slots, and each node is assigned the best possible time slot to transmit or receive without causing interferences to the already-scheduled sensors. In [19], Wu et al. propose efficient centralized and distributed scheduling algorithms that reduce the energy cost of state transitions and also propose an efficient method to construct an energy-efficient data-gathering tree. In [13], Ma et al. address the contiguous link scheduling problem by applying the interval vertex coloring in a merged conflict graph and assigning consecutive time slots to the links incident to one node to achieve better energy efficiency.

Instead of applying the interval vertex coloring, we apply the interval edge coloring in the compact wakeup scheduling. Interval edge coloring, introduced by Asratian and Kamalian [20] (available in English as [21]), is a special edge coloring in which the colors of edges incident to the same vertex must be contiguous, that is, the colors must be composed of an integer interval. Not every graph has an interval edge coloring, since a graph G with an interval edge coloring belongs to Class 1 graphs where the chromatic number of edge coloring is equal to the maximum degree Δ [21]. Sevastjanov [22] proves that the problem of determining the existence of an interval edge coloring is NP-complete, even for bipartite graphs, and Kubale [23] proves that the interval edge coloring problem with forbidden colors is also NP-complete. Experiments [24] with small and sparse graphs show that the existence of an interval edge-coloring is with high probability. Some examples of graphs with

interval edge-colorings are trees, complete bipartite graphs, and grid graphs [20, 21, 25]. Giaro and Kubale give several polynomially solvable graphs in [26].

Compared to the former studies on wakeup scheduling, the compact wakeup scheduling could minimize the energy cost of state transitions, and sensors can start up only once in a period T . Furthermore, the compact wakeup scheduling considers two-way (or bidirectional) communication while our early work [13, 19] only considers one-way communication.

3. Problem Formulation

In this section, we first present the system model then formulate the compact wakeup scheduling problem.

3.1. System Model. We assume that a WSN has n static sensor nodes equipped with single omnidirectional antennas, and all the nodes have the same communication range. The network is represented as a communication graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ denotes the set of nodes and $E = \{e_1, e_2, \dots, e_m\}$ denotes the set of edges referred to all the communication links. If $\{v_i, v_j\} \subseteq V$, the edge $e = (v_i, v_j) \in E$ if and only if v_j is located within the communication range of v_i . We assume that nodes have the ability of data aggregation and can use one time slot to transmit data in one link.

Each node operates in three states: active state (transmit, receive, and listen), sleep state, and transient state (state transition). The transient state comprises two processes: startup (from the sleep state to the active state) and turndown (from the active state to the sleep state). The startup process from the sleep state to the active state includes radio initialization, radio and its oscillator startup, and the switch of radio to active [27]. The startup process is slow due to the feedback loop in the phase-locked loop (PLL) [28], and a typical setting time of the PLL-based frequency synthesizer is on the order of milliseconds.

We assume that the interference range is equal to the communication range. Two types of interferences, primary interference and secondary interference [17], exist in the network. The primary interference occurs when a node has more than one communication task in a single time slot. Typical examples are sending and receiving at the same time and receiving from two different transmitters. The secondary interference (or called *the hidden terminal problem* [29]) occurs when a node v_i receives packets from a transmitter v_j and v_i is also within the communication range of another transmitter v_k which is intended for other nodes.

3.2. Problem Formulation. In TDMA wakeup schedulings, each bidirectional communication link l_{ij} is assigned two time slots: one time slot is that v_i is a transmitter and v_j is a receiver, while the other one is that v_j is a transmitter and v_i is a receiver. In the two time slots, nodes v_i and v_j start up and switch from the sleep state to the active state. After that, nodes v_i and v_j switch to the sleep state again. We can see that node v_i may start up $2w_i$ times to communicate

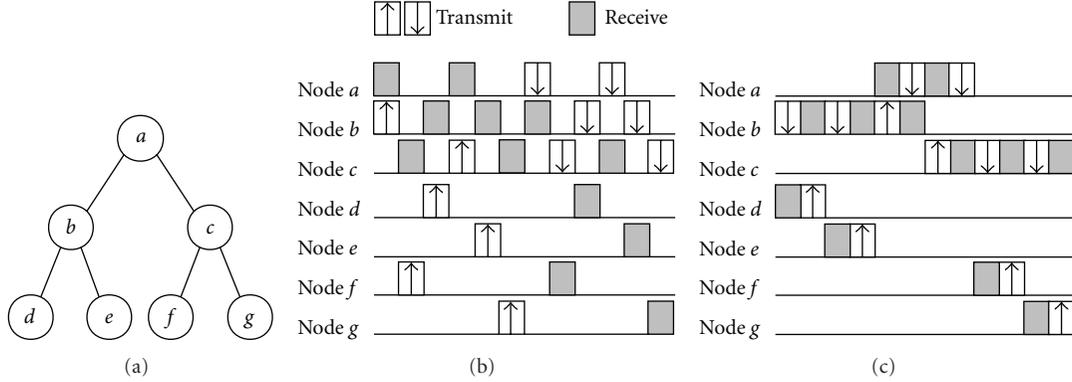


FIGURE 2: Wakeup scheduling and compact wakeup scheduling: (a) network topology, (b) wakeup scheduling, (c) compact wakeup scheduling.

bidirectionally with its neighbors in a scheduling period T in the worst case, where w_i is the number of neighbors of v_i . To minimize the frequency of state transitions, we propose a new scheduling approach called compact wakeup scheduling.

Definition 1. Compact wakeup scheduling is an interference-free wakeup scheduling aiming to assign consecutive time slots to all the links incident to a node v_i , and then v_i needs to start up only once to communicate bidirectionally with all its neighbors.

Compact wakeup scheduling attempts to assign consecutive time slots to all the links incident to a node, but it may fail to find such a scheduling. If it succeeds, the scheduling is said to be a *valid* scheduling. If not, the scheduling is said to be not a *valid* scheduling.

In the compact wakeup scheduling, the two time slots assigned to each bidirectional link l_{ij} are adjacent, and node v_i can finish its bidirectional communication with v_j in consecutive time slots. Figure 2(a) shows the given network topology. Figure 2(b) shows a wakeup scheduling, in which a node starts up numerous times in a period. Figure 2(c) shows a compact wakeup scheduling, in which a node could start up only once to communicate bidirectionally with its neighbors. Compact wakeup scheduling can reduce the time for a node to collect the data from its neighbors. As shown in Figures 2(b) and 2(c), node c needs 5 more time slots to communicate with all its neighbors without the compact wakeup scheduling.

An edge coloring of graph G is called a *valid coloring* if any two adjacent edges of G are assigned different colors. A valid coloring of G is called an *interval (or consecutive) edge coloring* if, for each vertex v , the colors of edges incident to v form an integer interval.

Theorem 2. The problem of deciding whether a valid compact wakeup scheduling exists for an arbitrary graph G is NP-complete.

Proof. The compact wakeup scheduling problem is in NP. To verify whether a scheduling is a solution to the compact

wakeup scheduling problem, we need to check (i) all the links incident to a node are assigned consecutive time slots; (ii) the scheduling is interference-free. Verifying (i) and (ii) requires $O(n)$ and $O(n^2)$ operations, respectively, where n is the number of nodes. It is clearly that this verification can be done in polynomial time.

To prove that the compact wakeup scheduling problem is NP-hard, we first restate the interval edge-coloring problem with forbidden colors which is NP-complete [21, 23]. “Given a graph G , a forbidding function F which represents the colors that cannot be assigned to each edge e , and an integer k , does there exist an interval edge coloring of G using k colors and avoiding F ?” The interference, such as the hidden terminal problem, in the compact wakeup scheduling is a special case of the forbidding function in the interval edge-coloring. Thus, the compact wakeup scheduling is equivalent to the interval edge-coloring with forbidden colors, which is NP-hard. Therefore, the problem is NP-complete. \square

Theorem 3. A communication graph G with a valid compact wakeup scheduling has an interval edge coloring and belongs to Class 1 graphs.

Proof. If graph G has a valid compact wakeup scheduling, any node v_i in G can wake up once to communicate with all its neighbors. Each two-way communication link can be colored with one color, and then the links incident to one node are assigned consecutive colors. Thus, graph G has an interval edge coloring. According to [21], graph with an interval edge coloring belongs to Class 1 graphs where the edge chromatic number is equal to the maximum degree Δ of graph G . Therefore, graph G is a Class 1 graph. \square

Unfortunately, the converse proposition is not true. The graph in Figure 3(a) belongs to Class 1 graphs, but has no valid interval edge coloring, and thus it has no valid compact wakeup schedulings. The Class 1 graphs even with valid interval edge colorings may not have valid compact wakeup schedulings. For example, the graph in Figure 3(b) has an interval edge-coloring, but all valid interval edge colorings could not avoid the hidden terminal problem. Thus, graphs with valid compact wakeup schedulings are a proper subset

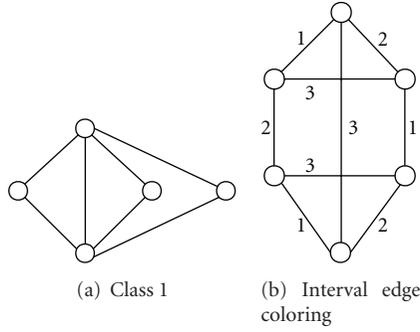


FIGURE 3: Class 1 graphs without valid compact wakeup schedulings.

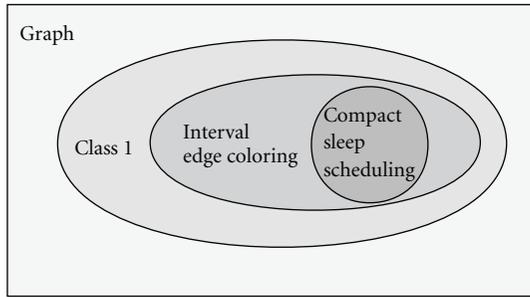


FIGURE 4: The relationship among Class 1 graphs, graphs with valid interval edge colorings and graphs with valid compact wakeup schedulings.

of graphs with valid interval edge colorings, and also a proper subset of Class 1 graphs, as shown in Figure 4.

Since not all communication graphs have valid compact wakeup schedulings and the problem of deciding whether a valid scheduling exists for an arbitrary graph is NP-complete, we will focus on particular graphs, such as tree and grid topologies. Interestingly and surprisingly, we can obtain polynomial-time algorithms using the optimum number of time slots in a period. By minimizing the number of time slots, the overall network throughput can be maximized.

3.3. Direction of Transmission Assignment in WSNs. In the link scheduling in WSNs, each edge in the communication graph has two transmission links: one is upload link, and the other one is downlowd link. We can easily find an edge coloring of a communication graph using $\Delta + 1$ colors [30], but how can this coloring be used to assign time slots to each transmission link? In [18], each color is mapped to two unique time slots and each transmission link is assigned a time slot according to the direction of transmission assignment (i.e., which end node of edge e will transmit or receive). Both the hidden terminal problem and the exposed terminal problem can be avoided. When the topologies are acyclic, the overall scheduling requires at most $2(\Delta + 1)$ time slots, where Δ is the maximum degree of a graph. When the topologies have cycles, additional time slots may be needed.

In this paper, the transmitter is marked with a sign “+” and the receiver is marked with a sign “-”. Given a coloring

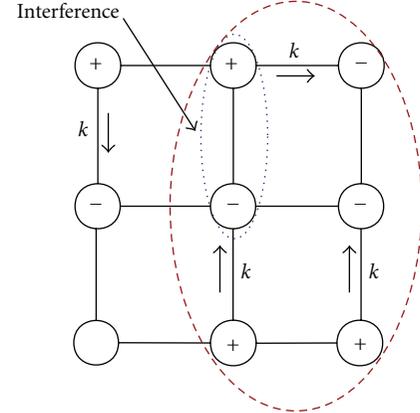


FIGURE 5: Cycle that has odd number of edges with color k cannot be assigned a valid direction of transmission.

of graph G and a color k , a subgraph $G^k = (V^k, E^k)$ is defined as follows. (a) V^k is the set of vertices incident to the edges colored with k . (b) E^k is the set of edges with both end vertices in V^k . When a node is assigned a sign “-”, the only neighbor assigned a sign “+” in G^k is the neighbor incident to the edge colored with k , and the other neighbors in G^k are individually assigned a sign “-”. Then, nodes incident to an edge colored with k always have an opposite sign, and nodes incident to an edge colored with other colors have the same sign. Algorithm 1, based on Depth First Search (DFS), can provide a *valid* direction of transmission assignment to each edge in G^k after a valid edge coloring is obtained in acyclic topologies. Such an assignment enables one-way communication. We can reverse the direction of transmission assignment along each edge to support bidirectional communication, and then each edge is assigned two time slots.

Gandham et al. [18] prove that a valid direction of transmission assignment exists in acyclic topologies (e.g., tree graphs). If a valid edge coloring is obtained in the topologies which are not acyclic (e.g., grid graphs), a valid direction of transmission assignment may not exist due to the hidden terminal problem, as shown in Figure 5. Interestingly, Gandham et al. [18] also prove that all the nodes in a cycle of G^k can be given a valid sign “+” or “-” if and only if there are an even number of edges with color k in the cycle.

4. Compact Wakeup Scheduling Algorithms

In this section, we propose polynomial-time algorithms to produce valid compact wakeup schedulings for tree and grid topologies, which are commonly used in WSNs [31–35].

4.1. Trees. To obtain a valid compact wakeup scheduling of a tree, we first obtain an interval edge coloring of a tree then try to assign time slots to each edge and make it interference-free.

If graph G is a tree of degree Δ , we could get an interval edge coloring with Δ colors for G using Algorithm 2 [26]: we first color any edge with 1, then find an uncolored

Input: A subgraph $G^k = (V^k, E^k)$.
Output: A valid direction of transmission assignment.

- (1) Start by visiting any node in V^k , and assign a sign “+” to it.
- (2) Initiate a Depth First Search (DFS) procedure.
- (3) **while** there are unvisited nodes **do**
- (4) Let edge e be traversed from a visited node v_i to an unvisited node v_j using the DFS procedure.
- (5) **if** e is colored with k **then**
- (6) Assign v_j the sign opposite to v_i .
- (7) **else**
- (8) Assign v_j the sign same to v_i .
- (9) **end if**
- (10) **end while**

ALGORITHM 1: DFS-based sign assignment algorithm [18].

Input: A tree $G = (V, E)$.
Output: A valid interval edge-coloring with Δ colors.

- (1) Color any edge with 1.
- (2) **while** there are uncolored edges **do**
- (3) Find an uncolored edge e whose end vertex v is adjacent to an already colored edge. Let $\{a, \dots, b\}$ be the interval of colors assigned to v .
- (4) **if** $a > 1$ **then**
- (5) Color edge e with $a - 1$.
- (6) **else**
- (7) Color edge e with $b + 1$.
- (8) **end if**
- (9) **end while**

ALGORITHM 2: Interval edge coloring of a tree [26].

Input: A tree $G = (V, E)$.
Output: A valid compact wakeup scheduling.

- (1) Use Algorithm 2 to obtain a valid interval edge-coloring with Δ colors for G .
- (2) **for** $k = 1$ to Δ **do**
- (3) Map color k to two consecutive time slots $\{2k - 1, 2k\}$.
- (4) Use Algorithm 1 to determine a valid direction of transmission assignment for time slot $2k - 1$.
- (5) Reverse the direction of transmission along each edge to obtain the other assignment for time slot $2k$.
- (6) **end for**

ALGORITHM 3: Compact wakeup scheduling of a tree.

edge e adjacent to an already colored edge, and assign e with a consecutive color until all the edges are colored. In the coloring process, when coloring a new uncolored edge, the consecutiveness of edge coloring remains invariant, and the edges already colored form a consecutively colored subgraph. After all edges are colored, we could get an interval edge coloring and the total number of colors assigned is Δ .

We now describe how the interval edge coloring is used to assign time slots to each edge in Algorithm 3. The idea

is to map color k to two consecutive time slots $\{2k - 1, 2k\}$, and use Algorithm 1 to determine a valid direction of transmission assignment for time slot $2k - 1$, and then reverse the direction of transmission along each edge to obtain the other assignment for time slot $2k$.

In Figure 6, links l_{ab} and l_{ce} are assigned the same color “1” in the interval edge coloring, while time slot t_{s_1} and t_{s_2} are allocated for color “1”. If time slot t_{s_1} is assigned in the directions of transmission as shown in Figure 6(a),

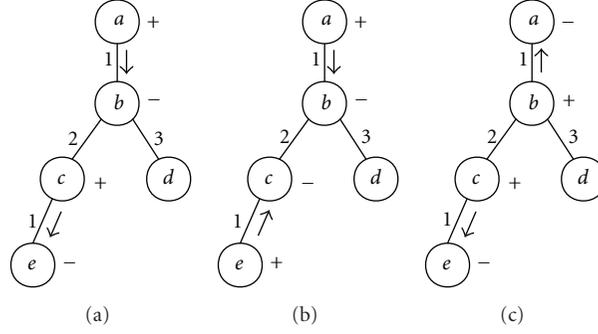


FIGURE 6: Compact wakeup scheduling of a tree: (a) hidden terminal problem, (b) avoid the hidden terminal problem, (c) avoid the exposed terminal problem.

the hidden terminal problem would happen because the reception at node v_b is garbled due to the collision of transmission from nodes v_a and v_c . Alternatively, if time slot ts_1 is assigned in the directions of transmission as shown in Figure 6(b), the hidden terminal problem could be avoided. Similarly, time slot ts_2 is assigned in the reverse directions of transmission as shown in Figure 6(c). Inspired by this, we should determine the directions of transmission along each link carefully to avoid the hidden terminal problem, that is, determine a node when to transmit and when to receive.

A tree does not have any cycles, and thus it is always possible to obtain a valid compact wakeup scheduling. Algorithm 1, based on Depth First Search (DFS), can provide a valid direction of transmission assignment to G^k . Note that the time slot assignment also avoids the exposed terminal problem [29], as shown in Figure 6(c).

Definition 4. The span of a valid compact wakeup scheduling of graph G is the number of colors assigned. The minimum and maximum span over all valid compact wakeup schedulings of G are denoted by $\chi_{cw}(G)$ and $\zeta_{cw}(G)$, respectively.

As any valid coloring in a tree requires at least Δ colors and an interval edge coloring can be obtained using Δ colors, $\chi_{cw}(G)$ is equal to Δ . Then, the number of time slots assigned in the compact wakeup scheduling is 2Δ , which is the optimum number of time slots. Algorithm 3 describes the compact wakeup scheduling for trees. Both the interval edge coloring of a tree and the time slot assignments can be obtained using $O(n)$, where n is the number of vertices in a tree. Thus, the algorithm to produce a valid compact wakeup scheduling for trees is polynomial time.

4.2. Grid Graphs. A $\mathcal{V} \times \mathcal{H}$ grid graph ($3 \leq \mathcal{V} \leq \mathcal{H}$) is a square lattice graph composed of $\mathcal{V}\mathcal{H}$ vertices. The grid graph has \mathcal{H} vertical paths and \mathcal{V} horizontal paths, where each vertical path consists of \mathcal{V} vertices and each horizontal path consists of \mathcal{H} vertices.

Definition 5. In a $\mathcal{V} \times \mathcal{H}$ grid graph, \bar{V}_{ij} ($1 \leq i \leq \mathcal{V} - 1$, $1 \leq j \leq \mathcal{H}$) denotes the i th vertical edge in the j th vertical path, and \bar{H}_{ij} ($1 \leq i \leq \mathcal{V}$, $1 \leq j \leq \mathcal{H} - 1$) denotes the j th horizontal edge in the i th horizontal path.

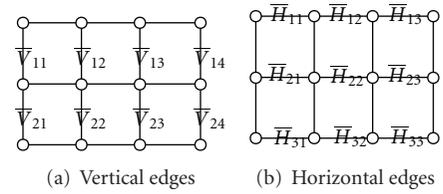


FIGURE 7: Vertical and horizontal edges in grid graphs.

Sample grids with labeled vertical and horizontal edges are illustrated in Figures 7(a) and 7(b). \bar{V}_{ij} is called *parallel* to \bar{V}_{mn} if $i = m$, and \bar{H}_{ij} is called *parallel* to \bar{H}_{mn} if $j = n$. For example, \bar{H}_{11} , \bar{H}_{21} , and \bar{H}_{31} are parallel in Figure 7(b).

Grid graphs can be consecutively colored with Δ colors, and one interval edge-coloring approach is given as follows. for a $\mathcal{V} \times \mathcal{H}$ grid graph, let c be a consecutive coloring of each horizontal path with colors 2 and 3. For each $i = 1, 2, \dots, \mathcal{V}$, we color the edges of i th horizontal path according to c . Let $\{a, \dots, b\}$ be an interval of colors assigned at each vertex in the corresponding horizontal path, then edge \bar{V}_{1j} is colored with $a - 1$, \bar{V}_{2j} with $b + 1$, \bar{V}_{3j} with $a - 1$, and so forth, where $1 \leq j \leq \mathcal{H}$. By repeating this for all edges, we could obtain an interval edge coloring of G , and a sample of the edge coloring is shown in Figure 8(a).

A valid direction of transmission assignment can be obtained to avoid the hidden terminal problem using Algorithm 1 in acyclic subgraphs G^k . But grid graphs contain cycles, and a valid assignment does not exist if we use the interval edge coloring approach above. For example, this edge coloring cannot avoid the hidden terminal problem as shown in Figure 8(b). Interestingly, Gandham et al. [18] prove that all the nodes in a cycle of G^k can be given a valid sign “+” or “-” if and only if there are an even number of edges with color k in the cycle.

If the edges colored with “3” in the cycle of Figure 8(b) are assigned with other colors, the consecutiveness of the colors assigned to the edges incident to one node cannot be held. Our solution for a grid graph first considers the property of the hidden terminal problem in the grid graph, and then deals with the consecutiveness of the edge-coloring. Our key results for grid graphs are summarized below.

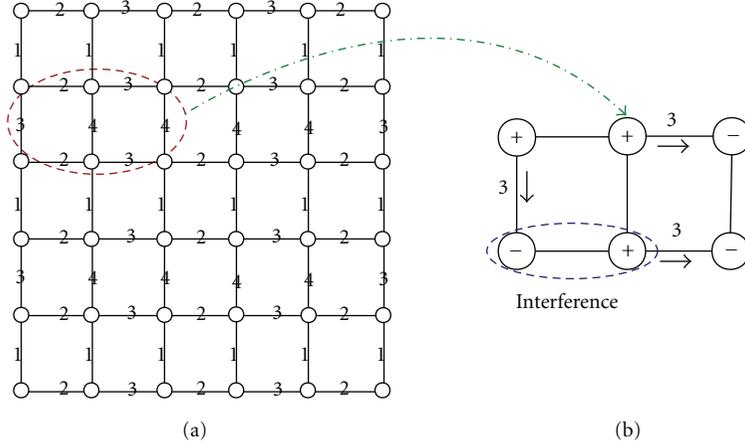


FIGURE 8: An interval edge coloring of a grid graph: (a) interval edge coloring, (b) hidden terminal problem.

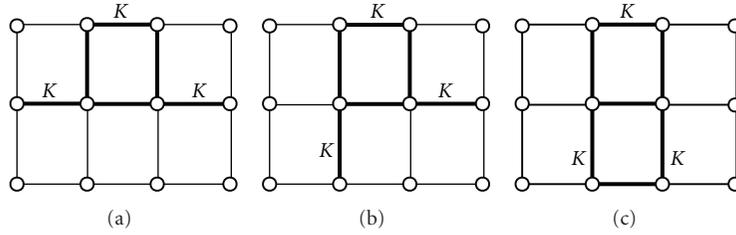


FIGURE 9: Invalid colorings in the compact wakeup scheduling.

- (1) We obtain an interval edge coloring according to the parity of \mathcal{V} and \mathcal{H} .
 - (i) If both \mathcal{V} and \mathcal{H} are even, $\chi_{\text{cw}}(G) = 4$.
 - (ii) If one of \mathcal{V} and \mathcal{H} is even and the other is odd, $\chi_{\text{cw}}(G) = 5$.
 - (iii) If both \mathcal{V} and \mathcal{H} are odd, $\chi_{\text{cw}}(G) = 6$.
- (2) We point out all the possible coloring patterns.
- (3) We give the upper bound of the compact wakeup scheduling.

Definition 6. In a grid graph, the maximum degree of vertices is $\Delta = 4$. The vertices of degree 4 are *inner vertices*, the vertices of degree 2 or 3 are *boundary vertices*, the edges incident to at least one inner vertex are *inner edges*, and the edges incident to two boundary vertices are *boundary edges*.

Definition 7. In the compact wakeup scheduling of a grid graph, the colors assigned to the inner edges incident to an inner vertex form an interval of 4 integers. When the total number of colors assigned is less than 8, certain color must appear in one of the inner edges and this color is referred to as a *critical color*.

In grid graphs, if the total number of colors assigned is M ($4 \leq M \leq 7$), then the number of critical colors is $8 - M$. For example, if $M = 4$, the set of critical colors is $\{1, 2, 3, 4\}$; if $M = 5$, the set of critical colors is $\{2, 3, 4\}$; if $M = 6$, the set of critical colors is $\{3, 4\}$.

Lemma 8. If K is a critical color assigned to an inner edge e incident to two inner vertices in the compact wakeup scheduling of a grid graph, the inner edges parallel to e are all colored with K .

Proof. Without loss of generality, we assume that an inner horizontal edge \bar{H}_{ij} ($2 \leq i \leq \mathcal{V}-1$, $2 \leq j \leq \mathcal{H}-2$) is colored with K in a $\mathcal{V} \times \mathcal{H}$ grid graph. The cases of colorings shown in Figure 9 would lead to odd number of edges with color K in subgraph G^K (see the thick lines in Figure 9), and no feasible direction of transmission can be obtained. Since K is a critical color, $\bar{H}_{(i+1)j}$ ($i+1 \leq \mathcal{V}-1$) must be colored with K . By applying recursion, the horizontal edges \bar{H}_{mj} ($2 \leq m \leq \mathcal{V}-1$) are in a parallel pattern, as shown in Figure 10(a). \square

Lemma 9. If K is a critical color, the inner edges colored with K are in an interlined pattern.

Proof. Without loss of generality, we assume an inner horizontal edge \bar{H}_{ij} ($2 \leq i \leq \mathcal{V}-1$, $2 \leq j \leq \mathcal{H}-2$) is colored with K in a $\mathcal{V} \times \mathcal{H}$ grid graph. According to Lemma 8, the horizontal edges \bar{H}_{mj} ($2 \leq m \leq \mathcal{V}-1$) are colored with K . Let $k = j + 2$ ($k \leq \mathcal{H}-2$), \bar{H}_{3k} must be colored with K , since K is a critical color and $\bar{H}_{3(k-1)}$, \bar{V}_{2k} as well as \bar{V}_{3k} cannot be colored with K . According to Lemma 8, the horizontal edges \bar{H}_{mk} ($2 \leq m \leq \mathcal{V}-1$) are colored with K , and the result still holds when $k = j - 2$ ($k \geq 2$). By applying recursion, the horizontal edges \bar{H}_{mk} ($k = j \pm 2n$, $n \in \mathbb{N}$, $2 \leq m \leq \mathcal{V}-1$, $2 \leq k \leq \mathcal{H}-2$) are colored with K . If $k = 1$ (or $\mathcal{H}-1$) and $k = j \pm 2n$, $n \in \mathbb{N}$, the horizontal edges

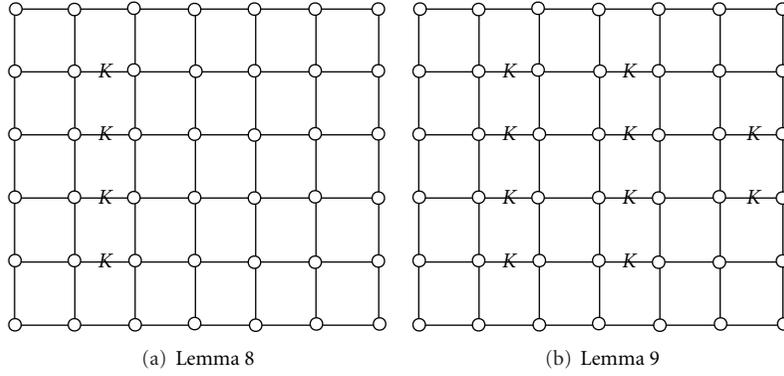
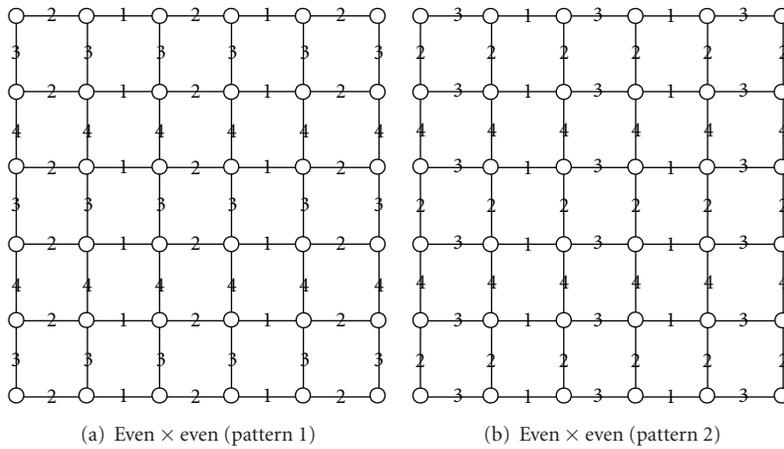


FIGURE 10: Coloring pattern in the compact wakeup scheduling.


 FIGURE 11: The coloring patterns in the compact wakeup scheduling (both \mathcal{V} and \mathcal{H} are even).

\bar{H}_{mk} ($3 \leq m \leq \mathcal{V} - 2$) are colored with K , since K is a critical color. Hence, the inner edges colored with a critical color are in an interlined pattern, as shown in Figure 10(b). \square

Theorem 10. *A $\mathcal{V} \times \mathcal{H}$ grid graph ($3 \leq \mathcal{V} \leq \mathcal{H}$, both \mathcal{V} and \mathcal{H} are even) can be consecutively colored with 4 colors in the compact wakeup scheduling, and the possible colorings must be the patterns as shown in Figure 11, and $\chi_{cw}(G) = 4$.*

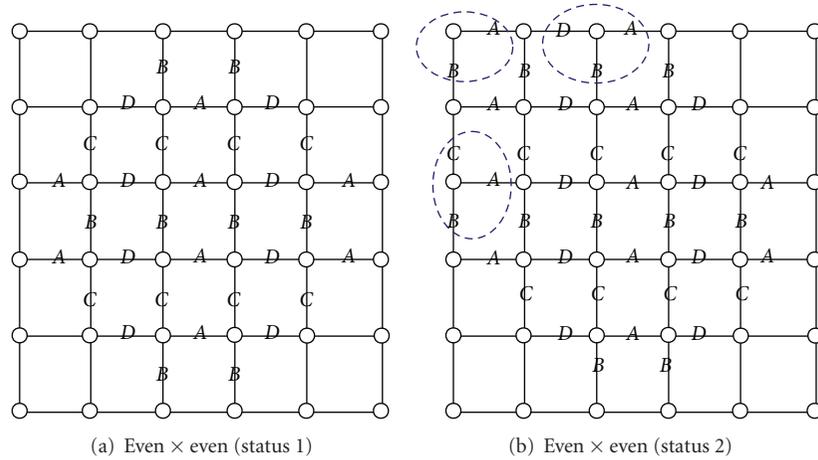
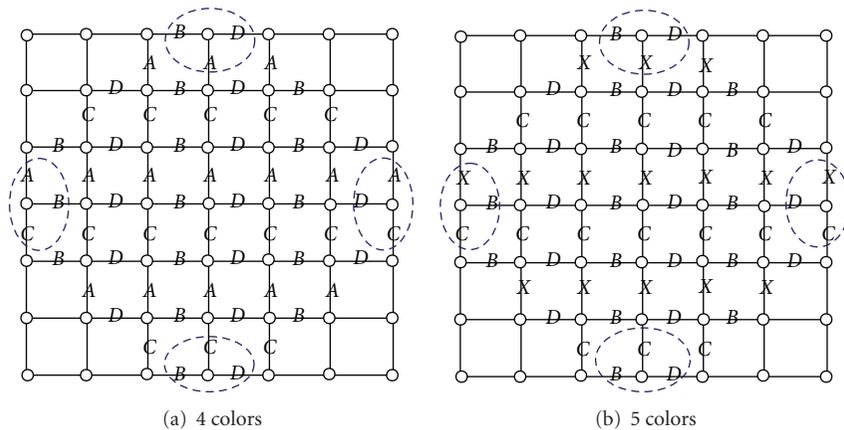
Proof. Figures 11(a) and 11(b) show two possible colorings in the compact wakeup scheduling, if both \mathcal{V} and \mathcal{H} are even. Since the edges with the same color are in a parallel and interlined pattern, there are an even number of edges with color k ($1 \leq k \leq 4$) in a cycle in the subgraph G^k and then the scheduling could avoid the hidden terminal problem. Therefore, $\chi_{cw}(G) = 4$.

As $\chi_{cw}(G)$ is equal to 4, we assume the four critical colors are A, B, C and D . According to Lemmas 8 and 9, A, B, C , and D are all in a parallel and interlined pattern shown as status 1 in Figure 12(a). To avoid the hidden terminal problem, \bar{H}_{13} cannot be colored with D or C and can only be colored with A . Then \bar{H}_{12} must be colored with D . Similarly, \bar{V}_{21} and \bar{V}_{31} are colored with C and B , respectively. Then \bar{H}_{11} , \bar{V}_{11} , \bar{H}_{21} , and \bar{V}_{12} are colored with A, B, A , and B , respectively. We can color other edges in a similar way. In status 2 shown in

Figure 12(b), we can see the color sets $\{A, B\}$, $\{A, B, D\}$, and $\{A, B, C\}$ must consist of consecutive numbers since these colors assigned to the edges incident to the vertices in the dashed circles must be consecutive. Therefore, $\{A, B, C\}$ and $\{A, B, D\}$ belong to $\{1, 2, 3\}$ and $\{2, 3, 4\}$, and $\{A, B\}$ belongs to $\{2, 3\}$. For C and D are symmetrical, we can get $C = 4$ and $D = 1$. For the case that $A = 2$ and $B = 3$, the coloring pattern is Figure 11(a). For the case that $A = 3$ and $B = 2$, the coloring pattern is Figure 11(b). \square

Lemma 11. *A $\mathcal{V} \times \mathcal{H}$ grid graph ($3 \leq \mathcal{V} \leq \mathcal{H}$, both \mathcal{V} and \mathcal{H} are odd) cannot be consecutively colored with 4 or 5 colors in the compact wakeup scheduling.*

Proof. (1) If the grid could be consecutively colored with 4 colors A, B, C , and D , the four colors belonging to $\{1, 2, 3, 4\}$ are all critical colors. For an inner vertex has 4 incident inner edges, the inner edges are colored with A, B, C , and D , respectively. According to Lemmas 8 and 9, A, B, C and D are all in a parallel and interlined pattern, and the coloring is shown in Figure 13(a). As the colors assigned to the edges incident to the vertices in the dashed circles must be consecutive, the color sets $\{A, B, C\}$, $\{B, C, D\}$, $\{A, C, D\}$, and $\{A, B, D\}$ must consist of three consecutive numbers. However, $\{1, 2, 3\}$ and $\{2, 3, 4\}$ are the only two possible

FIGURE 12: The coloring in the compact wakeup scheduling (both \mathcal{V} and \mathcal{H} are even).FIGURE 13: The colorings in the compact wakeup scheduling using 4 and 5 colors (both \mathcal{V} and \mathcal{H} are odd).

cases with three consecutive numbers, which leads to a contradiction.

(2) If the grid could be consecutively colored with 5 colors, B , C and D belonging to $\{2, 3, 4\}$ are critical colors and the noncritical color 1 or 5 is denoted by X . For an inner vertex has 3 incident critical inner edges, the inner edges are colored with B , C , and D , respectively. According to Lemmas 8 and 9, B , C , and D are all in a parallel and interlined pattern, and the coloring is shown in Figure 13(b). Since the colors assigned to the edges incident to the vertices in the dashed circles must be consecutive, the color sets $\{B, C, X\}$, $\{B, C, D\}$, $\{C, D, X\}$, and $\{B, D, X\}$ must consist of three consecutive numbers. However, $\{1, 2, 3\}$, $\{2, 3, 4\}$, and $\{3, 4, 5\}$ are the only three possible cases with three consecutive numbers, which leads to a contradiction. \square

Similarly, we could get the following lemma.

Lemma 12. *A $\mathcal{V} \times \mathcal{H}$ grid graph ($3 \leq \mathcal{V} \leq \mathcal{H}$, one of \mathcal{V} and \mathcal{H} is even and the other is odd) cannot be consecutively colored with 4 colors in the compact wakeup scheduling.*

Theorem 13. *A $\mathcal{V} \times \mathcal{H}$ grid graph ($3 \leq \mathcal{V} \leq \mathcal{H}$, one of \mathcal{V} and \mathcal{H} is even and the other is odd) can be consecutively colored with 5 colors in the compact wakeup scheduling, and the possible coloring must be the pattern as shown in Figure 14(a), and $\chi_{cw}(G) = 5$.*

Proof. Figure 14(b) shows a possible coloring in the compact wakeup scheduling by determining the colors for the rest uncolored edges in Figure 14(a), if one of \mathcal{V} and \mathcal{H} is even and the other is odd. Since the edges with the same color are in a parallel and interlined pattern, there are an even number of edges with color k ($1 \leq k \leq 5$) in a cycle in the subgraph G^k and then the scheduling could avoid the hidden terminal problem. By combining with Lemma 12, $\chi_{cw}(G) = 5$.

Since $\chi_{cw}(G)$ is equal to 5, we assume B , C , and D belonging to $\{2, 3, 4\}$ are critical colors and the noncritical color 1 or 5 is denoted by X . As an inner vertex has 3 incident critical inner edges, Figures 15(a), 15(c), and 15(d) are the possible coloring patterns.

Case 1. In status 1 shown in Figure 15(a), \bar{H}_{14} cannot be colored with B , C , or D and can only be colored with X .

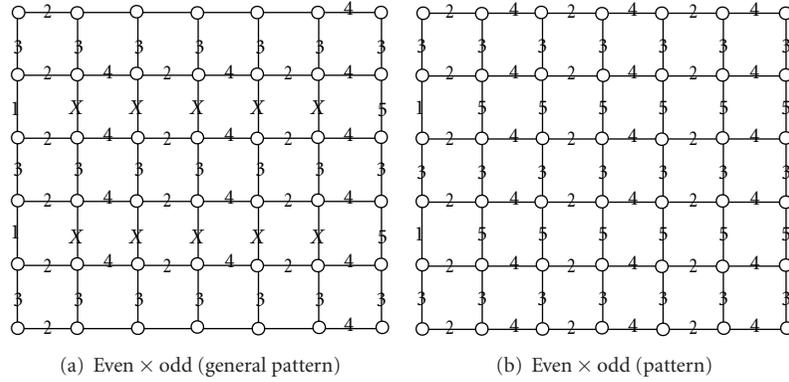


FIGURE 14: The general coloring pattern and coloring pattern in the compact wakeup scheduling (one of \mathcal{V} and \mathcal{H} is even and the other is odd. The uncolored edges depend on the edges colored with X , and $X = 1$ or 5).

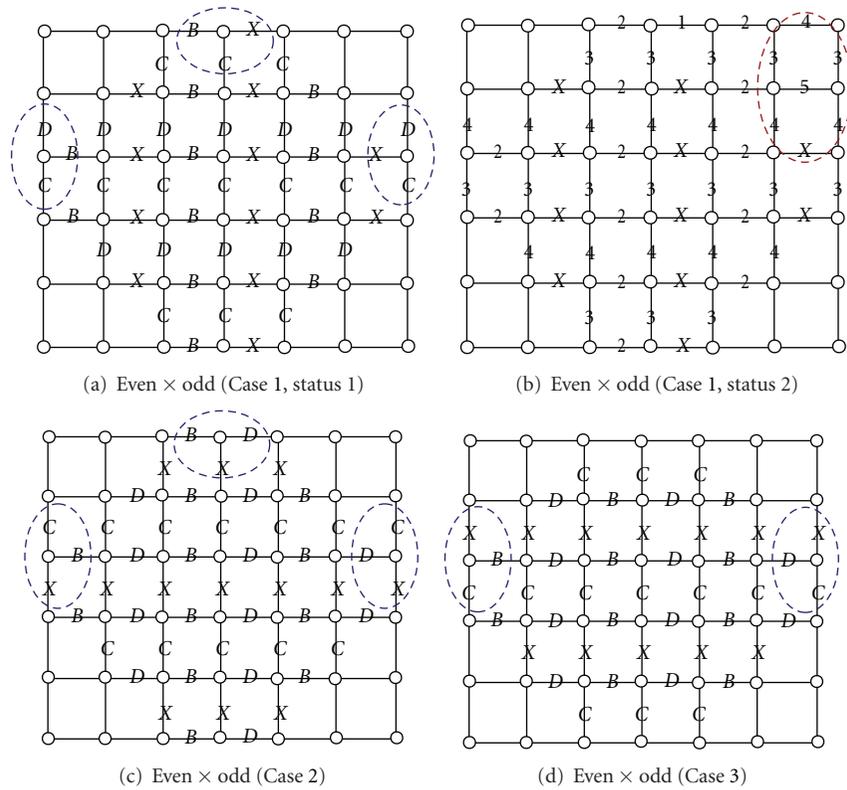


FIGURE 15: The colorings in the compact wakeup scheduling (one of \mathcal{V} and \mathcal{H} is even and the other is odd).

Then \bar{H}_{13} must be colored with B . Similarly, $\bar{V}_{21}, \bar{V}_{31}, \bar{V}_{27}, \bar{V}_{37}$ are colored with $D, C, D,$ and C , respectively. We can see that the color sets $\{B, C, X\}, \{B, C, D\},$ and $\{C, D, X\}$ must consist of consecutive numbers since these colors assigned to the edges incident to the vertices in the dashed circles must be consecutive. Then, $\{B, C, X\}$ and $\{C, D, X\}$ belong to $\{1, 2, 3\}$ and $\{3, 4, 5\}$. Then, we can get $C = 3$. If $B = 2$ and $D = 4, \bar{H}_{15}, \bar{V}_{16}, \bar{H}_{26},$ and \bar{V}_{17} are colored with $2, 3, 5,$ and 3 , respectively, shown as the status 2 in Figure 15(b). Then \bar{H}_{16} can only be colored with 4 , which leads to interferences in the dashed circle. Similarly, if $B = 4$ and $D = 2$, we cannot get

an interference-free scheduling either. Hence, the coloring pattern in Figure 15(a) is not valid.

Case 2. In Figure 15(c), \bar{H}_{14} cannot be colored with $B, C,$ or X and can only be colored with D . Then \bar{H}_{13} must be colored with B . Similarly, $\bar{V}_{21}, \bar{V}_{31}, \bar{V}_{27},$ and \bar{V}_{37} are colored with $C, X, C,$ and X , respectively. We can see that the color sets $\{B, D, X\}, \{B, C, X\},$ and $\{C, D, X\}$ must consist of consecutive numbers since these colors assigned to the edges incident to the vertices in the dashed circles must be consecutive. Moreover, $B, C, D \in \{2, 3, 4\}$ are consecutive.

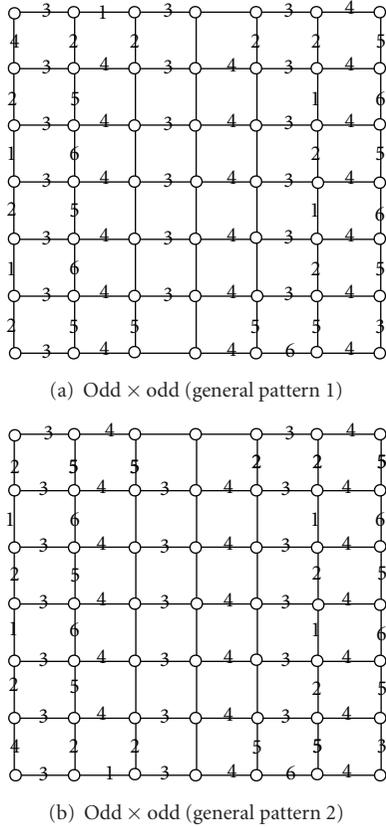


FIGURE 16: The general coloring patterns in the compact wakeup scheduling (both \mathcal{V} and \mathcal{H} are odd. The uncolored edges have various alternatives).

However, $\{1, 2, 3\}$, $\{2, 3, 4\}$, and $\{3, 4, 5\}$ are the only three possible cases with three consecutive numbers. Hence, the coloring pattern in Figure 15(c) is not valid.

Case 3. In Figure 15(d), \bar{V}_{21} cannot be colored with B , C , or D and can only be colored with X . Then \bar{V}_{31} must be colored with C . Similarly, \bar{V}_{27} and \bar{V}_{37} are colored with X and C , respectively. We can see that the color sets $\{B, C, X\}$ and $\{C, D, X\}$ must consist of consecutive numbers since these colors assigned to the edges incident to the vertices in the dashed circles must be consecutive. Then $C = 3$. For B and D are symmetrical, we can get $B = 2$ and $D = 4$. By assigning the possible colors in other edges, the coloring pattern in Figure 14(a) is obtained. \square

Theorem 14. A $\mathcal{V} \times \mathcal{H}$ grid graph ($3 \leq \mathcal{V} \leq \mathcal{H}$, both \mathcal{V} and \mathcal{H} are odd) can be consecutively colored with 6 colors in the compact wakeup scheduling, and the possible colorings must be the patterns as shown in Figure 16, and $\chi_{cw}(G) = 6$.

Proof. Figures 18(a) and 18(b) show two possible colorings in the compact wakeup scheduling by determining the colors for the rest uncolored edges in Figures 16(a) and 16(b), if both \mathcal{V} and \mathcal{H} are odd. Particularly, if $\mathcal{V} = 3$, the possible colorings are shown in Figures 17(a) and 17(b). Since there are even number of edges with color k ($1 \leq k \leq 6$)

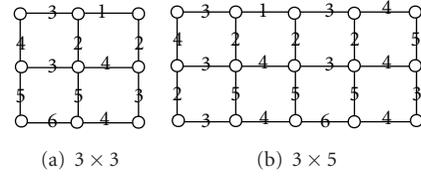


FIGURE 17: The coloring patterns in the compact wakeup scheduling ($\mathcal{V} = 3$ and \mathcal{H} is odd).

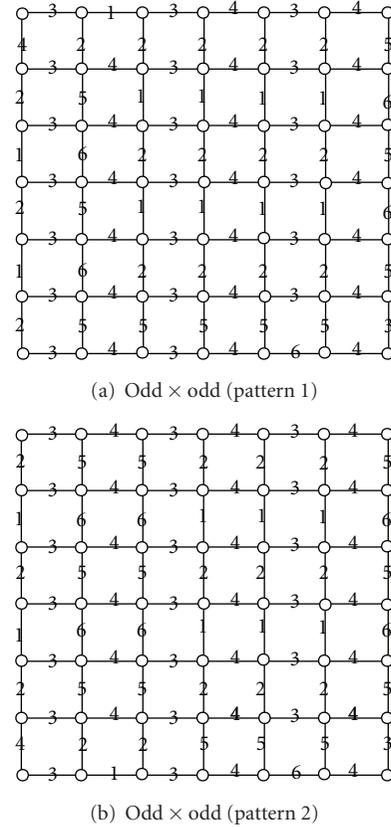
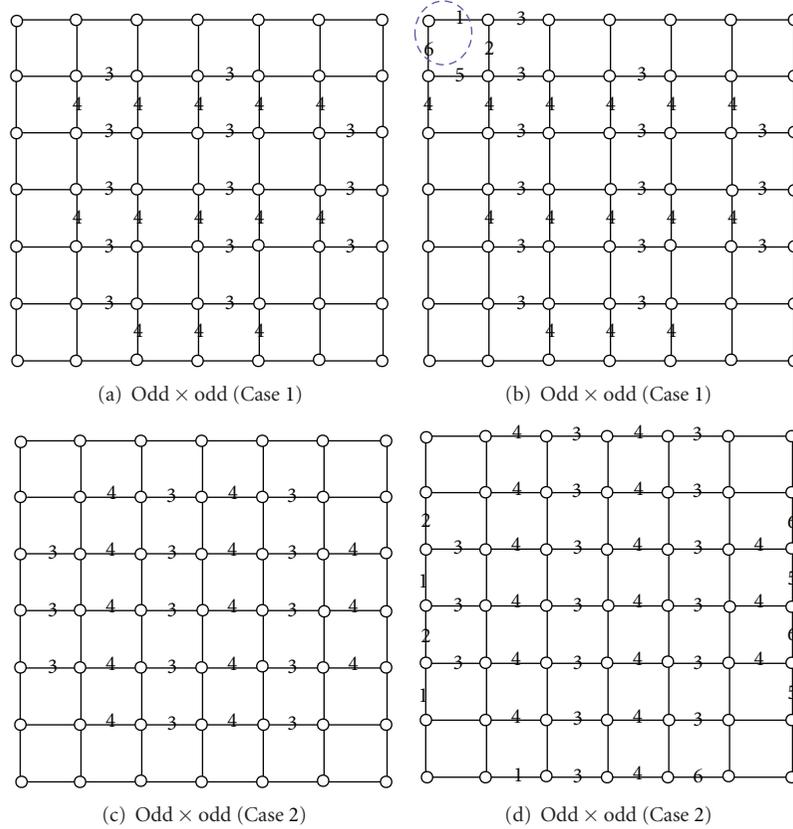


FIGURE 18: The coloring patterns in the compact wakeup scheduling (both \mathcal{V} and \mathcal{H} are odd).

in a circle in the subgraph G^k , then the scheduling could avoid the hidden terminal problem. According to Lemma 11, $\chi'(G) = 6$.

Since the grid graph can be consecutively colored with 6 colors, 3 and 4 are critical colors. For an inner vertex has two incident critical inner edges, Figures 19(a) and 19(c) are the possible coloring patterns.

Case 1. In Figure 19(a), \bar{V}_{21} , \bar{V}_{31} , and \bar{H}_{31} cannot be colored with 3, but can only be colored with $\{4, 5, 6\}$. For \bar{V}_{31} and \bar{H}_{31} cannot be colored with 4, \bar{V}_{21} must be colored with 4. Similarly, \bar{H}_{12} must be colored with 3. Then \bar{V}_{11} , \bar{V}_{21} , and \bar{H}_{21} must be colored with $\{4, 5, 6\}$, and \bar{H}_{11} , \bar{H}_{12} , and \bar{V}_{12} must be colored with $\{1, 2, 3\}$. For \bar{V}_{12} , \bar{V}_{22} , \bar{H}_{21} , and \bar{H}_{22} are consecutively colored, \bar{V}_{12} is colored with 2 and \bar{H}_{21} is colored with 5. Then, \bar{V}_{11} is colored with 6 and \bar{H}_{11} is colored with 1, which leads to an inconsecutive coloring, as shown in Figure 19(b). Hence, Figure 19(a) is not a possible coloring.


 FIGURE 19: The colorings in the compact wakeup scheduling (both \mathcal{V} and \mathcal{H} are odd).

Case 2. In Figure 19(c), \bar{V}_{21} , \bar{V}_{23} , and \bar{H}_{31} cannot be colored with 4, but can only be colored with $\{1, 2, 3\}$; \bar{V}_{27} , \bar{V}_{37} , and \bar{H}_{36} cannot be colored with 3, but can only be $\{4, 5, 6\}$. According to the symmetrical property, we suppose \bar{V}_{27} and \bar{V}_{37} are colored with 6 and 5, respectively. If \bar{V}_{21} is colored with 2 and \bar{V}_{31} is colored with 1, we can get Figure 19(d). By assigning the possible colors in other edges, the coloring pattern in Figure 16(a) is obtained. If \bar{V}_{21} is colored with 1 and \bar{V}_{31} is colored with 2, we can get the coloring pattern in Figure 16(b).

Hence, the possible colorings must be the patterns as shown in Figures 16(a) and 16(b), and $\chi'(G) = 6$. \square

Theorem 15. *In the compact wakeup scheduling of a $\mathcal{V} \times \mathcal{H}$ grid graph ($3 \leq \mathcal{V} \leq \mathcal{H}$), $2\mathcal{V} + 2\mathcal{H} - 6 \leq \zeta_{\text{cw}}(G) \leq (1/6)(13\mathcal{V} + 13\mathcal{H} - 8)$.*

Proof. Lower bound: we can get a valid consecutive edge coloring with a valid direction of transmission assignment in the compact wakeup scheduling using $2\mathcal{V} + 2\mathcal{H} - 6$ colors. For example, the number of colors assigned is $22 = 2 \times 7 + 2 \times 7 - 6$ in a 7×7 grid as shown in Figure 20(a).

Upper bound: for a consecutive edge coloring in the compact wakeup scheduling of a grid graph G , the difference in colors of edges incident to a node v cannot exceed $\deg(v_i) - 1$. Suppose that v_1, v_2, \dots, v_m is the vertex sequence of a path connecting edges with extremal colors, we could

get $\zeta_{\text{cw}}(G) \leq 1 + \sum_{i=1}^m (\deg(v_i) - 1)$. We suppose vertices A and B are on the path connecting edges with minimum and maximum colors, respectively, as shown in Figure 20(b). We assume vertex A is on the common point of $\bar{H}_{(b+1)(a+1)}$ and $\bar{V}_{(b+1)(a+1)}$, and vertex B is on the common point of $\bar{H}_{(\mathcal{V}-m)(\mathcal{H}-1-n)}$ and $\bar{V}_{(\mathcal{V}-1-m)(\mathcal{H}-n)}$. We can get $\zeta_{\text{cw}}(G) \leq 1 + 3(\mathcal{H} - 1 - a - n + 1) + 3(\mathcal{V} - 1 - b - m) = 3(\mathcal{V} + \mathcal{H} - a - b - m - n) - 2$ using route 1. We have also known $\zeta_{\text{cw}}(G) \geq 2\mathcal{V} + 2\mathcal{H} - 6$. $3(\mathcal{V} + \mathcal{H} - a - b - m - n) - 2$ should be no less than $2\mathcal{V} + 2\mathcal{H} - 6$. Otherwise, $2\mathcal{V} + 2\mathcal{H} - 6$ should also be the upper bound. Then we get $\mathcal{V} + \mathcal{H} + 4 \geq 3(a + b + m + n)$. Without loss of generality, we assume $a + m \geq b + n$. Then, $b + n \leq (1/6)(\mathcal{V} + \mathcal{H} + 4)$. We can also get $\zeta_{\text{cw}}(G) \leq 1 + 3b + 2(\mathcal{H} - a - 1) + 1 + 2(\mathcal{V} - m - 1) + 3n = 2(\mathcal{V} + \mathcal{H} - a - m) + 3b + 3n - 2$ using route 2. Then, $\zeta_{\text{cw}}(G) = 2(\mathcal{V} + \mathcal{H}) + 3(b + n) - 2(a + m) - 2 \leq 2(\mathcal{V} + \mathcal{H}) + b + n - 2 \leq (1/6)(13\mathcal{V} + 13\mathcal{H} - 8)$.

Thus, $\zeta_{\text{cw}}(G)$ is bounded by $2\mathcal{V} + 2\mathcal{H} - 6$ and $(1/6)(13\mathcal{V} + 13\mathcal{H} - 8)$. \square

According to Theorems 10, 13, and 14, the number of time slots assigned is optimum. If both \mathcal{V} and \mathcal{H} are even, the number of time slots assigned in a period is $4 \times 2 = 8$ in a $\mathcal{V} \times \mathcal{H}$ grid graph. If one of \mathcal{V} and \mathcal{H} is even and the other is odd, the number of time slots is $5 \times 2 = 10$. If both \mathcal{V} and \mathcal{H} are odd, the number of time slots is $6 \times 2 = 12$. Algorithms 4 and 5 describe the interval edge coloring and

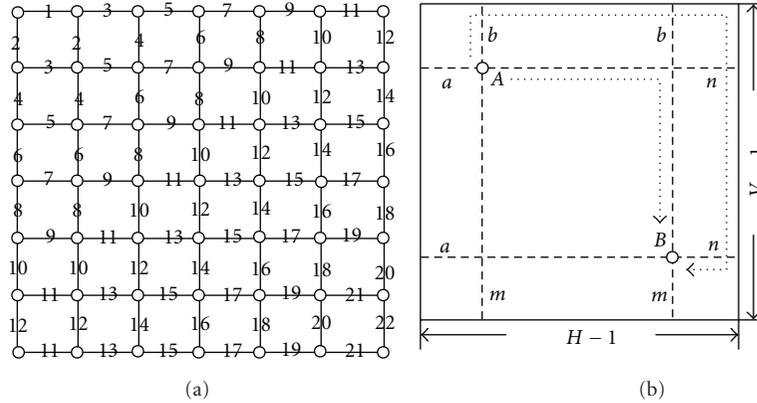


FIGURE 20: $\zeta_{cw}(G)$ in the compact wakeup scheduling: (a) lower bound of $\zeta_{cw}(G)$, (b) upper bound of $\zeta_{cw}(G)$.

Input: A $\mathcal{V} \times \mathcal{H}$ grid graph G ($3 \leq \mathcal{V} \leq \mathcal{H}$).
Output: A valid interval edge-coloring with $\chi_{cw}(G)$ colors.
(1) Decide the parity of \mathcal{V} and \mathcal{H} (even or odd).
(2) **if** both \mathcal{V} and \mathcal{H} are even **then**
(3) Color G using the pattern in Figure 11.
(4) **else if** one of \mathcal{V} and \mathcal{H} is even and one is odd **then**
(5) Color G using the pattern in Figure 14(b).
(6) **else**
(7) Color G using the pattern in Figure 18.
(8) **end if**

ALGORITHM 4: Interval edge coloring of a grid graph.

compact wakeup scheduling of a grid graph, respectively. The complexity of the compact wakeup scheduling of a grid graph is $O(n)$.

5. Performance Evaluation

In this section, we study the performance of the compact wakeup scheduling of trees and grid graphs, and we also compare our algorithms with the *degree-based heuristic* in [10] and the *contiguous link scheduling* in [13]. The performance metrics used in the evaluation are the transient energy consumption and the waiting period. The total energy consumption is an important metric in WSNs, but the energy consumption except the transient energy consumption is the same among the three schemes under identical traffic conditions, so we will only focus on the transient energy consumption. The waiting period is defined as the total time a node stays in the waiting status from the first neighbor waking up to the last neighbor waking up as the node waits for gathering the information from all its neighbors. The waiting period reflects the extra delay caused by the node if it stays in the sleep state for the wakeup of neighbors.

We adopt the following parameters in our simulation: the transient energy to activate a sensor is 17μ [12], a time slot is 0.1 second, a scheduling period T is 10 seconds (=100 time slots), and the network operating time is 1 day. In

the tree construction of n nodes, the number of children nodes of each sensor is randomly set from 1 to 4. The root node first determines its children nodes, and then each child node determines its children nodes, and so on until the total number of nodes in the tree reaches n . In the tree construction, we vary n from 20 to 120 in steps of 20, 10 trees are generated, and the average performance over all these trees is reported. For the grid graph, we use square grid graphs, where $\mathcal{V} = \mathcal{H}$. In the grid graph construction, we vary \mathcal{V} from 2 to 12 in steps of 2.

Figure 21 shows the total transient energy consumption of the following schemes: degree-based heuristic (degree-based), contiguous link scheduling (contiguous), and compact wakeup scheduling (compact). In both the tree and grid topologies, the transient energy consumption increases as the number of nodes increases. The energy consumption in the compact wakeup scheduling is the smallest among the three schemes, for the frequency of state transitions is minimized in the scheduling. As shown in Figure 21(a), compact wakeup scheduling reduces the energy consumption significantly by approximately 50% as compared to that in the degree-based heuristic and about 35% as compared to that in the contiguous link scheduling.

Figure 22 shows the total waiting period increases as the number of nodes increases in the degree-based heuristic and contiguous link scheduling, while the waiting period is zero in the compact wakeup scheduling. With smaller waiting periods, it would be faster for nodes to gather the information from their neighbors, thus reducing network delay.

We summarize observations from the simulation results as follows. (1) The waiting period of trees and grid graphs with valid compact wakeup scheduling is zero. (2) Compact wakeup scheduling can significantly reduce network delay and energy consumption.

6. Conclusion and Future Work

In this paper, we address a new interference-free TDMA wakeup scheduling problem in WSNs, called compact wakeup scheduling. In the scheduling, a node needs to

Input: A $\mathcal{V} \times \mathcal{H}$ grid graph G ($3 \leq \mathcal{V} \leq \mathcal{H}$).
Output: A valid compact wakeup scheduling.
 (1) Use Algorithm 4 to obtain a valid interval edge-coloring with $\chi_{cw}(G)$ colors for G .
 (2) **for** $k = 1$ to $\chi_{cw}(G)$ **do**
 (3) Map color k to two consecutive time slots $\{2k - 1, 2k\}$.
 (4) Use Algorithm 1 to determine a valid direction of transmission assignment for time slot $2k - 1$.
 (5) Reverse the direction of transmission along each edge to obtain the other assignment for time slot $2k$.
 (6) **end for**

ALGORITHM 5: Compact wakeup scheduling of a grid graph.

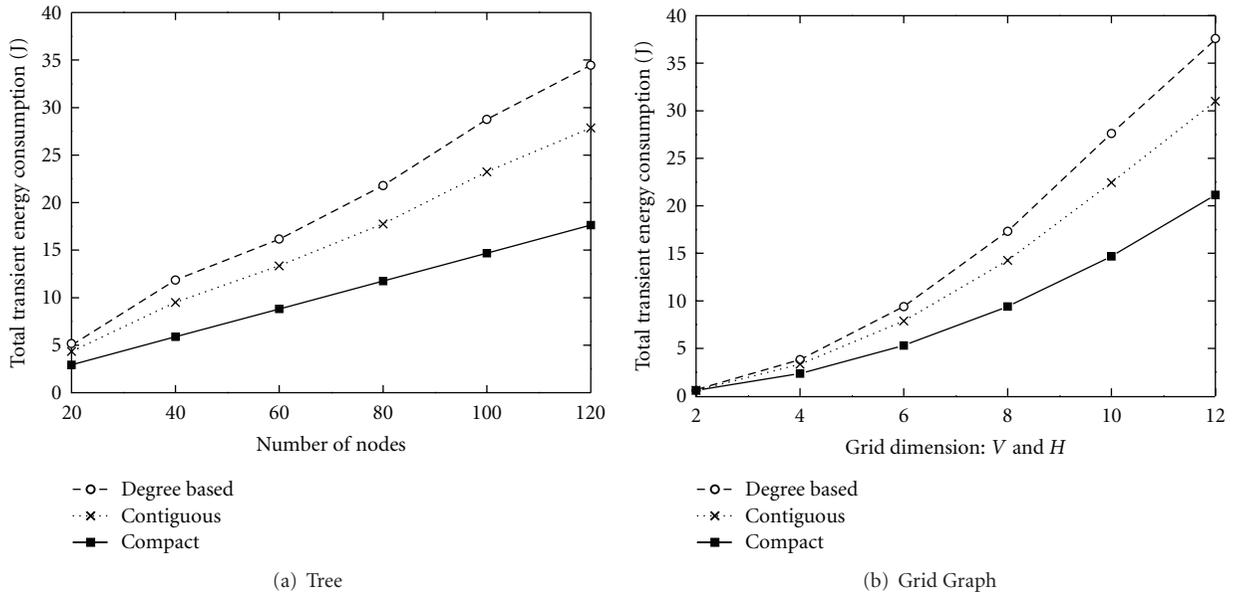


FIGURE 21: Transient energy consumption.

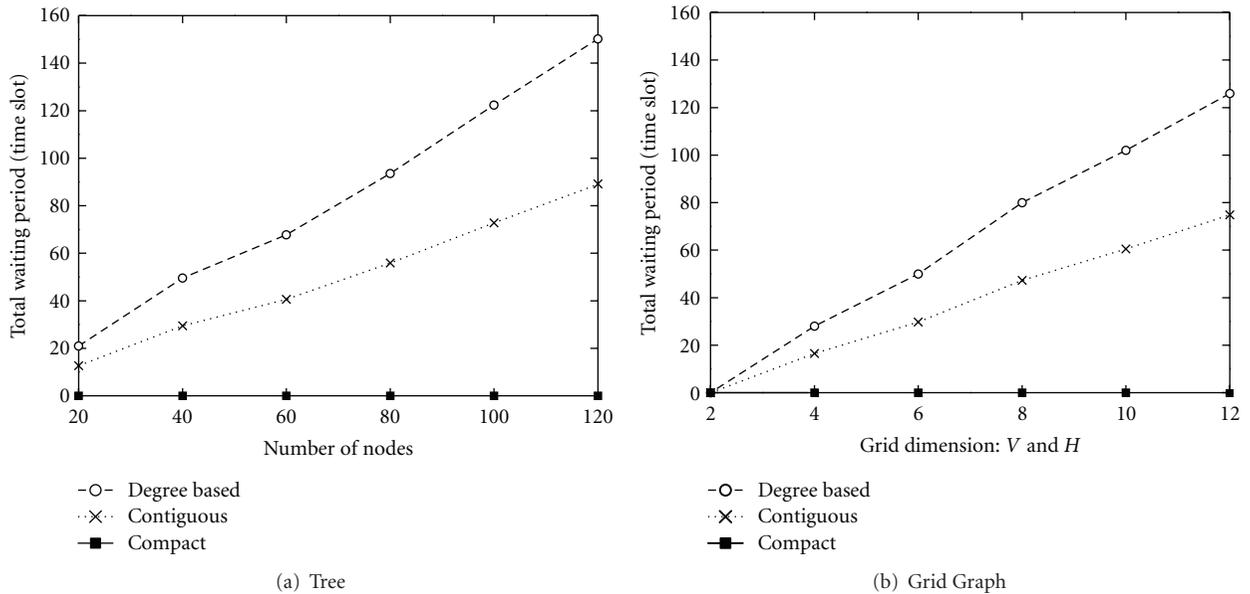


FIGURE 22: Waiting period.

wake up only once to communicate bidirectionally with all its neighbors, thus reducing the time overhead and energy cost in the state transitions. We propose polynomial-time algorithms to achieve the optimum number of time slots assigned in a period for trees and grid graphs. In grid graphs, we point out all the possible coloring patterns and give the lower bound as well as the upper bound of the compact wakeup scheduling. In the process of time slot assignments, both the hidden terminal and exposed terminal problems can be avoided. The simulation results corroborate the theoretical analysis and show the efficiency of compact wakeup scheduling.

In our future work, we will consider the heterogeneous network model and try to obtain efficient algorithms for other kinds of network topologies with valid compact wakeup schedulings. Another challenging topic is to find the scheduling with the minimum waiting period if a valid compact wakeup scheduling does not exist for a given topology.

Acknowledgments

This work is supported in part by Grants PolyU 5236/06E, PolyU 5243/08E, A-PJ16, and 1-ZV5N.

References

- [1] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '02)*, June 2002.
- [2] T. Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the First International Conference on Embedded Networked Sensor Systems (SenSys '03)*, November 2003.
- [3] M. A. Ameen, S. M. R. Islam, and K. Kwak, "Energy saving mechanisms for MAC protocols in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2010, Article ID 163413, 16 pages, 2010.
- [4] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 53–62, May 2008.
- [5] I. Bekmezci and F. Alagoz, "Energy efficient, delay sensitive, fault tolerant wireless sensor network for military monitoring," *International Journal of Distributed Sensor Networks*, vol. 5, no. 6, pp. 729–747, 2009.
- [6] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," in *Proceedings of the First International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 181–192, November 2003.
- [7] O. Chipara, C. Lu, and J. Stankovic, "Dynamic conflict-free query scheduling for wireless sensor networks," in *Proceedings of the 14th IEEE International Conference on Network Protocols (ICNP '06)*, pp. 321–331, November 2006.
- [8] I. Rhee, A. Warrior, J. Min, and L. Xu, "DRAND: distributed randomized TDMA scheduling for wireless ad-hoc networks," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '06)*, pp. 190–201, May 2006.
- [9] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup scheduling in wireless sensor networks," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, (MOBIHOC '06)*, pp. 322–333, May 2006.
- [10] W. Wang, Y. Wang, X. Y. Li, W. Z. Song, and O. Frieder, "Efficient interference-aware TDMA link scheduling for static wireless networks," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06)*, pp. 262–273, September 2006.
- [11] A. Wang, S. Cho, C. Sodini, and A. Chandrakasan, "Energy efficient modulation and MAC for asymmetric RF microsensor systems," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED '01)*, 2001.
- [12] Sentilla Corporation, "Tmote Sky Datasheet," <http://www.sentilla.com/files/pdf/eol/tmote-sky-datasheet.pdf>.
- [13] J. Ma, W. Lou, Y. Wu, X. Y. Li, and G. Chen, "Energy efficient TDMA sleep scheduling in wireless sensor networks," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 630–638, April 2009.
- [14] M. Li and Y. Liu, "Underground structure monitoring with wireless sensor networks," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 69–78, April 2007.
- [15] W. Song, R. Huang, B. Shirazi, and B. LaHusen, "TreeMAC: localized TDMA MAC protocol for real-time high-data-rate sensor networks," in *Proceedings of the IEEE Pervasive Computing and Communication Conference (PerCom '09)*, 2009.
- [16] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "PW-MAC: an energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '11)*, 2011.
- [17] S. Ramanathan and E. L. Lloyd, "Scheduling algorithms for multihop radio networks," *IEEE/ACM Transactions on Networking*, vol. 1, no. 2, pp. 166–177, 1993.
- [18] S. Gandham, M. Dawande, and R. Prakash, "Link scheduling in sensor networks: distributed edge coloring revisited," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '05)*, 2005.
- [19] Y. Wu, X. Y. Li, Y. Liu, and W. Lou, "Energy-efficient wakeup scheduling for data collection and aggregation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 2, pp. 275–287, 2009.
- [20] A. S. Asratian and R. R. Kamalian, "Interval coloring of the edges of a graph," *Applied Mathematics*, vol. 5, pp. 25–34, 1987 (Russian).
- [21] A. S. Asratian and R. R. Kamalian, "Investigation on interval edge-colorings of graphs," *Journal of Combinatorial Theory, Series B*, vol. 62, no. 1, pp. 34–43, 1994.
- [22] S. V. Sevastjanov, "On interval colorability of a bipartite graph," *Metody Diskretnogo Analiza*, vol. 50, pp. 61–72, 1990.
- [23] M. Kubale, "Interval edge coloring of a graph with forbidden colors," *Discrete Mathematics*, vol. 121, no. 1–3, pp. 135–143, 1993.
- [24] K. Giaro, *Compact task scheduling on dedicated processors with no waiting periods*, Ph.D. dissertation, Technical University of Gdansk, ETI Faculty, Gdansk, Poland, 1999.

- [25] K. Giaro and M. Kubale, "Consecutive edge-colorings of complete and incomplete cartesian products of graphs," *Congressus Numerantium*, vol. 128, pp. 143–149, 1997.
- [26] K. Giaro and M. Kubale, "Compact scheduling of zero-one time operations in multi-stage systems," *Discrete Applied Mathematics*, vol. 145, no. 1, pp. 95–103, 2004.
- [27] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 95–107, November 2004.
- [28] R. E. Best, *Phase-Locked Loops: Design, Simulation and Applications*, McGraw-Hill, New York, NY, USA, 2003.
- [29] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: a media access protocol for wireless LAN's," in *Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94)*, 1994.
- [30] J. Misra and D. Gries, "A constructive proof of Vizing's theorem," *Information Processing Letters*, vol. 41, no. 3, pp. 131–133, 1992.
- [31] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: coverage, connectivity and diameter," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 1073–1083, April 2003.
- [32] V. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff, "Design of surveillance sensor grids with a lifetime constraint," in *Proceedings of the First European Workshop on Wireless Sensor Networks (EWSN '04)*, 2004.
- [33] C. Zhang, J. Kurose, Y. Liu, D. Towsley, and M. Zink, "A distributed algorithm for joint sensing and routing in wireless networks with non-steerable Directional Antennas," in *Proceedings of the 14th IEEE International Conference on Network Protocols (ICNP '06)*, pp. 218–227, November 2006.
- [34] D. Musiani, K. Lin, and T. S. Rosing, "An active sensing platform for wireless structural health monitoring," in *Proceedings of the 6th IEEE International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 390–399, April 2007.
- [35] S. Bapat, V. Kulathumani, and A. Arora, "Analyzing the yield of exscal, a large-scale wireless sensor network experiment," in *Proceedings of the 13th IEEE International Conference on Network Protocols (ICNP '05)*, pp. 53–62, November 2005.

Research Article

Novel Energy-Efficient Miner Monitoring System with Duty-Cycled Wireless Sensor Networks

Peng Guo,¹ Tao Jiang,¹ and Kui Zhang²

¹ Wuhan National Laboratory for Optoelectronics, Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

² Center of Pervasive System, University of Twente, 7500 AE Enschede, The Netherlands

Correspondence should be addressed to Tao Jiang, tao.jiang@ieee.org

Received 15 July 2011; Revised 29 September 2011; Accepted 13 October 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Peng Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Target monitoring is an important application of wireless sensor networks. In this paper, we develop an energy-efficient miner monitoring system with sensor nodes. To keep monitoring miners' activities in tunnels, periodical localization and timely data transmission are both required. Since the localization and data transmission much depend on the media access control (MAC) scheme, codesign of localization and MAC scheme is actually needed for the resource-constrained system, which is seldom discussed in existing related works. Moreover, as sensor nodes form an ultra-sparse network with linear topology in tunnels, it is a challenge for existing range-free localization methods to localize targets. In this paper, we propose a localization-MAC codesign approach for the monitoring system under the environment of coal mine. With the proposed approach, the system can achieve higher localization accuracy with low energy consumption and transmission delay, compared with existing range-free localization methods for sensor nodes.

1. Introduction

Target monitoring is an important application of wireless sensor networks (WSNs). With targets being equipped with communication devices, for example, tiny wireless sensor nodes, each target can obtain its location by estimating the distance to some fixed sensor nodes whose locations are known by all targets. A WSN can be formed by fixed sensor nodes and targets, and the location information of targets can be transmitted through the WSN to a station connected with a monitor.

To monitor miners in tunnels, it is indeed needed to locate miners periodically since they change their locations frequently. In addition, locations of miners should be transmitted to the station during the localization period. Moreover, sensor nodes used in the monitoring system are usually battery-powered; thus, low energy consumption should be taken into account in the design of the localization and data transmission. Considering these requirements, challenges of designing an energy-efficient miner monitoring system with sensor nodes in a coal mine could be summarized as follows.

- (i) *Density Restriction for Localization.* It is unreasonable to deploy a large number of sensor nodes in coal mine tunnels because their deployment, management, and maintenance are difficult, costly, and labor-intensive. Therefore, it is a big issue for most range-free localization methods to locate sensor nodes deployed with low density in ultra long but very narrow coal mine tunnels. Moreover, as a miner with a sensor node moves around frequently, localization needs to be performed for him even when he is alone.
- (ii) *Energy Restriction for Localization.* It is known that a heavy heavy communication overhead consumes much energy which should be strictly limited in the system. Moreover, to work for a long time, sensor nodes usually work in a duty-cycle way according to a sleep schedule. However, existing localization approaches do not discuss how sensor nodes working with a sleep schedule perform the localization.
- (iii) *Time Restriction for Localization.* Obviously, the time needed by a sensor node to obtain its location should

be lower than the localization period, since the location information of miners needs to be gathered periodically. Furthermore, to maintain high accuracy of the localization, the localization period should be short, as miners may update their locations frequently.

- (iv) *Delay Restriction for Transmission.* Obviously, the location information should be transmitted quickly through the monitoring system to the station outside coal mine before the next round of localization starts. However, as tunnels are usually very long, the monitoring system deployed in tunnels has a large hop count which would lead to large transmission delay. Therefore, it is indeed needed for us to reduce the transmission delay as small as possible.

According to the challenges, it is indeed for the miner monitoring system to make a novel codesign of the localization and communication. On the one hand, the localization approach should efficiently utilize activities of sensor nodes, which should be based on the communication schedule. On the other hand, the communication schedule should efficiently support the localization and data transmission to maintain low energy consumption and low localization/transmission delay.

However, existing monitoring schemes usually focus on improving the localization accuracy, while seldom discuss integration design of localization and communication scheme [1–3]. Although some works [4] have discussed the problem of duty-cycled monitoring system in which targets are monitored by duty-cycled sensor nodes, localization accuracy is not strictly required. The targets in the system estimate their locations just by judging whether they are in other nodes' communication range. Therefore, the localization accuracy is low, especially in coal mine tunnels where sensor nodes' communication range is usually very large (e.g., about 100 m for MicaZ nodes). Furthermore, all existing monitoring schemes ignore the problem of location information transmission during the localization in WSNs. In a word, in miner monitoring system, high localization accuracy, low energy consumption, and low transmission delay are all required, which could hardly be achieved by existing schemes.

In this paper, we propose a novel miner monitoring system with codesigning the localization and communication media access control (MAC). The proposed system consists of a number of sensor nodes which are efficiently scheduled for both localization and data transmission. Meanwhile, sensor nodes wake up in a *level-by-level offset* way to achieve very low transmission delay, and each node keeps awake for a short time to transmit a beacon for localization or to receive a data packet about the location information of miners. Moreover, the localization in the proposed system is implemented by opportunistically using the wake-up duration of nodes scheduled for data transmission, which could largely reduce the communication overhead. Compared with existing range-free localization methods, the proposed system can achieve higher location accuracy, and the energy

consumption as well as the data transmission delay in the system is very low.

The rest of the paper is organized as follows. In Section 2, related works are presented. In Section 3, the proposed monitoring system is introduced in detail, and the reliability of the proposed system in practical environment is analyzed in Section 4. In Section 5, simulations are conducted to evaluate performance of the proposed system, followed by conclusions in Section 6.

2. Related Works

Generally, existing localization approaches with sensor nodes could be classified as two types of the range-based and range-free. Range-based methods usually employ the distance measured according to the packet arrival time and the angle measured from the angle of arrival signals to calculate the location. Although range-based methods could obtain a high accuracy of localization, it is obtained at the cost of the fast-speed hardware and high energy consumption. Therefore, it is hard in practice to deploy cheap, simple, and reliable sensor nodes for range-based solutions. Other range-based solutions apply the radio signal strength (RSS) to estimate the point-to-point distance [5], and they can be easily implemented. However, due to a dynamic range of the signal strength, it leads to a low accuracy of the localization. Therefore, to improve the accuracy of the localization, an RSS map was employed [3], in which a prior map for the expected RSS is required with extensive measurements at many positions. However, high computational complexity is usually needed in this kind of methods and a dense node deployment is usually expected to achieve high localization accuracy. Moreover, the performance is still much sensitive to the dynamics of the indoor communication environment, due to multipath fading, reflections, diffraction, and interference.

For range-free localization methods, they do not require the availability and validity of the range information, and they have been pursued as cost-effective alternatives for expensive range-based schemes. Most of range-free localization methods are based on the methodology proposed in the ad hoc positioning system (APS) [6], their key idea is to place small fraction of anchor nodes with known coordinates across the network, and locations of other sensor nodes are obtained from the estimated distance to multiple anchor nodes. Obviously, the location error can be masked by features such as node redundancy and data aggregation [7, 8]. However in sparse networks, the performance and accuracy of range-free localization methods greatly deteriorate and the location error will increase to such an extent of more than 100% of the transmission range [6]. Obviously, the sensor network deployed in coal mine is an ultra-sparse network, and miners may even have no neighbors nearby. In addition, as the rate of targets (miners) in tunnels is very small (e.g., 1 m/s) relative to the large communication range of sensor nodes in tunnels (e.g., about 100 m for MicaZ nodes), the typical Monte-Carlo localization method employed in recent existing range-free localization schemes [9, 10] can hardly work. Therefore, in such primitive circumstance in coal

mine tunnels, the performance of most existing range-free localization methods will decrease to the same level of the performance of APS.

Moreover, existing localization methods usually focus on the improvement of the localization accuracy, while they ignore the energy consumption of sensor nodes. Recently, some energy-efficient localization algorithms have been proposed to aim at reducing the sampling ratio of mobile sensor nodes during the localization [3, 10]. Since sensor nodes are battery-powered, to work for a long time without being recharged, sensor nodes usually apply a sleeping schedule to sleep in most of the time and to wake up for transmission or reception in a short duration in each duty cycle. Existing energy-efficient localization algorithms do not study how to implement the localization with duty-cycled sensor nodes.

Obviously, low transmission delay, as well as the localization accuracy and energy consumption, is also required to consider in the monitoring system. Currently, some delay-efficient MAC schemes have been proposed for duty-cycled sensor nodes. In [11], according to the demand from their sender nodes, sensor nodes wake up at the appropriate time, resulting in that the MAC scheme can achieve low delivery latency. However, the demand can only be delivered during short duration of nodes active time in each duty cycle, which limits the hop count of the transmission in each duty cycle. That is to say, the average transmission delay in each hop is a fraction of the duty cycle. According to an established directional data traffic path, a sleep schedule was designed in [12], in which all nodes wake up when their sender nodes get data packets and go to sleep as soon as they transmit the packets to their receiver nodes. The data packets can cascade step by step from the leaves of the tree towards the sink without any waiting on the path. Hence, the delay due to sleep latency can be essentially eliminated. For convenience, we call this kind of sleep schedule as *level-by-level offset* schedule. However, as the wake-up time of sensor nodes is fixed and limited in *level-by-level offset* schedule, it is challenging for sensor nodes to perform the localization and the gathering of location information.

3. The Proposed Monitoring System

3.1. System Model and the Basic Idea. Figure 1 illustrates a model of the miner monitoring system in a mine tunnel, where each miner takes a sensor node with himself and walks around in the tunnel. For convenience, we call this kind of sensor node as the mobile node in the monitoring system. To cover all mobile nodes in the tunnel, a number of sensor nodes are deployed equidistantly in the tunnel. These sensor nodes are called as anchor nodes. The distance between any two adjacent anchor nodes is D , and the maximum transmission range R of anchor nodes satisfies $D < R < 2D$.

We suppose that anchor nodes are required to work for a long time without recharging their batteries. In practice, it is unexpected to charge anchor nodes with an electrical cable in tunnels, due to the inconvenience of nodes deployment and expectation of immunity from some critical event, for example, destruction of the cable. Hence, anchor nodes have to work in a duty-cycled way, while we suppose that mobile

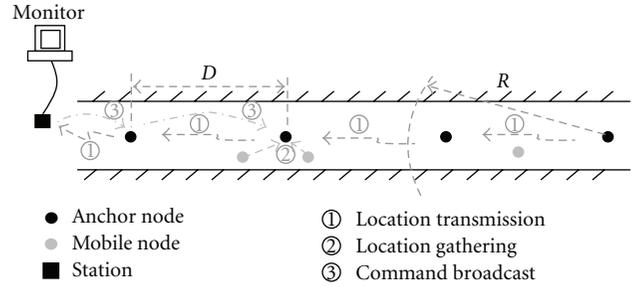


FIGURE 1: Model of the monitoring system in coal mine.

nodes can keep working during the monitoring, because miners can take mobile nodes out of the tunnel and recharge the batteries. As the current needed by the transceiver of a sensor node is usually about 20 mA, a battery with capacity 1000 mAh can support a mobile node to work for more than two days, which is sufficient for miners working in the tunnel.

Before introducing the details of the proposed monitoring system, we give an overview of the system. During the monitoring, anchor nodes periodically broadcast beacons in each duty cycle with two different levels of transmission power. Each beacon contains the anchor node's ID and the current level of the transmission power. Hence, mobile nodes in the tunnel can estimate their location range according to the beacons that they receive. Then, mobile nodes report their location information to nearby anchor nodes (i.e., action ② in Figure 1), and the anchor nodes will relay the location information to the station outside of the tunnel (i.e., action ③ in Figure 1). With efficiently scheduling the activities of anchor nodes and mobile nodes, there could be no collision among the beacon broadcasting, location reporting, and location relaying, and the transmission delay for the location relaying among anchor nodes can be very low. In addition, based on the schedule, the station in the system can also quickly disseminate some commands to all anchor nodes in the tunnel without needing any extra active time of the anchor nodes.

3.2. Localization. As miners may walk around during the localization, it is meaningless to expect the localization approach to have high location accuracy. For example, suppose that the moving rate of miners is about 1.5 m/s, the period of localization is 3 s, and the transmission delay among anchor nodes is about 1 s. Hence, when the station obtains a miner's location information, the miner may have left the location to a new place $1.5 * (3 + 1) = 6$ meters away. Due to this fact, range-free localization method, with low requirement of the hardware and computation, is employed in the proposed localization approach, although the localization accuracy is not so high as that of the range-based localization methods.

To improve location accuracy, the proposed localization approach sets anchor nodes with two different transmission power levels, which is shown in Figure 2(a). During the localization, anchor nodes periodically change their transmission power levels to achieve different transmission ranges.

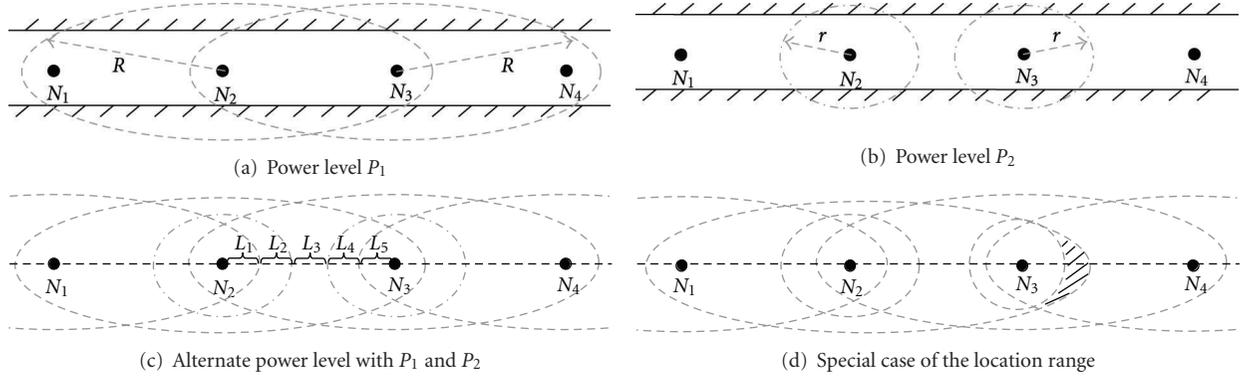


FIGURE 2: Different transmission ranges due to two power levels of anchor nodes.

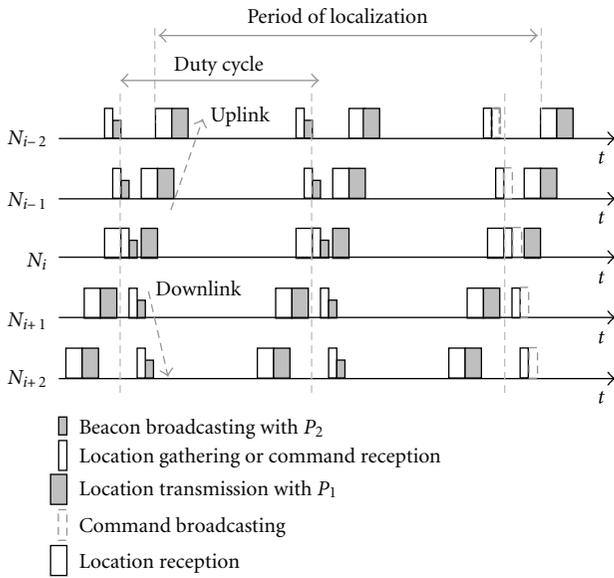


FIGURE 3: Schedule of anchor nodes.

It is obvious from the right top part of Figure 2(a) that when the transmission power level of anchor nodes is P_1 , anchor nodes could corporately cover all mobile nodes in the tunnel. Moreover, each mobile node can detect at least two anchor nodes if anchor nodes periodically broadcast beacon packets with transmission power P_1 . Some mobile nodes may even detect three anchor nodes in some place. For example, a mobile node nearby anchor node N_2 in Figure 2(a) may receive beacon packets broadcast by anchor nodes N_1 , N_2 , and N_3 . Hence, by checking the number of anchor nodes detected by mobile nodes, the area between any two adjacent anchor nodes can be partitioned into three location ranges with different detection results.

When the transmission power levels of anchor nodes are P_2 , as shown in Figure 2(b), only part of the tunnel could be covered by all anchor nodes. In this case, some mobile nodes can detect only one anchor node, for example, the mobile node nearby an anchor node, while, at some positions, mobile nodes may detect none anchor node, for

example, the mobile node in the middle of two adjacent anchor nodes. Through this way, the area between any two adjacent anchor nodes can be partitioned into three new location ranges according to the detection result.

Combining the two partitions illustrated in Figures 2(a) and 2(b), five location ranges can be distinguished within the area between any two adjacent anchor nodes, as shown in Figure 2(c). Mobile nodes can determine which location range it is according to the power level contained in the packets that they receive. Furthermore, as the area between any two adjacent anchor nodes is partitioned into five ranges, the location accuracy of mobile nodes can be effectively improved.

However, the communication ranges of anchor nodes are dynamically changed as the wireless communication environment in tunnels is unsteady in practice. Sometimes, there even could be a special location range (as the shaded part shown in Figure 2(d)), where the detection result is different from those in location ranges shown in Figure 2(c). To handle this new case (as well as all possible cases), in next section, an effective estimation method for location based on wireless link measurement in practice is introduced.

3.3. Anchor Nodes. As described in the previous system model, three tasks are executed by anchor nodes:

- (i) periodically localize mobile nodes nearby,
- (ii) periodically gather locations of mobile nodes nearby and transmit locations to the neighboring anchor node closer to the station,
- (iii) occasionally broadcast command information sent from the station to mobile nodes nearby and the neighboring anchor node further to the station.

To save energy, anchor nodes need to employ a sleep schedule which guarantee anchor nodes to sleep in most time and transmit packets with low transmission delay. Moreover, the sleep schedule should also guarantee anchor nodes to easily perform the localization. According to these requirements, the idea of *level-by-level offset* schedule is actually much suitable to anchor nodes which are deployed with a linear topology. Based on this idea, the schedule of anchor nodes is illustrated in Figure 3.

As shown in Figure 3, all anchor nodes work in a duty-cycle way. Each of them takes two *level-by-level offset* sleep schedules. One is for the possible traffic of command broadcasting which is from the station to all anchor nodes and mobile nodes (called as downlink traffic). The other is the periodical traffic of the location transmission which is from anchor nodes to the station (called as uplink traffic). Anchor nodes wake up one by one according to their hop counts to the station and transmit or receive beacon/command/location packets according to the traffics. In this way, packets can be transmitted along anchor nodes without waiting, although anchor nodes sleep in most of the time.

As the downlink traffic of command broadcasting does not always occur, the wake-up duration of anchor nodes for downlink traffic can be utilized for beacon broadcasting and location gathering. To avoid the collision between the location gathering and the possible command broadcasting, mobile nodes always overhear the channel before they send their locations during the location gathering. For example, suppose that a mobile node is going to send its location information to anchor node N_i which is i hop counts away from the station; the mobile node should overhear the channel at the time when anchor node N_{i-1} wakes up for command reception in advanced. If the mobile node finds that N_{i-1} does not acknowledge any command, it can send its location information to N_i at the time when anchor node N_i wakes up for command reception.

Note that the size of duration for downlink traffic is different from that of duration for uplink traffic in the schedule. Within duration for downlink traffic, command or location information of local mobile nodes needs to be successfully transmitted, while, within the duration for uplink traffic, the location information of all miners in the tunnel may need to be transmitted. In practice, the period of localization is usually larger than the length of duty cycle, as shown in Figure 3. There can be several durations for uplink traffic within the period of localization; that is, anchor nodes have several chances to transmit the location information of all miners. Therefore, the size of duration can be set according to the ratio of localization period to duty cycle. Considering the issue of unreliable communication links in practice, retransmission also needs to be taken into account when setting the duration size.

3.4. Mobile Nodes. As described in the aforementioned system model, tasks of mobile nodes are as follows:

- (i) estimate their locations,
- (ii) transmit their locations to anchor nodes.

To estimate the location, each mobile node keeps detecting beacon packets and location packets transmitted by anchor nodes. Since the two kinds of packets are periodically transmitted with different transmission power levels, the mobile node can estimate its location according to detection results.

To send its location information to anchor nodes, the mobile node needs to select one of anchor nodes detected

by it as the destination. In the proposed system, the mobile node chooses the anchor node detected with largest hop count to the station. This is because that the mobile node needs to overhear the anchor node with smaller hop count to check whether there is command broadcasting, as mentioned previously.

To avoid the collision due to the transmissions from multiple mobile nodes to one anchor node, one of the mobile nodes is elected to gather locations of mobile nodes with the same destination anchor node in advanced. The election is executed during the time when the nearby anchor nodes sleep. Each mobile node broadcasts its location in a randomly chosen time slot before the destination anchor node wakes up for location gathering. Specially, the mobile node broadcasts the packet with its radio module working in data burst transmission mode [13]. As the duration of data burst transmission is only about $80 \mu\text{s}$ which is much shorter than the sleep duration of anchor node, collision can be almost avoided. When the destination anchor node wakes up for location gathering, each mobile node has obtained a location list, and the mobile node with largest ID is selected to send the location list to the anchor node.

4. Reliability Analysis of the System

In practice system, there are some aspects that should be taken into consideration such as time synchronization, dynamic communication range, and packet loss. We analyze them as follows.

4.1. Time Synchronization. In the proposed system, anchor nodes periodically sleep and wake up to transmit or receive message according to their sleep schedules, and all mobile nodes should be synchronized with anchor nodes to detect packets or transmit their locations. As there could be clock drift for nodes, synchronization is extremely important for the proposed system.

Since anchor nodes are deployed with a linear topology, actually, the synchronization can be easily maintained with the command broadcasting in the system. By using time stamp in the command, the maintenance of synchronization can achieve an accuracy of $2.24 \mu\text{s}$ with broadcasting command every 15 minutes [14]. The accuracy of $2.24 \mu\text{s}$ is sufficient for sensor nodes, since they usually need several milliseconds to transmit a packet.

4.2. Dynamic Communication Range. As the communication range of anchor nodes is dynamic in practice, a mobile node in the expected communication range of an anchor node may not receive the packet sent by the anchor node. Similarly, the mobile node out of the expected communication range of an anchor node may possibly receive packet sent by the anchor node. Obviously, the dynamic communication range will decrease the localization accuracy. To solve it, we introduce a novel and simple method for the location estimation in practice.

Before the monitoring system begins to work, communication link measurement in the coal mine tunnel is suggested

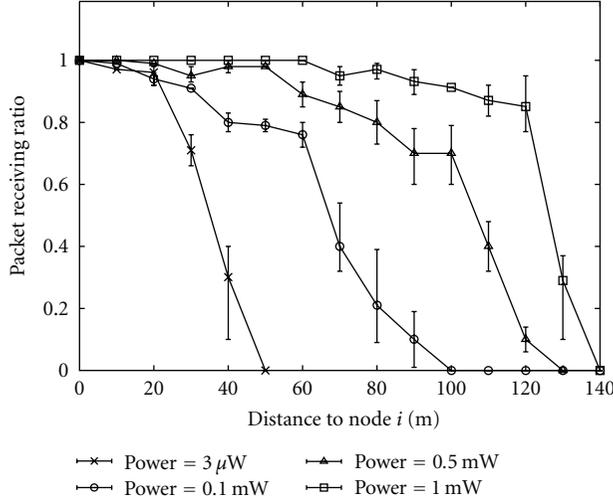
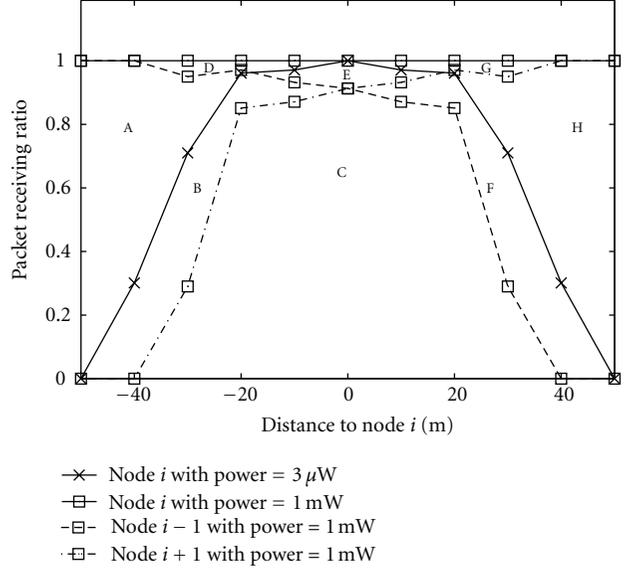
(a) Reception from node i with different power levels(b) Reception from node i , $i - 1$, and $i + 1$

FIGURE 4: Location estimation based on link measurements in the coal mine tunnel.

to be made. Figure 4(a) shows some measurement results about the relationship between the transmission range and power level in real coal mine environments in the Shan Dong province in China. The radio chip used in sensor nodes is CC2420. Figure 4(a) shows the average value of the packet receiving ratio as well as the maximum and minimum value for 200 tests. It can be seen that when transmission power is 1 mW, the transmission range of the sensor node can be up to 120 m, which is much longer than the transmission range of node in indoor environment. On the other hand, when transmission power is $3 \mu\text{W}$, the transmission range is about 30 m.

In practice, the transmission power levels of anchor nodes need to be set according to the requirement of location accuracy. Obviously, the lower the power level is, the shorter distance between adjacent anchor nodes should be, and the higher accuracy would be achieved. Suppose that 1 mW and $3 \mu\text{W}$ are chosen as the transmission power levels in the monitoring system, and the gap of anchor nodes is set to be $D = 100$ m. It can be seen that the transmission range of anchor nodes with power level 1 mW is usually larger than D , and the transmission range of anchor nodes with $3 \mu\text{W}$ is usually smaller than $D/2$. To list all possible detection results of mobile nodes, it is sufficient to just observe the range of $D/2 = 50$ m around an anchor node i , as shown in Figure 4(b) where the transmission ranges of anchor nodes $i - 1$, i , and $i + 1$ with 1 mW and $3 \mu\text{W}$ are illustrated, based on the measurement data in Figure 4(a). It can be seen that the size of each region outlined in Figure 4(b) actually stands for the probability for one kind of detection result. For example, region A is above the curve for “node i with power $3 \mu\text{W}$ ”, while is below the curve for “node i with power 1 mW” and the curve for “node $i - 1$ with power 1 mW”. Hence, the size of region A can denote the probability for the

detection result that a mobile node can only detect anchor node $i - 1$ and i with power level 1 mW. To estimate the location according to one detection result, the maximum height of the corresponding region at all possible positions is searched, and the location corresponding to the maximum height would be the mobile node’s location with the highest probability.

In addition, as there may be several duty cycles in each period of localization, the location error can be reduced by averaging the multiple locations estimated.

4.3. Packets Loss and Data Recovery. Although an anchor node could be well covered by the transmission range of its adjacent anchor node, packet transmission between them may still be possible to fail with a small probability. In addition, the collision among mobile nodes may also be possible to take place. The failed transmission and collision obviously results in packets loss, and retransmitting lost packets surely cause extra transmission delay. If the extra transmission delay leads to the outcome that the total transmission delay would be larger than the period of localization, the packet has to be dropped.

It is acceptable for monitoring system to lose some location packets, because mobile nodes are periodically localized and their locations are periodically updated. Moreover, the lost location information can even be recovered with those locations successfully received by the station. For example, suppose that mobile node M_i ’s locations in the j th and $j + 2$ th period of localization are successfully transmitted to the station, while the location in the $j + 1$ th period of localization is lost. The station can recover the lost location as the middle of the two locations in the j th and $j + 2$ th period of localization. To solve more complex cases, methods

in numerical analysis theory can be applied, and the location error can also be flatted in this way.

5. Performance Evaluation of the System

We give some simulations with an example for the proposed system, showing the location accuracy of the proposed localization approach.

5.1. Simulations for Localization. Suppose that a monitoring system consisting of some anchor nodes and mobile nodes is deployed in a tunnel. The maximum hop count in the system is 20, that is, the number of anchor nodes is 20. The period of localization is required to be 3 s, and the length of duty cycle should be no shorter than 1 s based on a requirement about the lifetime. The duration of anchor nodes for uplink traffic is set to be $1/20$ s = 50 ms. Since the location information of all mobile nodes should be transmitted to the station within 3 s, at least one third of the location information is expected to be transmitted within the duration in each duty cycle. Assume that the location information of a mobile node as well as its ID can be denoted with two bytes. Hence, about 80 mobile nodes' location information can be transmitted by an anchor node with radio chip CC2420 within 50 ms. Therefore, the proposed system can monitor almost about $80 * 3 = 240$ miners in a tunnel, which is much sufficient in practice. From another perspective, if the real number of miners is far less than 240, for example, 120, it is possible to transmit the same location information twice within the duration, which helps to enhance the link reliability.

We use the packet receiving ratio measured in Figure 4(a) to denote the link quality between sensor nodes in simulations. Transmission power levels of anchor nodes are set to be 1 mW and 3μ W. The distance between any adjacent anchor nodes is 100 m. A mobile node is deployed at 10 different positions between two adjacent anchor nodes to measure its location according to the proposed localization approach.

To compare the location accuracy, we choose the basic range-free localization method, that is, APS method in [6], due to the reason that the performance of most existing range-free localization methods is actually equivalent to that of APS in ultra-sparse network with linear topology. In coal mine tunnels, each target usually has just two neighbors (anchor nodes) in most of time, and there could hardly be any more reference provided for the range-free localization. Moreover, as the rate of targets is far small compared with anchor nodes' communication range, the slight movement of targets could hardly bring any change about the connection relationship between nodes in most of time. Hence, under this simple communication environment, most localization methods proposed in existing range-free localization schemes ([8–10] and the references therein) could hardly work. Targets have to estimate their locations just by judging whether they are inside an anchor node's transmission range in these schemes, resulting in the same performance of these schemes as that of APS.

Figure 5 shows results of location estimation at the 10 positions. For each result, we give the average location

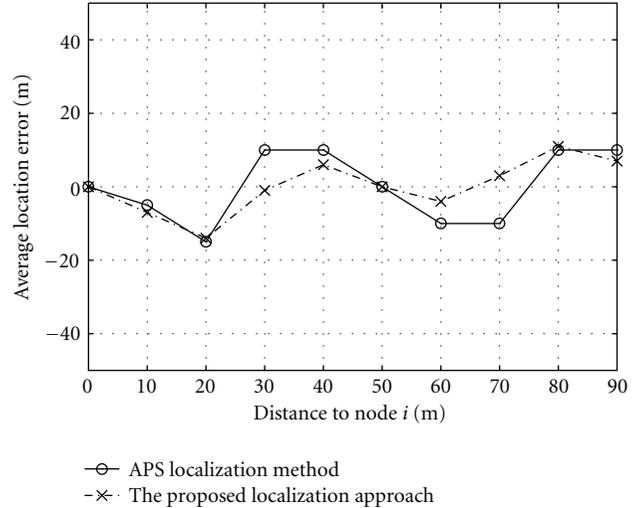


FIGURE 5: Average location error at different positions.

error with 10 times of estimation. It can be seen that, with alternating transmission power levels, the proposed approach can achieve smaller location error than that of APS under this primitive circumstance with unreliable links. However, due to unreliable links, the location estimated with the two methods is dynamic. During the monitoring, the dynamics can be mitigated by modifying locations according to the continuity of miners movements.

5.2. Simulations for Monitoring. To show the monitoring performance of the system, we set an arbitrary path for a miner between anchor nodes i and $i + 1$. The speed of the miner is 1 m/s. Considering that there are three duty cycles with a localization period in the example, since the miner can make a location estimation during each duty cycle, he averages the three locations estimated within each localization period and sends the average result to anchor nodes. Figure 6(a) shows the monitoring results with the proposed approach and the APS method. It takes 180 s for the miner to finish walking along the path. It can be seen that as the speed of the miner is low, he may stay in the same location range for several seconds. The location results estimated are usually dynamic although the miner is still in the same location range, while, when he walks into the middle region between the two anchor nodes, the locations estimated keep invariable even when the miner walks tens of meters away. As a whole, the path estimated with the proposed approach can follow more closely to the real path than that with APS method.

To mitigate the dynamics of localization results and the affection of packet loss during the transmission in practice, a data smoothing technology can be applied for the results. Figure 6(b) shows the paths smoothed with the five-point triangular smoothing algorithm. It can be seen that the proposed system has better monitoring performance than that of the APS.

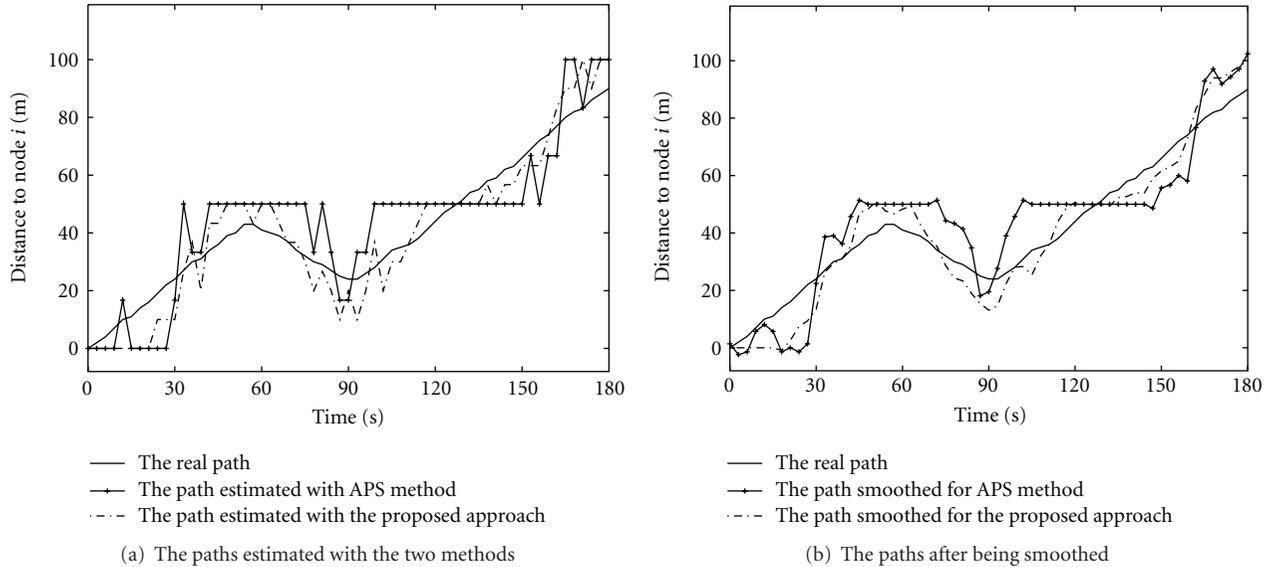


FIGURE 6: Monitoring a miner walking along an arbitrary path.

5.3. Analysis of Energy Consumption and Delay. We analyze the energy consumption and transmission delay of the proposed system as follows.

In the proposed system, the energy consumption due to idle listening is ultra low, as anchor nodes turn on their transceivers only in the determined duration in each duty cycle. In addition, the size of the active duration can be set to be optimal for beacon transmission and location information gathering. Moreover, the localization duty in the proposed system almost causes no extra communication cost except for the periodical beacon broadcasting with a lower level of transmission power in each duty cycle. Therefore, it can be seen that, for a given length of duty cycle (related to the lifetime of anchor nodes) and the traffic load (related to the number of miners in the tunnel), the proposed system is much efficient in energy consumption.

As for the transmission delay, since anchor nodes in the proposed system use the *level-by-level offset* sleep schedule, packets can be relayed along anchor nodes without waiting for the destination to wake up, and minimum transmission delay can be achieved in ideal environment. To avoid retransmission in next duty cycle due to unreliable communication links, which causes high retransmission delay in the *level-by-level offset* sleep schedule, larger duration is set for the traffic so that lost packets can be retransmitted within the duration in current duty cycle. As the link quality between anchor nodes is about 91.4% in the example, another transmission chance in the duration could improve the equivalent link quality up to $1 - (1 - 91.4\%)^2 = 99.3\%$. Hence, packets can be relayed along anchor nodes with little retransmission in next duty cycle, and the transmission delay along anchor nodes with 20 hop counts usually can be about $50 \text{ ms} * 20 = 1 \text{ s}$, while, in most asynchronous schedule schemes, sender nodes usually have to wait for a long duration till destinations wake up. The average transmission delay within each hop is about half of the duty cycle, that is, 0.5 s in the example.

Hence, the total transmission delay along anchor nodes with 20 hop counts will be about 10 s which is far larger than the localization period required.

6. Conclusions

In this paper, we proposed an energy-efficient miner monitoring system with sensor nodes for coal mine. Specially, a novel codesign solution of the localization and MAC schemes was proposed. With sensor nodes being well scheduled, the localization can be simply implemented by opportunistically using the wake-up duration scheduled for the possible command transmission, which needs little extra communication overhead. The designed schedule for sensor nodes is very suitable for the tunnel environment, where sensor nodes are deployed with a linear topology and time synchronization can be easily maintained. With the proposed schedule, low energy consumption and transmission delay can both be achieved in the proposed mine monitoring system. Moreover, with the transmission power level of sensor nodes being periodically changed, the localization accuracy could be improved.

Acknowledgments

The work presented in this paper was supported in part by the NSF of China with Grant 60903171, the National S&T Major Project with Grant 2010ZX03006-002, and the Fundamental Research Funds for the Central Universities with Grant 2011TS111.

References

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

- [2] I. F. Akyildiz and E. P. Stuntebeck, "Wireless underground sensor networks: research challenges," *Ad Hoc Networks*, vol. 4, no. 6, pp. 669–686, 2006.
- [3] A. Cenedese, G. Ortolan, and M. Bertinato, "Low-density wireless sensor networks for localization and tracking in critical environments," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 6, Article ID 5456173, pp. 2951–2962, 2010.
- [4] C. F. Hsin and M. Liu, "Randomly duty-cycled wireless sensor networks: dynamics of coverage," *IEEE Transactions on Wireless Communications*, vol. 5, no. 11, pp. 3182–3192, 2006.
- [5] 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 (MOBICOM '01)*, pp. 166–179, Rome, Italy, July 2001.
- [6] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM '01)*, San Antonio, Tex, USA, November 2001.
- [7] 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 Annual International Conference on Mobile Computing and Networking (MOBICOM '03)*, San Diego, Calif, USA, Spetember 2003.
- [8] D. Ma, M. J. Er, and B. Wang, "Analysis of hop-count-based source-to-destination distance estimation in wireless sensor networks with applications in localization," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 6, Article ID 5447686, pp. 2998–3011, 2010.
- [9] 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, Article ID 5210109, pp. 516–526, 2010.
- [10] S. Zhang, J. Cao, L. Chen, 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.
- [11] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing 2008, (MOBIHOC'08)*, pp. 53–62, Hong Kong, China, May 2008.
- [12] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup scheduling in wireless sensor networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing 2008, (MOBIHOC'06)*, pp. 322–333, Florence, Italy, May 2006.
- [13] 2.4 GHz IEEE 802.15.4/ZigBee-Ready RF Transceiver, <http://www.ti.com/product/cc2420>.
- [14] G. S. M. Maroti, B. Kusy, and A. Ledeczi, "The flooding time synchronization protocol," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04)*, November 2004.

Research Article

Survey: Discovery in Wireless Sensor Networks

Valerie Galluzzi and Ted Herman

Department of Computer Science, University of Iowa, Iowa City, IA 52242, USA

Correspondence should be addressed to Ted Herman, ted-herman@uiowa.edu

Received 16 July 2011; Revised 7 October 2011; Accepted 13 October 2011

Academic Editor: Yuhang Yang

Copyright © 2012 V. Galluzzi and T. Herman. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Neighbor discovery is a component of communication and access protocols for *ad hoc* networks. Wireless sensor networks often must operate under a more severe low-power regimen than do traditional *ad hoc* networks, notably by turning off radio for extended periods. Turning off a radio is problematic for neighbor discovery, and a balance is needed between adequate open communication for discovery and silence to conserve power. This paper surveys recent progress on the problems of neighbor discovery for wireless sensor networks. The basic ideas behind these protocols are explained, which include deterministic schedules of waking and sleeping, randomized schedules, and combinatorial methods to ensure discovery.

1. Introduction

In the decade following the introduction of Wireless Sensor Networks (WSNs) to the lexicon, the technical landscape of applications, network protocols, and research problems has shifted somewhat. The early focus on basic communication issues enabled more applications to be deployed, and the catalog of available WSN platforms increased to include many types of radio and processor features. Experience with applications and platforms showed that early perceptions of power challenges and solutions to power management were perhaps misinformed. For example, the lifetime of a sensor node running on battery was not significantly extended by attenuating transmission power. Rather, the most effective means of power conservation consists in powering off components entirely, including sensors and the radio. The appendix of this paper has a small example illustrating how the lifetime of a battery-powered sensor node could vary from days to years depending on effective use of sleep modes. Scheduling operations across a WSN, for example, selectively powering on and off nodes, is a problem of distributed control. Indeed, a fundamental balance is needed to minimize power utilization on one hand, yet facilitate application data forwarding through the WSN on the other hand. The situation is yet more challenging if the network topology is dynamic, nodes are mobile, or nodes depend on harvesting devices to scavenge sufficient power for radio operation.

This paper surveys the literature of one facet of power management in sensor network protocols, namely the problem of *neighbor discovery*. Informally, the problem is to devise an efficient protocol whereby sensor nodes learn of the presence of other nodes within communication range even as they adhere to low-power operation, with the radio mostly off. The crux of the problem is as follows.

A sensor node p needs to communicate with some neighbor q , but that is only possible when both p and q have their radios powered on at the same time.

This problem is particularly relevant to *ad hoc* or mobile deployments where the set of (communication) neighbors of a node is unpredictable or dynamic. The parameters of the problem are many: design choices for power schedules, constraints of processor (resources and timing facilities), hardware features of the radio, and application requirements control what is the set of conceivable solutions. Though several techniques from the literature on neighbor discovery have some combinatorial flavor, the dependence on problem parameters makes framing neighbor discovery as a purely algorithmic problem somewhat difficult. To give the survey context, we examine some related problems, technology and older results from areas of networks and distributed computing. The survey then explains prominent techniques for

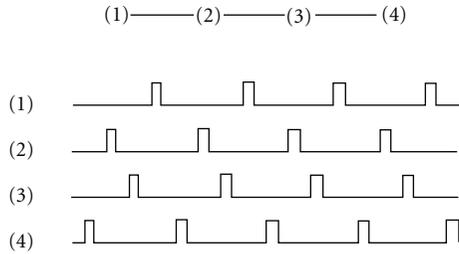


FIGURE 1: An uncoordinated node schedule.

neighbor discovery, metrics for analysis, and several important results from the literature. We have also simulated representative protocols for neighbor discovery, to illustrate for the reader how different design choices affect performance metrics.

Organization. For readers unfamiliar with the problem of neighbor discovery, Section 2 discusses a brief motivating scenario. The paper then presents historical background, sensor node and radio platform considerations before presenting protocols in Section 5 (eager readers may wish to start with Section 5). After reviewing some background concepts from distributed computing in Section 3, considerations that constrain and affect evaluation of protocols are discussed in Section 4. Material in Section 5 organizes neighbor discovery protocols thematically, grouping them by their basic discovery techniques (which mostly repeat historical themes mentioned in Section 3). Section 6 is devoted to performance metrics for neighbor discovery protocols. Final remarks are in Section 7. Some details about hardware and protocols considerations are deferred to the appendix.

2. Motivating Scenario

Protocols for low-power operation in sensor networks turn the radio off between communications. Schedules for turning radio on and off could be periodic, random, or some hybrid of these approaches. Figure 1 shows a scenario for four nodes, (1)–(4). The top part of the figure shows the communication network, which is a linear structure (node (1) is out of range of node (3)). Each of these nodes uses the same periodic awake schedule, waking once every five time intervals. Unfortunately, their schedules are not coordinated in the figure; hence no two nodes are unable to communicate, because their radios are not on at the same time. This unfortunate situation could be the result of improper initialization, crash and restart events at unpredicted times, or the normal dynamic arrival of nodes in an *ad hoc* WSN.

Some MAC protocols arrange to have nodes occasionally sample radio activity during sleeping periods, with the aim of learning what other nodes are in the vicinity and what are their schedules; the appendix cites papers on low-power MAC protocols for WSNs that use sampling. These sampling techniques depend on a radio feature for channel sampling that is fast and consumes very little power. By contrast, the discovery protocols surveyed in Section 5 do not depend on extra radio features. If sampled neighboring node schedules

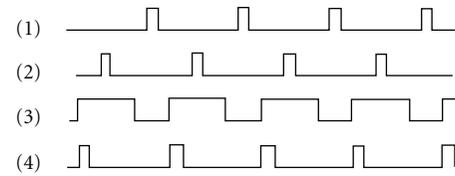


FIGURE 2: Node (3) accommodates its neighbors.

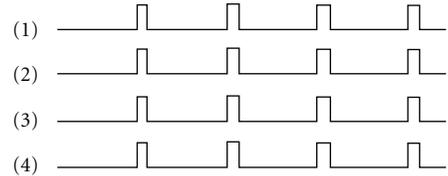


FIGURE 3: Coordinated node schedule.

are predictable (i.e., periodic), then some additional waking time can be scheduled. Figure 2 shows a modification to the scenario of Figure 1 in which node (3) has learned the schedules of (2) and (4), and then added to its waking times to facilitate communication. Note that in addition to enlarging its waking time to accommodate (2) and (4), node (3) retains its original schedule as well, in case any other nodes that have learned about (3) depend on its schedule.

Does learning of neighboring nodes then accommodating their schedules solve the problem of neighbor discovery? Yes, however, one would hope to have a power-optimal solution to neighbor discovery rather than adding additional waking times to accommodate neighbor schedules. Remarkably, protocols (surveyed in Section 5), by careful arrangement of their schedules, are able to learn of neighbors without extra sampling of the radio during their sleeping periods. It should be possible for a WSN to overcome improper initialization or *ad hoc* network formation, so that eventually all schedules are coordinated, as shown in Figure 3. Having all nodes move to a common, coordinated schedule (analogous to TDMA) will result in lower power consumption.

For more traditional, mobile *ad hoc* networks (MANETs) where power conservation is not so critical, neighbor discovery is the simpler problem of continuously detecting that mobile stations come into range—converging to a common schedule like that shown in Figure 3 is not important. Many WSN applications are either event-driven (and nodes cannot wait long to transmit data) or the power requirements are not so stringent. For these applications, learning of and accommodating to neighbor schedules is adequate.

3. Wakeup in Distributed Systems

A sensor node saves power by turning off its radio. While the radio is off, that sensor neither receives messages nor responds to queries or commands. It is dormant as far as other nodes in range of communication are concerned. We thus say that a node is *asleep* when its radio is off, and *awake* when the radio is on. Two nodes are (communication) *neighbors* if they can communicate when both are awake.

Results from the WSN literature on neighbor discovery explore arbitrary communication topologies that have bidirectional communication links. That is, if p and q are neighbors, then p can hear q 's messages and *vice versa*; in reality, the neighbor relation could be asymmetric, so that p could hear from q , whereas q would be unable to receive from p presentation. In WSN deployments, it could be possible that a link is asymmetric; the papers surveyed in this paper generally presume bidirectional links. The assumption of symmetry simplifies analysis and protocol research. In our opinion, neighbor discovery with unidirectional links is an open problem. We suggest in Section 7 some considerations regarding unidirectional links in research. Note, however, if a network can be connected using bidirectional links, then unidirectional links could be ignored or discarded for routing or other applications; whether or not the case of unidirectional communication really matters depends on empirical properties of WSNs in practice.

Other simplifying assumptions about timing are introduced later in the article. Generally, we shall ignore the possibility of failures, including message corruption, radio interference, and frame collision during transmission. Because discovery is an ongoing protocol, engineered to cope with dynamic, *ad hoc* WSNs, the consequence of simplifying assumptions is that the latency for discovery is prolonged by communication failures. So long as communication succeeds with sufficient probability, discovery eventually occurs. Even when more realistic models are used, the techniques and themes surveyed in this article would be valid starting points for design and implementation of neighbor discovery protocols.

The neighbor discovery problem has a trivial solution if nodes are given the ability to “wake up” sleeping neighbors. It is common in wired local area networks to have a special *wakeup* command, which causes sleeping nodes to become awake. This feature turns out to be difficult or prohibitively expensive for sensor nodes at the current level of technology. There is one commonly used exception, passive RFID technology, where nodes receive not only a message but also the power needed to compute and respond, from an electromagnetic signal. Limitations on range and the power needed for signaling (plus the cost of extra components) rule this option out for WSN deployments, so the trivial solution of transmitting a wakeup command and hearing acknowledgments is not considered to be a satisfactory neighbor discovery protocol. However, wakeup considered in a broader context is sufficiently important, yielding many interesting and relevant techniques, as mentioned briefly in the following paragraphs.

Among the well-studied problems for distributed algorithms are variants of the wakeup problem. Perhaps the oldest of these is the firing squad problem [1]: a multihop network is given with all nodes initially asleep; one node is selected to spontaneously wake up, and the goal is to have all nodes perform some action only once, and simultaneously. Algorithms for this task thus rely on the initiator node sending messages to neighbors, which are propagated to their neighbors, and so on, to wake up the entire network; superimposed on this wakeup scheme, there needs to be a timing strategy so that nodes only perform the desired action at the

same instant all their neighbors do (transitively, the entire network). Metrics for optimization include the number of messages, the size of messages, the latency period between initial activation and the firing of the action, and the overhead (memory, program size) of the algorithm. Obviously, a sleeping node cannot know when the initiator will wake up, and this resembles one of the fundamental difficulties of the neighbor discovery problem: a node cannot know (except for very specific deployments) when another node enters into its neighborhood and is capable of being awake.

Theoretical study of wakeup in a shared communication medium network starts with [2]. The network there is single-hop (i.e., a clique topology) and messages are unicast, unlike the wireless model where a single message can be received by all neighbors. Despite such differences from the WSN model, an important observation is relevant to neighbor discovery in a sensor network: the timing of *when* a node sends a message (or engages in some higher-layer multicast protocol) is important. The *schedule* of transmitting messages can be deterministic or random, and the choice of a schedule is crucial to efficiency. One schedule described in [2] gives each node in the system a different schedule, based on periodically transmitting after some silent period. The length of the period is chosen to be a prime number so that each node has a different prime, and this turns out to guarantee certain synchronization properties. We will see in Section 5.4 that such a technique has been exploited in several investigations of the neighbor discovery problem.

Another perspective on discovery, again from the literature of distributed computing, is found in [3], which shows how to match up servers and clients in a distributed, message-passing system. With n servers and n clients, it turns out that $O(\sqrt{n})$ messages suffice to guarantee a fully distributed *rendez-vous* between matching parties (a nondistributed solution would be to have one leader node coordinate all of the matching, but it then becomes a single point of failure or contention). The idea is based on an $n \times n$ matrix, with rows representing servers and columns representing clients. The server of row i tells a set of $P(i)$ nodes about itself, whereas the client of column j queries a set of $Q(j)$ nodes for the desired service. Then, by arranging P, Q so that $P(i) \cap Q(j)$ is nonempty, discovery will occur. It turns out that $|P(i)| = \lceil \sqrt{n} \rceil$ and $|Q(i)| = \lceil \sqrt{n} \rceil$ effectively load balances the match-making process: a lower bound of $\Omega(\sqrt{n})$ is shown in [3] on the average message complexity for discovery between client and server. The view of discovery through a matrix or table with rows and columns representing different parties occurs in the WSN neighbor discovery literature (see intersecting designs in Section 5.3). An earlier reference to such a problem can also be found in the seminal paper [4] on replica control in databases; later, a more sophisticated construction [5] was discovered for mutual exclusion, which achieves $O(\sqrt{n})$ complexity, n being the number of nodes (incidentally, the initial construction depends on finding a prime factor to establish the existence of a particular subset of nodes that guarantee *rendez-vous*). Finding special arrangements of awake times, rather than node locations, turns out to be similar and useful for neighbor discovery.

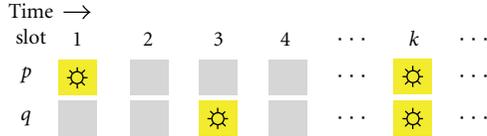


FIGURE 4: Slots for p and q . ☀ denotes awake.

4. Platform

This section briefly introduces terminology and facts about sensor nodes used later in descriptions of protocols. More detail on platform issues can be found in the appendix. Most of the protocols are based on discrete models of time and communication, so a slotted model of time is a reasonable discipline for protocols and a convenient analysis abstraction. The appendix discusses some of the concerns of the slot abstraction as well as duty cycle, processor, clock, and radio facilities relevant to neighbor discovery.

The Slot Model. Before delving into the details of protocols, it is helpful to explain some terminology found in the literature of neighbor discovery. Time is modeled in discrete units called *slots*, which are supposed to be intervals of real time of sufficient length to permit communication. A node can either be awake or asleep in any given slot. Discovery protocols use schedules of awake and asleep intervals, most of them based on slots, with the objective of keeping the ratio of awake slots to total slots to a suitably low duty cycle (see appendix for details and motivation of duty cycles). During operation, we can refer to a node's current slot by some fictional counter value, so that a protocol or schedule may be concisely described. For convenience of presentation and analysis, we further suppose that all nodes commence and terminate their slots in unison: a trace of a WSN execution is thereby depicted by a diagram in Figure 4 where rows are nodes, slot numbers increase left to right, and the starting points for all slots line up vertically. Given such an ideal arrangement of slotted time, the basis for neighbor discovery is easy to define. If p and q are neighbors who have not yet discovered each other, and if they are awake concurrently in some slot k , then they discover each other and the fact of this discovery is retained for slots $k + 1$ and higher. We call the event of p and q discovering each other *mutual recognition*.

Slots are a convenient abstraction, though time on WSN platforms is not inherently divided into slots. Nodes can approximate being awake and asleep for intervals that would approximate multiples of slot length; also, nodes cannot be expected to have their slots precisely aligned as Figure 4 shows. Assume that a slot is the minimum-length time interval for two nodes to exchange messages, thus adequate for mutual recognition; that is, if neighbors are both awake for the duration of one slot, then each neighbor receives some message from the other. While it would be ideal for nodes to discover each other in one slot time, it is quite improbable in practice that that neighbors would have slots so precisely aligned, starting their slots simultaneously. Therefore, implementations of these slotted protocols may stretch the

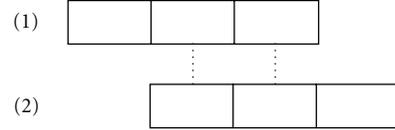


FIGURE 5: Unaligned slot sequences.

length of the awake interval, aiming for sufficient overlap even when the slots are not aligned. To see this, consider an awake interval comprising three slots, that is, three times the minimum-length time needed for mutual recognition.

Figure 5 shows awake periods for two nodes, (1) and (2), which do not have aligned slots. Because each interval's length comprises three slots, a sufficient condition is that these two awake intervals satisfy: *the center slots of each have some (even small) overlap*. That condition guarantees that the two intervals overlap by at least the duration of one slot, indicated, for instance, by the dotted vertical lines in the figure. The condition suggests an approach to designing a neighbor discovery protocol. First, assume that slots are aligned; then design a protocol that guarantees, neighbors eventually are both awake during some slot (the subject of Section 5). Second, when implementing this protocol, prefix any scheduled contiguous sequence of awake slots with one extra awake slot, and similarly add a suffix of one extra awake slot to the sequence. The idea from Figure 5 then overcomes the fact that slots are not aligned in practice. Thus the consequence of using simplistic model of aligned slots is that neighbor discovery protocol results, presented in later sections, will be degraded by some (hopefully constant) factor in an implementation of the protocol. Further motivations for extending the awake periods (due to collisions and other phenomena) are discussed in the appendix. The work of [6] suggests another way to deal with unaligned slots: at the start and end of each slot, a beacon is transmitted, which is enough to trigger discovery. The implementation findings reported in [6] using the slot abstraction state that only in 2% of cases did the actual discovery time exceed that predicted by analysis and simulation based on aligned slots.

5. Protocols

Protocols for neighbor discovery exploit three basic themes, though a variety of constructions combine the themes and emphasize them differently. First, a sensor node can use randomness to influence behavior. Random choice of which slots are awake or sleeping is a probabilistic method of obtaining mutual recognition. Second, there are patterns of awake slots that guarantee neighbor discovery when all nodes use them. Third, a node can remain awake for a number of consecutive slots to assure neighbor discovery.

Whatever the technique used for obtaining discovery, an important question is what should be done after discovery? Most papers gloss over this question, though it deserves some explanation. Suppose that neighbors p and q achieve mutual recognition in some slot at time t . One design choice would be for both p and q to record the fact of a new neighbor in local state variables and then continue with

the discovery protocol after time t , each perhaps discovering other neighbors. If the discovery protocol also exchanges some extra information, then with each discovery a node may also obtain the schedule for each neighbor. Thus, node p would have a table of its neighbors and their sleeping schedules. A different design choice would be for nodes to change behavior following the event of neighbor discovery. Thus, at time t when p and q discover each other, at least one of the two nodes changes its sleeping schedule so that thereafter p and q have identical sleeping schedules. We call this a *merge* event. If two nodes merge, at least one of them switches its sleeping schedule (or changes its current slot number within the schedule). While this may seem simple, it can be more involved after a history of merge events: perhaps two connected components A and B each contain multiple nodes, that is, $|A| > 1$ and $|B| > 1$. Now if a node in $p \in A$ and a node in $q \in B$ discover each other, how should a merge event proceed? If p is to adopt q 's schedule, then does p “move” from A to B , or should all nodes of A and B merge into one schedule? The latter choice would imply some kind of distributed algorithm to effect the schedule change, which is problematic for a low duty-cycle WSN application. We leave the details of merging questions open in this article, due to the lack of literature on this topic.

Some WSN applications make use of the radio's local broadcast ability: with one transmission, a node can send data to all its neighbors. If this feature is desired, then a merging protocol is superior to a nonmerging protocol, because after merging, all neighbors would be awake to receive a local broadcast (because they would have the same sleeping and awake schedules). For applications that use only unicast, a non-merging protocol could be sensible. A hybrid of these two approaches would be a discovery protocol for single-use deployments, where nodes engage in non-merging neighbor discovery for some fixed time period, and then all nodes switch to a common sleep schedule.

5.1. The Birthday Protocol. The idea of the birthday protocol is dual to the randomization strategy behind CSMA/CA in 802.15.4 for sensor networks. Recall that for CSMA/CA, a node delays for some random interval before attempting transmission. The purpose of delay is to increase the probability of finding a transmission time that avoids collision, that is, neighbors do not transmit simultaneously. In contrast, the goal of the birthday protocol is to use random selection between awake and sleeping states so that neighbors are awake simultaneously. The work of [7] proposed the birthday protocol for low-power communication, based on transitions between three node states. Entering a state amounts to starting either an asleep or awake interval of fixed duration, which is effectively a slot. At the start of each slot, a node chooses with probability p_s , p_t , and p_ℓ whether the state for that slot is to be sleeping, transmitting, or listening. During a transmitting slot, a node broadcasts a discovery message. The work of [7] refines this approach further by arranging for nodes to have different modes, using timing parameters tuned for performance, which tune the frequency of entering a transmitting state.

A notable feature of the birthday protocol is that it does not require neighbor discovery as such. (Though [7] is oriented to neighbor discovery, we observe that it could directly be used as a MAC protocol in which nodes may sleep.) Nodes could use this protocol to send and receive messages, without needing any particular sleeping schedule, because the duration of sleeping is a random variable. An open question is how nodes should behave in birthday protocol following discovery—the argument of the authors of [7] is that the birthday protocol can be memoryless, with no durable consequences of discovering a neighbor. In contrast, other papers [7] as a discovery mechanism do suggest that discovering a neighbor could modify subsequent protocol behavior. We thus consider as an open problem how the randomized technique of the birthday protocol could be used for more durable discovery and scheduling. It seems that merging would be possible, though the behavior of a merged set of nodes should be the same with respect to random choices after the merge. That is, when nodes merge, they should adopt a common seed for a pseudorandom number generator, so that they coordinate sleeping.

Analysis of the essential ideas of the birthday protocol appears in [8, 9] for a 1-hop network (fully connected) with application to *ad hoc* networks. We did not find analytic results on the birthday protocol for multihop networks. Analysis in [9] derives a time period after which, with high probability, all neighbor discovery is completed (this assumes that all nodes start at approximately the same time). Analysis in [8] compares the energy cost of the simple birthday protocol, of the kind outlined here, to a round-robin birthday protocol.

5.2. Brute Force. The simplest deterministic protocol for neighbor discovery is the periodic schedule of n slots, with the first $\lceil (n+1)/2 \rceil$ of these being awake and the remaining slots for sleeping. This can informally be called the “51%” solution, since the idea is to remain on for slightly more than half of a period. No matter how two neighbors are initially *offset* in where their periods begin, mutual recognition is assured because their awake intervals must overlap. Several papers either explicitly or implicitly use this brute force technique or similar [10–12]. Let the periodic interval of n slots be called a *round*. Clearly, neighbors discover each other within one round, which is optimal in terms of the latency of the discovery process. Unfortunately, the duty cycle is at least 50%, which is unacceptable for low-power operation.

A method of reducing the duty cycle below 51% is proposed in [10, 12]. Let $k = \lceil (n+1)/2 \rceil - 1$ and consider a logical division of the initial $\lceil (n+1)/2 \rceil$ awake slots into the first slot and the k subsequent slots. Suppose that r is a divisor of k . Now partition the k slots that follow the first slot of the round into r sequences, labeled f_0, f_1, \dots, f_{r-1} . Each interval f_i consists of r consecutive slots. With this terminology, we consider a transformation of the 51% solution by letting each round begin with one awake slot, but distributing the intervals f_i over r rounds. For example, if $r = 2$, there are two intervals f_0 and f_1 , each having $k/2$ slots, spread over two rounds: in the first round, we have $1 + k/2$ awake slots followed by $n - (1 + k/2)$ sleeping slots;

Slot	1	2	3	4	5	6	7	8	9
	☼	☼							
	☼		☼						
	☼			☼					
	☼				☼				

FIGURE 6: Brute force transformed to lower duty cycle.

in the second round, there is one awake slot followed by $k/2$ sleeping slots, then $k/2$ awake slots, and $n - (1 + k)$ sleeping slots. The schedule for a node is to repeat the pattern of these two rounds. Observe that, except for their first slots, the awake times of the first and second rounds are disjoint.

Figure 6 illustrates an example where $n = 9$, $k = 4$, and $r = 1$, which produces a sequence of four rounds. The figure shows the four rounds as rows of a table, with columns representing slots. The symbol \star indicates that a slot is awake. The duty cycle for this schedule is less than half of the 51% approach, $2/9$. To get some intuition why this translation of brute force into multiple rounds is valid, consider two neighbors p and q that have an offset of six slots and the following scenario. Node p starts with the first round (time and slot numbers are synonymous here). Thus p is awake at times 0 and 1, then asleep until time 9, when its second round begins. Following the patterns of rows for Figure 6 awake times for p are 0, 1, 9, 11, 18, 20, 27, 31, 36, and 37. Node q is awake at times 6 and 7, then asleep until time 15. The awake times for q include 6, 7, 15, 17, 24, 26, 33, 37, 42, and 43. We see that both p and q are awake at time 37, which suffices for discovery. The rationale for the pattern is seen from the table. The “union” of relative awake times for the four rows is the 51% schedule.

The translation of brute force to a scheme that spreads awake slots over time does reduce duty cycle, but at a cost: the time needed to guarantee discovery is larger: the discovery time is in the worst case r times greater (see analysis in [10]). This translation partitions a consecutive sequence of slots into an interrupted, irregular pattern of waking and sleeping. In many of the other discovery protocols we see a similar idea, where an irregular sleep pattern is used to obtain low duty cycle. Depending on application constraints, an irregular sleep pattern may not be useful for the application’s tasks of sensing, computing, and communicating. At least the first slot of each round in the transformed brute force approach occurs periodically. A hybrid adaptation of the idea, combining the translation and randomized selection, would be a schedule with two awake slots, one at the start of each round and the other randomly selected from the remaining slots in the round.

A different method to reduce the duty cycle, again based on the brute force technique, is used in [11]. The basis for their method is to have one awake slot at the start of every round; however this is augmented by sometimes using the 51% solution in a round. For example, once every n rounds, a node is awake during $\lceil (n + 1)/2 \rceil$ consecutive slots, this achieves a $2/n$ duty round, at the cost of increasing the worst-case discovery time of the protocol. Because a node would

need to be awake for $n/2$ consecutive slots, this method might be unsuited to energy-harvesting platforms.

5.3. Intersecting Designs. Finding schedules of awake and sleeping slots to guarantee neighbor discovery has a combinatorial interpretation. The problem is to devise schedules with minimum duty cycle that are self-intersecting with respect to any rotation. Suppose that π denotes the indices of awake slots in an n -slot round. An example of this for $n = 16$ is $\pi = \{0, 3, 4, 12\}$. A k -rotation of π is obtained by adding k to each index, modulo n , denoted by π^k . Thus the 5-rotation of the example produces $\{5, 8, 9, 2\}$. The combinatorial task is to find minimum size π such that $\pi \cap \pi^k \neq \emptyset$ for $0 \leq k < n$. Finding such a sequence readily provides a schedule so that no matter how neighbors are offset, a common awake slot is guaranteed within a complete round.

While self-intersection for any k -rotation yields discovery within a round, the neighbor discovery problem in general does not require discovery within one round. Depending on how important discovery is to the application, weaker combinatorial problems, perhaps asking for intersection over a history of rounds, would be satisfactory. The results surveyed in this subsection target guaranteed discovery within one complete round. Also, note that the problem of finding self-intersecting sequences need not be restricted to all nodes using *the same* sequence. We may distinguish between symmetric solutions, where all nodes use the same sequence, and asymmetric solutions where nodes use different sequences from a set \mathcal{S} of patterns, such that any $S \in \mathcal{S}$ is guaranteed intersection with T^k for $T \in \mathcal{S}$. We concentrate first on symmetric solutions and return in the next subsection to asymmetric solutions.

Lower bounds on the number of awake slots needed for self-intersection are explored in [13, 14]. The problem requires $\Omega(\sqrt{n})$ slots to be awake for discovery. In effect, the solution schedules with $O(\sqrt{n})$ awake slots correspond to the match-making work cited in Section 3. The schemes proposed in [13] are combinatorial designs, which have other applications in discrete mathematics. However, the first paper to explore such schedules for discovery is [15], which used the quorum idea, similar to the work of [5] on mutual exclusion. The works of [13, 14] improve on the quorum construction with lower power and considering multihop topologies and investigating randomized schedules with high probability of self-intersection with rotation.

Designs based on self-intersecting schedules are chiefly of interest to applications that need to minimize discovery latency, while also minimizing power usage. Whereas the brute force approach has a duty cycle of at least $1/2$ to minimize latency, the existence of self-intersecting schedules would argue that power can be substantially reduced. However, there are some considerations for using these schedules with low duty cycles. Suppose that a 0.1% duty cycle is needed, and ignore constants in the $O(\sqrt{n})$ bound, for estimation purposes. To obtain the 0.1% duty cycle, we require $\sqrt{n}/n = 1/1000$; hence $n = 10^6$. A deterministic self-intersecting schedule could impose some complex representation challenges for software implementation, depending on platform resource. Also, the schedule will be irregular, which may

not be compatible with desired application behavior. Finally, the platform typically puts a practical lower bound on the duration of a slot, typically in tens to hundreds of milliseconds: the duration of a round, and therefore worst-case discovery time, will be on the order of several hours. (This last observation merely illustrates that there is a tradeoff between low duty cycle and discovery latency.)

5.4. Coprime Schedules. The last type of protocol we survey is based on periodic rounds that have relatively prime length with respect to neighbors, reprising an idea mentioned in Section 3. Thanks to the Chinese Remainder Theorem [16] if neighbors p and q have rounds in which the first slot is awake and the rest sleeping, and the two round lengths are relatively prime, then discovery is guaranteed. Put more formally, let c_p and c_q be the respective number of slots in the rounds of p and q . Numbers c_p and c_q are *coprime* if their greatest common divisor is 1. The latency for mutual recognition is, in the worst case over any offset between the two rounds, $c_p \cdot c_q$.

Several observations concerning coprime schedules affect its suitability for WSN deployments. First, nodes need individualized programs so that each node has a round length coprime to all its neighbors. This can be done by assigning each node its own prime number; however this adds to deployment cost (and may be error-prone). Second, the duty cycle for a schedule of c_p rounds is $1/c_p$; if different primes are used, the asymmetry of different duty-cycle rounds in the network will depend on the set of primes chosen (though, after fully merging, all nodes could use a common round). Third, the schedule's arrangement of periodic rounds with one awake slot is a good fit for applications performing periodic sampling, perhaps by extending the one awake slot to an awake interval of slots. Note that in coprime scheduling we see a tradeoff between latency and duty cycle: lower latency is obtained by using smaller primes, but this entails higher duty cycle.

Several papers propose coprime scheduling for neighbor discovery [6, 10, 11], using different techniques that deal with the possibility that neighbors were given the same prime number for their rounds. The choice of a prime can be dynamic, by random selection. That enough does not ensure that neighbor rounds have coprime length, because there remains the possibility of unlucky random choices that deal with the same prime to a pair of neighbors. The technique proposed in [10] is to repeat the random prime selection process. For example, p may start with a randomly selected c_p , use that for $k \cdot c_p$ rounds, and then choose again randomly another prime value for c_p . The value k can be tuned, and random selection is confined to a set of two coprime numbers $\{z, z + 1\}$ to get an expected discovery latency in $O(z^2)$ slots. Two deterministic approaches are investigated in [6, 11]. The idea of [6] is to assign a pair of primes $\{c_p, d_p\}$ to any node p . The sleeping schedule is modified so that a slot t is awake when either $t \bmod c_p$ or $t \bmod d_p$ is zero. Because $c_p \neq d_p$, even if neighbors have the same pair of primes, discovery is assured. Tuning the selection of primes for a desired duty cycle, and a refinement using a triple of primes per node is also proposed in [6]. An advantage of multiple primes is that duty cycles can be adjusted at finer grain, because the

duty cycle is approximately $1/c_p + 1/d_p$; moreover, different nodes can have distinct duty cycles, if an asymmetric schedule is useful to the application. The remaining technique, proposed in [11], to overcome using the same prime at different nodes is the one mentioned at the end of Section 5.2 proposed (a transform of brute force) where one in every c_p rounds is a round with the 51% solution.

6. Metrics

Two criteria for evaluating a neighbor discovery protocol are latency and duty cycle. Latency is informally the time taken to discover a neighbor. Formally, latency can be measured in several ways: (i) the {mean, median, maximum} times for two neighbors to mutually recognize, taken over all nodes and all initial configurations (of offsets and protocol parameters, such as prime number assignments) between the nodes; (ii) the mean time for a node to discover all its neighbors, taken over different offsets, nodes, and topologies; and (iii) the mean and maximum time to “termination”, that is, when all nodes have discovered all their neighbors, again taken over all initial conditions and protocol parameters. In addition to these systemic questions of latency, one could ask about time taken for a new node added to a network, or a topology change, to be recognized. We focus on latency in this section, particularly the worst case and distributions of discovery times.

The work of [11] starts with the observation that, for certain worst-case latencies in the class of deterministic protocols, the characteristics of optimal schedules, with respect to the number of awake slots, were shown in [13]. An optimal schedule's number of awake slots can be used as a benchmark for evaluating different protocols. The authors of [11] propose that a combined metric, the *power-latency product*, be a basis for comparison. Their analysis for power-latency product shows that the quorum protocol [15] and the pair-of-primes protocol [6] are at least a factor of two greater than the optimum, whereas the single prime protocol of [11] is a factor $3/2$ greater than the optimum. Another paper [12] suggests that randomizing the choice of which slot is awake (in the transform at the end of Section 5.2) may have better *mean* time to discovery than [11].

Is the power-latency product a good target for optimization? When either factor, time to discovery, or duty cycle is held constant, the other factor indeed should be minimized. The attractiveness of the power latency product is its neutrality with respect to tradeoffs for the several protocols surveyed: if latency is to be reduced, using several of the techniques explained in this article, it is at the cost of increased power utilization (for an optimal protocol), due to higher frequency of scheduled awake slots. If these two factors are commensurate, then tuning latency does not change the metric. Power-latency as a metric is similar to the delay-power product for design of switching circuits, where higher power can be necessary for faster response. One aspect missed in the power-latency comparisons in [11] is the distribution of discovery latency times, rather than comparing by analysis of the worst-case latency.

6.1. Simulations. The performance behavior of discovery has been evaluated in [6, 11] by simulation and implementations. The Disco paper [6] explores several questions by simulation, but left open the issue of how discovery times are distributed. To better understand how protocols surveyed in Section 5 compare with respect to discovery time distribution, we simulated them. While a number of sensor network simulators are available, for instance, [17, 18], these tools simulate protocols at a low level. How discovery times are distributed is a simpler question, readily answered by a discrete event simulator; we wrote a small simulator in Python for our investigation.

The simulator was constructed as follows. One iteration of the simulator runs a given discovery protocol on a fixed number of immobile nodes from a start state in which nodes have freshly generated random offsets to an end state in which all edges have been discovered. Each protocol evaluated was run through 1000 simulator iterations; the time at which each edge was discovered was recorded. For this set of simulations, we fixed the duty cycle to be approximately 1% on a clique topology of 48 nodes (for those protocols that could not realize a duty cycle of 1% a slightly lower approximation was used). For all the protocols, we chose parameters to get symmetric behavior (the same duty cycles), meaning that we chose a single prime for all nodes for protocol [11] and drew each prime randomly from a set of four candidate primes to simulate Disco [6], following the authors advice to choose for each node one prime near the reciprocal of the duty cycle (100 for the 1% duty cycle) and a second, larger prime to create a duty cycle closer to 1% (e.g., a valid 1% duty cycle prime pair would be (101, 10103)). To achieve the 1% duty cycle for the 51% solution, we used the transformation of one extra slot per round explained in Section 5.2. The initial state of a node, that is, where the node started the simulation with respect to the slot schedule of awake/asleep, was uniformly random for each of the 48 nodes. The resulting edge discovery times, totaled over the 1000 iterations, were then grouped into buckets (each representing 100 consecutive slot times); each bucket is a point in the graph, and lines through the points show the distribution of discovery counts on the y -axis versus slot times on the x -axis. Note that these simulations *do not* use any kind of merging strategy—each node adheres to its schedule throughout the simulation.

Figure 7 shows the results. The results show how different protocols exhibit different discovery time distributions. Deterministic approaches, quorum and the 51% protocol, end at a certain slot for all simulation runs (this ending slot represents the worst case). The Birthday protocol's distribution confirms the classical combinatoric distribution one would expect (which can be analytically predicted by probability theory). The long tail on Disco is due to the effect of large primes, explained here in after.

The 51% protocol differs from the other deterministic protocols by its discovery occurring at a relatively constant rate. This protocol's neighbor-discovery time is directly proportional to the relative initial offset between neighbors. Since the relative offsets are uniformly distributed initially,

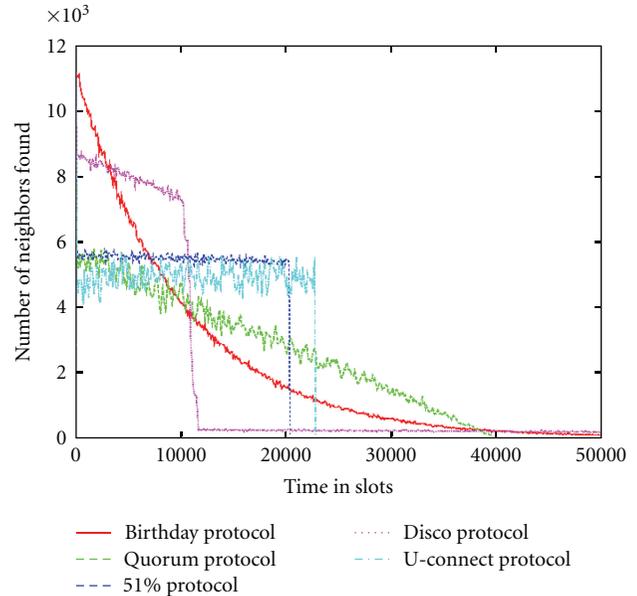


FIGURE 7: Discovery time distributions from simulation.

the discovery times are uniformly distributed (up to the worst case).

The distribution of [11] is relatively flat because all nodes use the same prime number for their schedules. Recall from Section 5 that a node is awake once every p slots (p being the prime), but after $p - 1$ such iterations are awake for $p/2 + 1$ slots. Hence, a node is awake for short intervals frequently and long intervals infrequently. Due to the uniform distribution of initial states and offsets, the long intervals are uniformly distributed over the simulation. In such a simulation, discovery only occurs when a long interval is involved, hence the distribution follows a flat curve. (We also ran simulations of [11] with many distinct primes, and the resulting distribution had a pronounced downward slope, reflecting the diversity of opportunities for discovery, which reduced over time.)

The Disco distribution [6] shows larger discovery rate in the beginning; the protocol gives an advantage to neighbors with distinct smaller primes. Over the course of the simulation, neighbors with smaller primes are simultaneously awake in numerous slots; however, we only count the first such slot for discovery. Hence, we see more discoveries toward the start of a simulation than later. After the edges discovered through such intersections (driven by smaller primes) are exhausted, we then see the distribution drop to a long tail. In the long tail, neighbors get mutual recognition during a slot governed by at least one larger prime. The distribution's flat long tail is explained by realizing that both of neighboring node's prime pairs and offsets are assigned randomly; once the choice of (large) primes and offset is set at the initial state, the distribution of meeting times between distinct large primes tends to be uniform over the simulation.

Note that the Quorum protocol's distribution is roughly linear, and the downward slope appears to be the same as the initial part of the Disco distribution. The explanation is the same: over the course of the simulation, a given pair

of neighbors (p, q) are awake simultaneously at numerous times. The simulation only counts the first intersection of (p, q) for discovery; the total number of opportunities for edge discovery diminishes over time until all edges are discovered.

7. Conclusion

This article collected the ideas prevalent in recent literature on the neighbor discovery problem for wireless sensor networks. The basic methods of randomization and combinatorial sequences appear in several papers, with the most recent papers beginning deeper, hybrid constructions for neighbor discovery. One technique not yet fully explored could be sharing of discovery information with neighbors. If p and q have discovered each other, and then p discovers r , there is some probability that r is also a neighbor of q , and this could be exploited to accelerate a (q, r) discovery. Generally, heuristics suitable to particular topologies or neighbor densities have not been explored in the literature.

In Section 3 it was observed that the case of unidirectional links in neighbor discovery is not explicitly researched in the papers we have surveyed. To give the reader some idea of the considerations involved, suppose that the simple linear topology of Figure 1 is unidirectional, where node (1) cannot receive from any other node, (2) only hears from (1), (3) only hears from (2), and (4) can only receive from (3). In principle, it should be possible for each node except (1) to learn of the upstream neighbor and adapt its schedule accordingly; as a result, all nodes may eventually adopt the schedule of (1). The case of a cyclic network topology with unidirectional links would be more challenging, apparently requiring some way to break symmetry. These considerations suggest that complete solutions to the case of unidirectional communication could be complex and perhaps unsuited to the simple sensor network platforms.

Though protocols surveyed in Section 5 can work for multihop WSNs, we did not find analysis or extensive measurements of behavior for the multihop case. The issue of merging after discovery, relatively simple for a single-hop network, becomes more intricate for multihop networks.

Only two of the papers surveyed [6, 11] report empirical work with implementation. Further practical experience would be most helpful to prioritize research issues in this area. The case study reported in [11] implemented neighbor discovery in the FireFly Badge platform, used in the Sensor Andrew project [19]. The platform was embedded into a key-chain form factor and then used to support a social networking application.

Appendix

A. Platform and Hardware Considerations

The purpose of this appendix is to provide background on hardware features and application requirements, which constrain the implementation of neighbor discovery. Some sensor networks are temporary, perhaps used for specific and short-term purposes; others are long-lived, usually requiring

maintenance or manual procedures to set up and modify the deployment. Additionally, application needs can limit the range of schedules for awake and sleeping periods. Hence, some considerations are driven by low-level characteristics of a particular radio chip, whereas others are dictated by high-level, use-case scenarios for the WSN. Implementation issues are driven by engineering issues and by application requirements or deployment experiences.

A.1. The Slot Model. The assumption of an aligned schedule of slots is briefly justified in Section 4 under the assumption that one slot is adequate for mutual recognition. How is such a slot time established? There are two constraints on the interval for nodes being awake, a lower bound due to platform and protocol timing issues, and a bound derived from power and application requirements. We look first at the timing issues for a hypothetical case, inspired by figures from representative hardware in WSNs [20–24].

Suppose that a radio frame (including synchronization prefix) is 128 bytes and effective transmission rate is 19.2 Kbps; the transmission time for a frame is thus approximately 53 milliseconds. For mutual recognition, each of two neighbors needs to transmit to the other, so a slot must be at least 106 milliseconds long. However such superficial analysis ignores important factors. (1) to compensate for the lack of alignment, extra radio on time is needed as shown in Figure 5. (2) If two neighbors transmit first in a slot, the result is message collision and both attempts fail. (3) A node cannot count on discovering a particular neighbor in a slot. There could be more than one neighbor to discover, exceeding the estimate of 106 milliseconds for mutual recognition (for a set of neighbors). (4) Awake slots could be used for other communication purposes than discovery, so during normal operation discovery is only one role of communication. (5) WSNs typically use CSMA/CA to avoid collision by randomized delay before transmission, and planning for the delay enlarges the awake interval. (6) Even if slots are aligned (which could occur after neighbor discovery and adjustment of node schedules), clocks can drift and the timing of schedules may deviate from the ideal. Other sources of timing error include device driver nondeterminism, perhaps due to interrupts from sensors hosted on the WSN platform. (7) In realistic deployments, messages are occasionally corrupted by a variety of noise effects. such messages are lost.

A realistic estimate of adequate slot time therefore includes different factors and likely depends on empirical measurements of protocol properties. At best, an estimated slot time is probabilistic. With some probability, mutual recognition will fail to occur when awake slots overlap. The latency for neighbor discovery could thereby be prolonged.

A.2. Duty Cycle. Application requirements as well as platform properties constrain power management in WSNs. A simple example illustrates power management. Suppose that a node is powered by a 1200 mAh battery, with the processor consuming 2 mA under full power, and the radio consuming 20 mA when turned on for listening (events of receiving and transmitting may use slightly more power). We estimate the

lifetime of a sensor node to be a little more than two days if processor and radio are continuously active. However using sleep modes, the processor's power consumption drops to $2\ \mu\text{A}$ and the radio's power draw is $1\ \mu\text{A}$. The lifetime of a sensor node is estimated to be decades if always in sleep mode. If an implementation uses duty cycling, with nodes at full power 1% of the time and sleep mode 99% of the time, the estimated lifetime for this hypothetical example would be half a year, and with full power only 0.1% of the time, a lifetime estimate is somewhat over five years. Using 0.01% duty cycling puts the lifetime estimate into decades. Such estimates ignore communication failures, power needs of sensing, and computation by the application. Nonetheless, if an application requirement is to run for years in the field on battery power, then exploiting sleep modes of processor, radio (and perhaps attached sensors) is crucial.

In the slot model, for a history \mathcal{H} of consecutive slots, a node will generally alternate between sleeping and awake states. Let T_{sleep} denote the number of sleeping slots and T_{awake} the number of awake slots in \mathcal{H} , so $|\mathcal{H}| = T_{\text{sleep}} + T_{\text{awake}}$. The *duty cycle* for \mathcal{H} is defined as the ratio $T_{\text{awake}}/|\mathcal{H}|$. In most cases, discovery protocols use similar scheduling patterns for all nodes, so the duty cycle of one node is representative of the duty cycle for the protocol. Some neighbor discovery protocols have irregular or random sleep schedules, in which case the duty cycle is evaluated for (asymptotically) large values of $|\mathcal{H}|$. Duty cycle is a useful metric for protocol comparison. However, lower duty cycles generally imply longer discovery times. If rapid neighbor discovery time is desired, extremely low duty cycles may not be possible.

A.3. Radio. The consensus of papers investigating neighbor discovery is that the radio uses a single frequency (even radio chips programmable for multiple frequencies allow only one to be used at a time), a node can transmit to all neighbors with a single message, and a MAC layer takes care of collision avoidance, usually by some random delay mechanism. Some special considerations of radio chips have influenced the design of a few neighbor discovery results. A number of low-level MAC considerations (framing and rate of beaconing) are studied in [25] especially for neighbor discovery. This level of attacking the problem is at a deeper layer than we consider for survey. Nonetheless, some properties of the radio are described here in after, because they influence the evaluation of higher layer protocols.

A.4. Warmup Delay. In practice, nodes cannot switch instantly from a sleeping state to a fully awake state. Radio chips typically need to activate frequency oscillators during the power up sequence, and this can introduce a delay on the order of a millisecond for WSN platforms (a typical WiFi platform's transition time is on the order of a hundred milliseconds). Though the power up sequence can begin during the last part of a sleeping slot, the timing depends on the node's ability to schedule activities in real time, mediate conflicts with sensor activities, and so on. Similarly, if the duty cycle for an application is geared to the duty cycle for neighbor discovery, there can be extra tasks needed to power

up and calibrate sensors, should they have sleep states. The time needed for radio warmup could make protocols with contiguous awake slots more attractive than another protocol with similar duty cycle and latency but more scattered awake slots.

A.5. Clear Channel Assessment. Some radio chips have a feature, made available to the device drivers of the system, to sample radio activity briefly at very low power. This function is called *clear channel assessment* (CCA). The power cost of a CCA sample is negligible compared to the power of fully operating the radio, and additionally has no significant warmup delay. This implies that substituting CCA slots for awake slots in a neighbor discovery protocol can significantly improve power consumption, an approach used in [11]. Additionally, periodic CCA sampling can also be used to determine when a neighbor is transmitting, at which point the radio can be powered to receive it, functionality utilized by protocols like B-MAC [26]. However, B-MAC and similar protocols require long message preambles to account for the delay introduced by recipients powering on the radio. Depending on application patterns of sending and receiving messages, B-MAC is able to approach 1% duty cycles without needing sophisticated neighbor discovery algorithms; some further refinements using scheduling of transmissions and lightweight neighbor discovery (but not adapting all nodes to a common schedule) are surveyed in [27], which can attain 0.1% duty cycles. The SCP-MAC protocol of [27] is especially relevant to discovery, since it shows how sampling using CCA can lead to discovery of neighbors, synchronizing schedules, and eliminating long preambles; however, at least some level of transmission is needed to sustain discovery for a dynamic network (sampling is not enough without transmissions). The SCP-MAC protocol may be more complex for implementation than some platforms can afford. All the low-power MAC results depend on having CCA, the ability to wake quickly from sleeping, setting frame preambles or other similar mechanisms; they also presume all nodes use the same protocol. The work of [6] points out that higher-level discovery protocols, of the kind surveyed in this article, can tolerate greater heterogeneity of MAC protocols (perhaps some nodes using low-power MAC and others not) and even permit asymmetric operation where not all nodes use the same duty cycle.

A.6. Clock. Timing is crucial to neighbor discovery. Processors typically have at least two clocks, which are essentially programmable counters that generate interrupts. In sleep modes, a counter which is based on an external oscillator (outside of the processor chip) continues to run at extremely low power, generating an interrupt when a designated value is obtained. Using the processor clocks, a virtual clock service can be programmed, supplying alarm signals and query functions to applications. Further, through messaging, the clock services of all the WSN nodes can be synchronized, so that all nodes have the same "virtual time" available.

A trivial solution to neighbor discovery is possible with synchronized clocks: nodes all wake at designated clock

times, say when the clock modulo some value K is zero, and this enables discovery. Clocks might be synchronized using dedicated hardware; for example, a powerful transmitter on a special frequency that periodically sends a time beacon could synchronize all node clocks. However, in the absence of a specialized setup, synchronizing clocks is only achieved through message exchange between neighbors. Thus the argument for using this trivial solution is circular.

Other practical concerns using clocks are jitter and skew. Some software may insert delay processing clock-generated interrupts, which degrades the timing of slot boundaries. Also, counter hardware is often selected for low cost and may deviate from ideal behavior; typical counters could run at rates $(1 \pm 10^{-5}) \cdot t$ where t is the rate of a corresponding neighbor's counter. Clock skew thereby degrades the assumption that slots of neighbors have the same start times. Periodic resynchronization and other countermeasures can improve performance of slot-based protocols.

A.7. Application Requirements. Protocols for neighbor discovery all specify schedules of sleeping and awake slots that either guarantee mutual recognition or provide for mutual recognition with some probability. Every WSN application has one or more specified behaviors, which also require awake periods for processing and communication. The ideal situation would be to find compatible schedules between application needs and discovery protocols, since doing so would leverage warm-up overhead for both purposes and share processing cycles and bandwidth during the awake slots. When implemented, applications dictate or exploit architectural features of the target WSN deployment: some applications depend on the continuous availability of a base station whereas others record data in local flash memory for later extraction.

A.8. Event-Driven versus Periodic. Two extremes in application requirements are event-driven and periodic sampling styles. In the event-driven style, sensors detect phenomena and trigger software handlers, which may then initiate data transfer protocols to alert the base station. In the periodic sampling style, nodes poll sensors for environmental values and engage in communication rounds at prescribed times. Many hybrids of these two styles are possible; for instance, nodes may accumulate sensed values which are triggered by external events (like the event-driven style) but only transmit messages to report their sensor readings at regular, periodic times. Extremely low duty-cycle operation generally favors a periodic sampling style. In event-driven systems with higher duty-cycles, the low-power neighbor discovery schemes may be inappropriate, particularly if low-power protocols such as B-MAC [26] are acceptable.

A.9. Tethered Applications. Neighbor discovery is simplified when applications enjoy the assistance of a continuously powered subnetwork with adequate communication coverage. In such applications, nodes are of two types, continuous power and those that need to sleep for power management. Provided every sleeping node p is within

range of a continuously powered node, then when p wakes up, mutual recognition with a powered node occurs. The simplest construction puts the powered nodes into a connected subnetwork, or backbone, which can assemble all neighborhood information and collect this information at a base station (or alternatively work on the information with a distributed algorithm). The powered subnetwork dictates a power schedule to the other nodes, which arranges them to be awake concurrently for neighbor recognition.

Tethered applications use an *asymmetric* architecture of nodes, as opposed to a fully symmetric network where all nodes have the same capabilities and power constraints. An architecture need not be tethered, with some nodes continuously powered, to be asymmetric. There can be applications where some nodes use power harvesting, some use small batteries, and some have larger battery reserves for higher duty cycles. We observe here that asymmetry in the architecture is possible mainly to distinguish between later usages of symmetry in the construction of discovery protocols, where symmetry and asymmetry refer to algorithmic properties, such as having equal or unequal duty cycles among the nodes running a discovery protocol.

A.10. Deployment. The costs of deploying a WSN application, which encompass the programming of the individual nodes, installing sensor nodes in the field, testing components and connectivity, and managing battery supplies and assorted inventory chores, are “hidden costs” of software designs and application architectures. An example of this is the difference between a design where all nodes have identical programs from a design where each node should have an individualized program. With identical programs, setup costs and replacement costs are reduced; in fact, programming over-the-air is more efficient if all nodes install the same code image [28]. An advantage of a randomized discovery protocol can thus be lower cost of deployment.

References

- [1] E. F. Moore, “The firing squad synchronization problem,” in *Sequential Machines*, E. F. Moore, Ed., pp. 213–214, Addison-Wesley, 1964.
- [2] L. Gasieniec, A. Pelc, and D. Peleg, “Wakeup problem in synchronous broadcast systems,” in *Proceedings of the 19th Annual ACM Symposium on Principles of Distributed Computing (PODC '00)*, pp. 113–121, 2000.
- [3] S. J. Mullender and P. M. B. Vitányi, “Distributed match-making,” *Algorithmica*, vol. 3, no. 1, pp. 367–391, 1988.
- [4] D. K. Gifford, “Weighted voting for replicated data,” in *Proceedings of the 7th ACM Symposium on Operating Systems Principles (SOSP '79)*, pp. 150–162, 1979.
- [5] M. Maekawa, “A \sqrt{N} algorithm for mutual exclusion in a decentralized systems,” *ACM Transactions on Computer Systems*, vol. 3, no. 2, pp. 145–159, 1985.
- [6] P. Dutta and D. Culler, “Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications,” in *Proceedings of the 6th International Conference on Embedded Networked Sensor Systems (SenSys '08)*, pp. 71–84, 2008.
- [7] 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.
- [8] S. Fang, S. M. Berber, and A. K. Swain, “Analysis of neighbor discovery protocols for energy distribution estimations in wireless sensor networks,” in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 4386–4390, 2008.
- [9] 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.
- [10] T. Herman, S. Pemmaraju, L. Pilard, and M. Mjelde, “Temporal partition in sensor networks,” in *Stabilization, Safety, and Security of Distributed Systems (SSS '07)*, vol. 4838 of *Springer Lecture Notes in Computer Science*, 2007.
- [11] A. Kandhalu, K. Lakshmanan, and R. Rajkumar, “U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol,” in *Proceedings of the 9th International Conference on Information Processing in Sensor Networks (IPSN '10)*, pp. 350–361, April 2010.
- [12] M. Bakht and R. Kravets, “SearchLight: asynchronous neighbor discovery using systematic probing,” *Mobile Computing and Communications Review*, vol. 14, no. 4, pp. 31–33, 2010.
- [13] 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 (MOBIHOC '03)*, pp. 35–45, June 2003.
- [14] M. Bradonjić, E. Kohler, and R. Ostrovsky, “Near-optimal radio use for wireless network synchronization,” in *Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS '09)*, vol. 5804 of *Springer Lecture Notes in Computer Science*, pp. 15–28, 2009.
- [15] Y. C. Tseng, C. S. Hsu, and T. Y. Hsieh, “Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks,” in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communication Society*, 2002.
- [16] I. Niven, H. S. Zuckerman, and H. L. Montgomery, *An Introduction to the Theory of Numbers*, John Wiley & Sons, New York, NY, USA, 1991.
- [17] P. Levis, N. Lee, M. Welsh, and D. Culler, “TOSSIM: accurate and scalable simulation of entire TinyOS applications,” in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys' 03)*, pp. 126–137, November 2003.
- [18] V. Shnayder, M. Hempstead, B. R. Chen, G. W. Allen, and M. Welsh, “Simulating the power consumption of large-scale sensor network applications,” in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 188–200, November 2004.
- [19] Sensor Andrew, <http://www.ices.cmu.edu/censcir/sensor-andrew/>.
- [20] nRF24 series data sheet, Nordic Semiconductor, <http://www.nordicsemi.com/>.
- [21] CC2500 data sheet, Texas Instruments, <http://www.ti.com/>.
- [22] CC2420 data sheet, Texas Instruments, <http://www.ti.com/>.
- [23] MSP430 data sheets, Texas Instruments, <http://www.ti.com/>.
- [24] ATMEGA128 data sheet, <http://www.atmel.com/>.
- [25] M. Kohvakka, J. Suhonen, M. Kuorilehto, V. Kaseva, M. Hännikäinen, and T. D. Hämäläinen, “Energy-efficient neighbor discovery protocol for mobile wireless sensor networks,” *Ad Hoc Networks*, vol. 7, no. 1, pp. 24–41, 2009.
- [26] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 95–107, November 2004.
- [27] W. Ye, F. Silva, and J. Heidemann, “Ultra-low duty cycle MAC with scheduled channel polling,” in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 321–334, November 2006.
- [28] J. W. Hui and D. Culler, “The dynamic behavior of a data dissemination protocol for network programming at scale,” in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 81–94, November 2004.

Research Article

Subjective Logic-Based Anomaly Detection Framework in Wireless Sensor Networks

Jinhui Yuan,^{1,2,3} Hongwei Zhou,^{1,2,3} and Hong Chen^{1,2}

¹Key Laboratory of Data Engineering and Knowledge Engineering, MOE, Beijing 100872, China

²School of Information, Renmin University of China, Beijing 100872, China

³Institute of Electronic Technology, Information Engineering University, Zhengzhou 450004, China

Correspondence should be addressed to Jinhui Yuan, jcyjh@126.com

Received 15 June 2011; Revised 25 September 2011; Accepted 28 September 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Jinhui Yuan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In existing anomaly detection approaches, sensor node often turns to neighbors to further determine whether the data is normal while the node itself cannot decide. However, previous works consider neighbors' opinions being just normal and anomalous, and do not consider the uncertainty of neighbors to the data of the node. In this paper, we propose SLAD (subjective logic based anomaly detection) framework. It redefines opinion deriving from subjective logic theory which takes the uncertainty into account. Furthermore, it fuses the opinions of neighbors to get the quantitative anomaly score of the data. Simulation results show that SLAD framework improves the performance of anomaly detection compared with previous works.

1. Introduction

Recently wireless sensor networks (WSNs) have been widely used in military surveillance, traffic monitoring, habitat monitoring and object tracking, and so forth [1, 2]. Such networks deploy lots of sensor nodes with sensing, data processing, and wireless communication capabilities in the monitoring area. Sensor nodes are resource-constrained and susceptible to interference from the environment so that their sensing data are often unreliable. Potential sources of anomalous data in WSNs are classified into three categories: faults (errors), events, and malicious attacks [3, 4]. While sensor nodes fail, their sensing data are faulty data [5]. Once the number of faulty data increases, it will bring great influence on the user query. Thus, they should be eliminated or corrected. When some event happens, the sensing data of the nodes in the area are informational data, which are different from the normal data. They should be reported to user for further deciding. The thirdly potential source of anomalous data is attacks which are beyond the scope of this paper. Anomaly detection is considered as a solution to detect faulty data and informational data.

In existing anomaly detection approaches, sensor nodes turn to neighbors to further determine whether the data is

normal while the node itself cannot decide. In this process, existing solutions, including voting algorithms [6, 7] and aggregation frameworks [8–10] which detect anomaly in the process of aggregating data, provide neighbors' opinions being just normal and anomalous. However, no neighbor can always say that the data of the node are absolutely normal or anomalous, and something is neglected by previous works which we call uncertainty. Thus, taking the degree for neighbors' opinions about the data being normal or anomalous into account can more realistically describe the view of neighbors. Consequently, the performance of anomaly detection is able to be improved.

In this paper, we propose SLAD (subjective logic-based anomaly detection) framework, which takes uncertainty into account, to improve the performance of anomaly detection. It includes three phases: preprocessing, self-monitoring, and cooperant detecting. Among them, pre-processing run on sink and self-monitoring execute on each node. After the two phases, sensor nodes send suspicious data to its neighbors to turn to further determine. The third phase is the key of our framework.

The important element of SLAD is ESLB (extended subjective logic-based algorithm), which is the key of the third phase mentioned above. Before plunging into the detail of

ESLB, we first propose SLB (subjective logic-based algorithm) which elementarily describe our work. In SLB, each neighbor gives the quantitative opinion to the suspicious data involving with subjective logic theory. After fusing the opinions of all the neighbors, SLB gets the quantitative anomaly score, which demonstrates the degree of the suspicious data being considered as an anomaly. We extend SLB to ESLB in order to avoid the impact of those neighbors whose data are suspicious, effectively distinguish the faulty data from the informational data, and take the historical spatial correlations of the node and its neighbors into account.

The main contributions of this paper are as follows.

- (i) Proposes SLAD framework which takes the uncertainty of neighbors to the data of the node into account. It redefines opinion deriving from subjective logic theory and can more realistically describe the view of neighbors on the data of the node.
- (ii) Presents SLB and ESLB algorithms. SLB fuses all the opinions of neighbors for the data of the node to get the quantitative anomaly score of the data. We extend SLB to ESLB to improve the performance further.
- (iii) Constructs the experiments to verify the detection performance of the framework we propose. Simulation results show that SLAD framework is effective and gains a lot of performance improvement of anomaly detection compared with the previous approaches.

The rest of the paper is organized as follows. Section 2 summarizes the related work of this paper. Section 3 presents preliminary concepts. Our framework SLAD is introduced in Section 4. Section 5 gives SLB algorithm and its extended algorithm ESLB. Section 6 discusses some problems which are not involve in the above sections. Section 7 describes the experimental setup and evaluates the performance of framework in realistic data set. Finally, Section 8 concludes the paper.

2. Related Work

Lots of efforts have been made in recent years to detect the anomaly in wireless sensor networks. We briefly survey the recent researches relevant to our work as follows.

First category involves voting algorithm and its improved algorithms. Authors in [6] propose majority voting algorithm. If some node v is aware that it's sensing data x maybe anomalous, it sends x to its all one-hop neighbors. Each neighbor v' compares x with its sensing data x' . If the difference is less than the threshold, v' casts a positive vote for v , otherwise casts a negative vote. Node v collects all the votes of its neighbors and gets the determination. If the number of positive votes is more than negative votes, x is thought to be normal, otherwise is anomalous. Based on majority voting algorithm [6, 7] proposes weighted voting algorithm which considers that the neighbors who are closer to the node should have greater weights. Authors in [11] discuss how to detect the faulty (erroneous) data in WSNs. It uses extended Jaccard's coefficient to compute the similarity

degree between sensor nodes and set the different levels for the nodes to set up the correlation network. It presents an efficient two-phase voting algorithm called TrustVoting to determine whether the data is faulty. However, the algorithms mentioned above provide neighbors' opinions being just normal and anomalous. In addition, taking the degree for neighbors' opinions about the data being normal or anomalous into account is able to improve detection performance [4].

Second category is to detect anomaly in the process of aggregating data in the network. Authors in [8] propose a robust aggregate framework, which performs the similarity tests among sensor nodes to classify the particular node as anomaly. It returns the aggregate results excluding anomaly, which is also maintained and sent to the users. Furthermore, authors in [9] define minimum support MinSupp, which is the minimum count of sensor nodes to prove the data of the node being normal or anomalous. For some node holds on anomalous data, if it has MinSupp number of nodes whose data are similar to it, it is determined that some events happen, otherwise it is faulty data. On this basis, [10] present the in-network anomaly detection framework based on position sensitive hash function. It achieves the load balance of the network. Using comparison pruning methods, it assures the detection performance and energy efficiency. Authors in [12] introduce PAO framework to reliably and efficiently detect anomaly in WSNs, which is able to operate over multiple window type, and operate in exact or approximate mode suiting for a variety of application requirements. However, the outputs of similarity test for all these frameworks mentioned above are also only yes or no, which depends on the prethreshold, and do not provide quantitative determination, which are similar to the voting algorithms.

The third one regard the sensing data of the nodes as time-series data to some extent. Authors in [13, 14] construct autoregressive (AR) models for sensor nodes. Every sensor node sends the coefficients of the models to sink after establishing AR models, and sink estimates approximate values of the sensor nodes in the following rounds without getting real data from sensor nodes. Thus, it reduces the number of messages sent in the network a lot. Once the data are no longer predicable from AR models, it maybe due to that the models are not suitable to the data or anomalous data arise. If the reason is the former, it needs reconstructing AR models and repeating the process mentioned above. Otherwise, the anomalous data are identified to be eliminated or corrected. Authors in [13] use two thresholds to distinguish them. However, the approach only relies on the predefined thresholds and does not employ the spatial correlations among sensor nodes. If taking spatial correlations into account, it can make full use of neighbors' opinions to achieve better performance of anomaly detection.

According to the above-related works, we can draw the conclusion that providing quantitative opinions is very important for anomaly detection after self-monitoring on each node in WSNs. As we know, in subjective logic theory, the subjects express subjective beliefs about the truth of the

objects with degree of uncertainty and indicate subjective belief ownership whenever required [15, 16]. Subjective logic provides the quantitative evaluation for the trust degree of the object. From this perspective, judgment among the adjacent nodes in WSNs is similar to trust evaluation. So we take subjective logic theory into the anomaly detection in WSNs. Subjective logic is involved to offer quantitative neighbors' opinions about the suspicious data of the node.

Besides, authors in [17–19] use machine learning techniques to detect anomaly in WSNs, which are different from our solution. For machine learning techniques are resource intensive that are difficult to be implemented on sensor nodes, the early studies, for example [17], run their algorithms on gateway (or sink). Authors in [17] identify anomalies in critical gas monitoring using offline echostate network in an underground coal mine. The following researches try to do something to make it possible to run the algorithms on sensor nodes. Authors in [18] compares and classifies the input signals in accordance with online learned prototypes on node-level, and then sends the results of classification to a fusion center for further processing. Based on [17], the authors in [19] propose a general anomaly detection framework which unifies fault and event detection. It runs on sensor nodes, distinguishes faults from events, and improves the performance of detection. The focuses of [18, 19] are how to select appropriate machine learning techniques and then decrease the complexity to make the algorithms be suitable to run on nodes. It is different from our solution, the difficulty of which is how to provide the quantitative neighbors' opinions to improve the performance of detection.

3. Preliminaries

Suppose that a sensor network is modeled as an undirected connected graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where \mathbb{V} is the set of all sensor nodes (including n sensor nodes v_1, \dots, v_n and one sink v_0 , denoted as $\mathbb{V} = V_n \cup v_0$) and \mathbb{E} is the set of the edges. An anomaly is defined as a measurement that significantly deviates from the normal pattern of the sensing data [3]. Generally, the anomaly mentioned in this paper includes fault (error) and event, and the anomalous data includes the faulty (erroneous) data and the informational data, respectively.

For the data of sensor nodes can be regarded as time series data [13, 14], we construct AR model on each node. Suppose that the data of node v_i at time t can be denoted by AR(p) as $x_{it} = \sum_{k=1}^p \varphi_k x_{i(t-k)} + \varepsilon$, where $x_{i(t-k)}$ is the data of v_i at time $t - k$ ($1 \leq k \leq p$), φ_k is the corresponding coefficient of $x_{i(t-k)}$, and ε is the random error and is the normal distribution of the mean being 0 and the variance being σ^2 . After that, given $\Phi = [\varphi_1 \cdots \varphi_p]'$ and $X_t = [x_{1t} \cdots x_{nt}]'$ we can get $\hat{\Phi}$ and \hat{X}_t . Among them, $\hat{\Phi}$ is the linear and the least variance-unbiased estimation of Φ , and \hat{X}_t is the unbiased estimation of X_t :

$$\begin{aligned} \hat{\Phi} &= [\hat{\varphi}_1 \cdots \hat{\varphi}_p]' = (Y'Y)^{-1}Y'Z, \\ \hat{X}_t &= \hat{\varphi}_1 X_{t-1} + \cdots + \hat{\varphi}_p X_{t-p}, \end{aligned} \quad (1)$$

where $Y = [X_j \cdots X_{j-p+1}]_{j=p \cdots M-1}$, $X_j = [x_{1j} \cdots x_{nj}]'$, $Z = [X_{p+1} \cdots X_M]'$. At last, given the confidence level $1-\alpha$, the confidence interval of the estimate value \hat{X}_t is

$$\left(\hat{X}_t \pm t_{\alpha/2}(M-2p) \cdot \hat{\sigma} \sqrt{1 + Y_0(Y'Y)^{-1}Y_0'} \right). \quad (2)$$

We make the following assumptions about our framework.

- (1) The wireless sensor network is static, and the topology does not change in the network lifetime.
- (2) All sensor nodes are homogeneous and have the same energy and capabilities, and there is only one sink which holds on infinite energy.
- (3) Sensor nodes are deployed densely; that is, if some events happen in the network, adjacent sensor nodes (one-hop neighbors) can monitor them at the same time. Of course, the situation can be extended to not densely deployed, which will be discussed in Section 6.

4. SLAD Framework

SLAD framework consists of three phases: preprocessing, self-monitoring, and cooperant detecting. Among them, preprocessing phase is executed on sink, self-monitoring run on each node, and cooperant detecting is semidistributed algorithm, that is, run on sink and sensor node.

In the first phase, all sensor nodes collect N rounds of data and transmit them to sink. Sink constructs autoregressive models AR(p) and uses the least squares to estimate the coefficients φ_k ($1 \leq k \leq p$). As for ε , it is estimated by use of the first M rounds of data. Using the least p rounds of data and the coefficients φ_k , we get the estimate value of the nodes. After that, using the last $N-M$ rounds data, we get the confidence interval $(\hat{X} \pm c_{it})$ under the given confidence level $1-\alpha$.

For each node v_i , if its data x_{it} at time t is within the range of its confidence interval $(\hat{x}_{it} \pm c_{it})$, it is considered as normal, otherwise anomalous. However, this computation run on each node, if it is computed at each round on each node, the computational complexity is so high as to consume too much energy, which significantly leads to increased energy consumption. Consequently, a simple approach is taken to approximate as shown below. Through the use of φ_k , each node predicts the latest $N-M$ rounds of data and compares them with the real data to get the average value of the confidence intervals of those $N-M$ rounds data, which is set as approximate confidence interval $(\hat{x}_{it} \pm \tau_i)$ at the given confidence level. Then it reduces the computational complexity on each node a lot. Sink sends the messages to each node including p coefficients of its AR model and its respectively approximate confidence interval.

In the second phase, each node uses p coefficients of its AR model and the most recently p rounds of data to predict current round of data. If the difference between the predicative data and the real data is less than the threshold τ , SLAD considers the data as normal. Otherwise, the data is

regarded as suspicious which needs to be determined further among adjacent neighbors. It is noted that, if the data is thought to be normal, it does not compute the confidence interval. However, while v considers x_t to be suspicious, it computes $(\hat{x}_t \pm c_t)$ at $1-\alpha$. And then, it sends the message to all its one-hop neighbors, which include x_t and $(\hat{x}_t \pm c_t)$.

In the third phase, sensor node whose data is suspicious sends its data to all its neighbors, and each neighbor produces its opinion about the suspicious data. SLAD fuses all the neighbors' opinions and gets the expectation of the consensus opinion. And thus we get the anomaly score of the suspicious data. If the anomaly score is more than the threshold, the suspicious data is anomalous, or else the data is normal. Additionally, to avoid the impact of those neighbors' opinions whose sensing data are suspicious, SLAD removes those opinions from the consensus opinion. In order to take the historical spatial correlations of the node and its neighbor nodes into account, SLAD computes the neighbors' opinions in another way. For the reason of different treatments to faulty data and informational data, SLAD adopt the approach as follows. The suspicious data, if anomalous, is to be marked as faulty data. When the faulty data of sensor nodes at this round are all sent to sink, sink distinguishes faulty data and informational data by employing the spatial correlations of adjacent nodes. The detailed process will be discussed further in Section 5. The third phase is the fundamental step of SLAD framework, which will be discussed in detail in Section 5.

5. Subjective Logic-Based Algorithms

In WSNs, no neighbor can always say that the data of the node are absolutely normal or anomalous, and something is neglected by previous works which we call uncertainty. On the other hand, subjective logic theory is suitable to model the situations with consideration to uncertainty. This drives us to involve subjective logic theory in anomaly detection to improve the detection performance.

Before detailing the subjective logic-based algorithms, it is necessary to address three problems, including expressiveness of neighbors' opinions, value assignment of neighbors' opinions, and consensus of neighbors' opinions. With the solutions of the problems, we propose SLB and ESLB which is the extension of SLB.

5.1. Expressiveness of Neighbors' Opinions

Definition 1. Given sensor network $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, $v, v_i \in \mathbb{V}$, $(v, v_i) \in \mathbb{E}$, the opinion of the neighbor v_i about the sensing data of node v is defined as follows:

$$\omega_v^{v_i} = (s_v^{v_i}, d_v^{v_i}, u_v^{v_i}, a_v^{v_i}), \quad s_v^{v_i} + d_v^{v_i} + u_v^{v_i} = 1, \quad (3)$$

where $s_v^{v_i}$ is the degree of belief that neighbor v_i considers the data of node v to be normal. $d_v^{v_i}$ is the degree of disbelief that v_i considers the data of node v to be anomalous. $u_v^{v_i}$ is the degree of uncertainty that v_i regards the data of node v as normal or anomalous. $a_v^{v_i}$ is the base rate of that v_i regards the data of node v as normal or anomalous (i.e., a priori probability).

Definition 1 defines neighbor v_i 's opinion about the degree of node v 's data. $s_v^{v_i}$, $d_v^{v_i}$ and $u_v^{v_i}$ are combined to express the opinion thoroughly. The following problem is how to determine the opinion $\omega_v^{v_i}$ of neighbor v_i about the data of node v .

5.2. Value Assignment of Neighbors' Opinions. In this section, we discuss how to determine neighbor's opinion $\omega_v^{v_i}$. We compute the similarity degree and difference degree of node v and v_i to denote as $s_v^{v_i}$ and $d_v^{v_i}$, respectively. It is worth mentioning that the sum of $s_v^{v_i}$ and $d_v^{v_i}$ maybe more than one by use of the above method. In the case, we should scale the sum down to no more than one because of the requirement of the subjective logic theory. $u_v^{v_i}$ is equal to subtract the sum of $s_v^{v_i}$ and $d_v^{v_i}$ from one.

To scale them down, we take advantage of the observation that the data of the nodes are changing smoothly most of the time and changing nonsmoothly every some periods for the reason the sampling rates of the nodes are high in WSNs. We have taken into account the data trends while constructing AR model. So we just use the data at the current round to determine neighbors' opinions while the data are changing smoothly. Only while the data are changing non-smoothly, we use several rounds of data to get the neighbors' opinions. As we know, data trends of the nodes can be get according to historical data.

The detailed opinion $\omega_v^{v_i}$ of neighbor v_i about the data of node v is determined as follows.

- (1) If the data are changing smoothly,

$$s_v^{v_i} = \begin{cases} \frac{x_i}{x}, & x_i \leq x \\ \frac{x}{x_i}, & x < x_i, \end{cases} \quad d_v^{v_i} = \frac{2 \cdot |x_i - x|}{(x_i + x)}, \quad (4)$$

where x_i and x are the data of node v_i and node v , respectively, at current round. If $s_v^{v_i} + d_v^{v_i} > 1$, the sum is scaled down to no more than one. $u_v^{v_i} = 1 - s_v^{v_i} - d_v^{v_i}$ is the prior probability of v_i 's opinion about v 's data, that is, the expectation of the prior opinion. Initially it is set to 0.5; that is, v_i considers the probability of the data of v being normal and anomalous is 0.5.

- (2) If the data are changing nonsmoothly,

$$s_v^{v_i} = \frac{X_i \cdot X}{\|X_i\|^2 + \|X\|^2 - X_i \cdot X}, \quad (5)$$

$$d_v^{v_i} = \sum_{j=1}^l \frac{2 \cdot |X_i(j) - X(j)|}{l \cdot (X_i(j) + X(j))},$$

where $X_i = [x_{i1} \cdots x_{il}]$, $X = [x_1 \cdots x_l]$, supposing the current round is l , X_i and X are the vector data of node v_i and v from 1 round to l rounds, $X_i(j)$ and $X(j)$ are the j th element of X_i and X , l is the length of vector data (X_i and X). If $s_v^{v_i} + d_v^{v_i} > 1$, the sum is scaled down to no more than one. $u_v^{v_i} = 1 - s_v^{v_i} - d_v^{v_i}$ is same as above.

5.3. *Consensus of Neighbors' Opinions.* The opinions of neighbors v_i and v_j about node v 's data can be fused to get the consensus which is the new opinion about the proposition on node v 's data being anomalous according to Lemma 2.

Lemma 2. *Given $v, v_i, v_j \in \mathbb{V}, (v, v_i) \in \mathbb{E}, (v, v_j) \in \mathbb{E}, \omega_v^{v_i} = (s_v^{v_i}, d_v^{v_i}, u_v^{v_i}, a_v^{v_i})$ and $\omega_v^{v_j} = (s_v^{v_j}, d_v^{v_j}, u_v^{v_j}, a_v^{v_j})$ are the opinions of neighbors v_i and v_j about the data of node $v, \omega_v^{v_i, v_j} = (s_v^{v_i, v_j}, d_v^{v_i, v_j}, u_v^{v_i, v_j}, a_v^{v_i, v_j})$ is the consensus of two neighbors' (v_i and v_j) opinions about the proposition on node v 's node being anomalous, it can be computed as follows. Let $k = u_v^{v_i} + u_v^{v_j} - u_v^{v_i} u_v^{v_j}$.*

If $k \neq 0$,

$$\begin{aligned} s_v^{v_i, v_j} &= \frac{d_v^{v_i} u_v^{v_j} + d_v^{v_j} u_v^{v_i}}{k}, \\ d_v^{v_i, v_j} &= \frac{s_v^{v_i} u_v^{v_j} + s_v^{v_j} u_v^{v_i}}{k}, \\ u_v^{v_i, v_j} &= \frac{u_v^{v_i} u_v^{v_j}}{k}, \\ a_v^{v_i, v_j} &= \frac{(1 - a_v^{v_i})(1 - u_v^{v_i})u_v^{v_j} + (1 - a_v^{v_j})(1 - u_v^{v_j})u_v^{v_i}}{k - u_v^{v_i} u_v^{v_j}}. \end{aligned} \quad (6)$$

If $k = 0$,

$$\begin{aligned} s_v^{v_i, v_j} &= \frac{d_v^{v_j} + d_v^{v_i} \gamma}{\gamma + 1} \\ d_v^{v_i, v_j} &= \frac{s_v^{v_j} + s_v^{v_i} \gamma}{\gamma + 1} \\ u_v^{v_i, v_j} &= 0 \\ a_v^{v_i, v_j} &= \frac{(1 - a_v^{v_j}) + \gamma(1 - a_v^{v_i})}{\gamma + 1}, \end{aligned} \quad \gamma = \lim \left(\frac{u_v^{v_j}}{u_v^{v_i}} \right) \quad (7)$$

Proof. From [15], we know that posteriori probabilities (ppdf) of binary events can be expressed as

$$\begin{aligned} f(p | r, t, a) &= \frac{\Gamma(r+t+2)}{\Gamma(r+2a)\Gamma(t+2(1-a))} p^{r+2a-1} \\ &\quad \times (1-p)^{t+2(1-a)-1}, \end{aligned} \quad (8)$$

where $0 \leq p \leq 1, r \geq 0, t \geq 0, 0 < a < 1$.

Here r, t , and a represent positive evidence, negative evidence, and relative atomicity (base rate), respectively. The probability expectation value is $E(f(p)) = (r+2a)/(r+t+2)$.

Let $f(p | r_v^{v_i}, t_v^{v_i}, a_v^{v_i})$ and $f(p | r_v^{v_j}, t_v^{v_j}, a_v^{v_j})$ be two ppdfs, respectively, held by the neighbor nodes v_i and v_j regarding

the truth of the suspicious sensing data of the node v . The ppdf $f(p | r_v^{v_i, v_j}, t_v^{v_i, v_j}, a_v^{v_i, v_j})$ defined as that [15]:

$$\begin{aligned} r_v^{v_i, v_j} &= r_v^{v_i} + r_v^{v_j}, \\ t_v^{v_i, v_j} &= t_v^{v_i} + t_v^{v_j}, \\ a_v^{v_i, v_j} &= \frac{a_v^{v_i}(r_v^{v_i} + t_v^{v_i}) + a_v^{v_j}(r_v^{v_j} + t_v^{v_j})}{r_v^{v_i} + t_v^{v_i} + r_v^{v_j} + t_v^{v_j}}. \end{aligned} \quad (9)$$

Let $\omega = (s, d, u, a)$ be a neighbor node's opinion about the suspicious sensing data, and let $f(p | r, t, a)$ be the same neighbor node's probability estimate regarding the same data. For $E(f(p)) = E(\omega)$, that is, $(r+2a)/(r+t+2) = s+au$, and $s+d+u=1$, it is easy to get $r=2s/u, t=2d/u$, where $u \neq 0$.

The following is the process to prove that the equations of the lemma are correct. Because we want to get the consensus about the proposition on node v 's data is anomalous, we get the equations with exchanging r and t of (9); respectively,

$$r_v^{v_i, v_j} = t_v^{v_i} + t_v^{v_j} = \frac{2d_v^{v_i}}{u_v^{v_i}} + \frac{2d_v^{v_j}}{u_v^{v_j}} = \frac{2d_v^{v_i} u_v^{v_j} + 2d_v^{v_j} u_v^{v_i}}{u_v^{v_i} u_v^{v_j}} = \frac{2d_v^{v_i, v_j}}{u_v^{v_i, v_j}}, \quad (10)$$

$$t_v^{v_i, v_j} = r_v^{v_i} + r_v^{v_j} = \frac{2s_v^{v_i}}{u_v^{v_i}} + \frac{2s_v^{v_j}}{u_v^{v_j}} = \frac{2s_v^{v_i} u_v^{v_j} + 2s_v^{v_j} u_v^{v_i}}{u_v^{v_i} u_v^{v_j}} = \frac{2s_v^{v_i, v_j}}{u_v^{v_i, v_j}}, \quad (11)$$

$$(10) \Rightarrow \frac{d_v^{v_i} u_v^{v_j} + d_v^{v_j} u_v^{v_i}}{u_v^{v_i} u_v^{v_j}} = \frac{d_v^{v_i, v_j}}{u_v^{v_i, v_j}}, \quad (12)$$

$$(11) \Rightarrow \frac{s_v^{v_i} u_v^{v_j} + s_v^{v_j} u_v^{v_i}}{u_v^{v_i} u_v^{v_j}} = \frac{s_v^{v_i, v_j}}{u_v^{v_i, v_j}}, \quad (13)$$

$$(12) + (13) \Rightarrow \frac{(s_v^{v_i} + d_v^{v_i})u_v^{v_j} + (s_v^{v_j} + d_v^{v_j})u_v^{v_i}}{u_v^{v_i} u_v^{v_j}} = \frac{s_v^{v_i, v_j} + d_v^{v_i, v_j}}{u_v^{v_i, v_j}}, \quad (14)$$

$$(14) \Rightarrow \frac{(1 - u_v^{v_i})u_v^{v_j} + (1 - u_v^{v_j})u_v^{v_i}}{u_v^{v_i} u_v^{v_j}} = \frac{1 - u_v^{v_i, v_j}}{u_v^{v_i, v_j}}, \quad (15)$$

$$(15) \Rightarrow 1 + \frac{u_v^{v_j} - u_v^{v_i} u_v^{v_j} + u_v^{v_i} - u_v^{v_i} u_v^{v_j}}{u_v^{v_i} u_v^{v_j}} = \frac{1}{u_v^{v_i, v_j}}. \quad (16)$$

Let $k = u_v^{v_i} + u_v^{v_j} - u_v^{v_i} u_v^{v_j}$.

If $k \neq 0$,

$$(16) \Rightarrow u_v^{v_i, v_j} = \frac{u_v^{v_i} u_v^{v_j}}{k}. \quad (17)$$

Combining (17) onto (12), we get

$$s_v^{v_i, v_j} = \frac{d_v^{v_i} u_v^{v_j} + d_v^{v_j} u_v^{v_i}}{u_v^{v_i} u_v^{v_j}} \times \frac{u_v^{v_i} u_v^{v_j}}{k} = \frac{d_v^{v_i} u_v^{v_j} + d_v^{v_j} u_v^{v_i}}{k}. \quad (18)$$

Combining (17) onto (13), we obtain

$$\begin{aligned} d_v^{v_i, v_j} &= \frac{s_v^{v_i} u_v^{v_j} + s_v^{v_j} u_v^{v_i}}{u_v^{v_i} u_v^{v_j}} \times \frac{u_v^{v_i} u_v^{v_j}}{k} \\ &= \frac{s_v^{v_i} u_v^{v_j} + s_v^{v_j} u_v^{v_i}}{k} \end{aligned} \quad (19)$$

$$\begin{aligned}
a_v^{v_i, v_j} &= \frac{(1 - a_v^{v_i})(r_v^{v_i} + s_v^{v_i}) + (1 - a_v^{v_j})(r_v^{v_j} + s_v^{v_j})}{r_v^{v_i} + s_v^{v_i} + r_v^{v_j} + s_v^{v_j}} \\
&= \frac{(1 - a_v^{v_i})(2(s_v^{v_i} + d_v^{v_i})/u_v^{v_i}) + (1 - a_v^{v_j})(2(s_v^{v_j} + d_v^{v_j})/u_v^{v_j})}{(s_v^{v_i} + d_v^{v_i})/u_v^{v_i} + (s_v^{v_j} + d_v^{v_j})/u_v^{v_j}} \\
&= \frac{(1 - a_v^{v_i})(1 - u_v^{v_i})u_v^{v_i} + (1 - a_v^{v_j})(1 - u_v^{v_j})u_v^{v_j}}{k - u_v^{v_i}u_v^{v_j}}. \tag{20}
\end{aligned}$$

If $k = 0$, let $\gamma = \lim(u_v^{v_i}/u_v^{v_j})$, it is easy to get the equation (7) which is similar as the above. \square

To be simply presented, we denote $\omega_v^{v_i, v_j} = (s_v^{v_i, v_j}, d_v^{v_i, v_j}, u_v^{v_i, v_j}, a_v^{v_i, v_j})$ as $\omega_v^{v_i, v_j} \equiv \omega_v^{v_i} \bar{\oplus} \omega_v^{v_j}$, among which $\bar{\oplus}$ is the new operator which is similar to the consensus operator of subjective logics. The expectation of consensus of neighbors' opinion about the data of node v decides the thorough consideration of neighbors about the data of v . Given consensus of neighbors' opinion $\omega_v^{v_i, v_j}$, the expectation of the opinion is $E(\omega_v^{v_i, v_j}) = s_v^{v_i, v_j} + a_v^{v_i, v_j} u_v^{v_i, v_j}$.

Example 3. Suppose that the opinions of neighbors v_i and v_j about the data of node v are $\omega_1 = (0.7, 0.2, 0.1, 0.5)$ and $\omega_2 = (0.8, 0.1, 0.1, 0.5)$ at some round, respectively, then the consensus of the opinions is $\omega_{1,2} = \omega_1 \bar{\oplus} \omega_2 = (s_{1,2}, d_{1,2}, u_{1,2}, a_{1,2}) = (0.158, 0.789, 0.053, 0.50)$, and the expectation is $E(\omega_{1,2}) = s_{1,2} + a_{1,2}u_{1,2} = 0.158 + 0.5 \times 0.053 = 0.184$.

As we all know, each node has many neighbors in WSNs. We need to fuse the opinions of all neighbors into the consensus opinion. Suppose that node v has m neighbors, their opinions about the data of v are $\omega_v^{v_1} = (s_v^{v_1}, d_v^{v_1}, u_v^{v_1}, a_v^{v_1}), \dots, \omega_v^{v_m} = (s_v^{v_m}, d_v^{v_m}, u_v^{v_m}, a_v^{v_m})$. To get the thorough consideration of neighbors about v 's data, we fuse all its neighbors' opinions, which denote as $\omega_v^{v_1, v_2, \dots, v_m} \equiv \omega_v^{v_1} \bar{\oplus} \omega_v^{v_2} \bar{\oplus} \dots \bar{\oplus} \omega_v^{v_m}$, that is, $\omega_v = (s_v, d_v, u_v, a_v)$. The consensus process is recursively called by use of Theorem 4.

Theorem 4. Given i neighbors v_1, v_2, \dots, v_i of node v , their opinions about the data of v are $\omega_v^{v_1} = (s_v^{v_1}, d_v^{v_1}, u_v^{v_1}, a_v^{v_1}), \dots, \omega_v^{v_i} = (s_v^{v_i}, d_v^{v_i}, u_v^{v_i}, a_v^{v_i})$, the consensus of their opinions about the proposition on node v 's node being anomalous is $\omega_v^{v_1, \dots, v_i}$, then it can be computed as follows:

$$\omega_v^{v_1, \dots, v_i} = \omega_v^{v_1, \dots, v_{i-1}} \bar{\oplus} \omega_v^{v_i} = \omega_v^{v_1} \bar{\oplus} \omega_v^{v_2, \dots, v_i} \quad (2 \leq i \leq m). \tag{21}$$

Proof. We utilize the mathematical induction approach to prove the theorem.

- (1) If $i = 2$, $\omega_v^{v_1, v_2} = \omega_v^{v_1} \bar{\oplus} \omega_v^{v_2}$, which illustrates that (21) is true.
- (2) Suppose that, if $i = k$, (21) is true; that is,

$$\omega_v^{v_1, \dots, v_k} = \omega_v^{v_1, \dots, v_{k-1}} \bar{\oplus} \omega_v^{v_k} = \omega_v^{v_1} \bar{\oplus} \omega_v^{v_2, \dots, v_k}, \tag{22}$$

we need to prove that (21) is true while $i = k + 1$; that is,

$$\omega_v^{v_1, \dots, v_{k+1}} = \omega_v^{v_1, \dots, v_k} \bar{\oplus} \omega_v^{v_{k+1}} = \omega_v^{v_1} \bar{\oplus} \omega_v^{v_2, \dots, v_{k+1}}. \tag{23}$$

It is equivalent to

$$\begin{aligned}
s_v^{v_1, \dots, v_{k+1}} &= s_v^{v_1, \dots, v_k} \bar{\oplus} s_v^{v_{k+1}} = s_v^{v_1} \bar{\oplus} s_v^{v_2, \dots, v_{k+1}}, \\
d_v^{v_1, \dots, v_{k+1}} &= d_v^{v_1, \dots, v_k} \bar{\oplus} d_v^{v_{k+1}} = d_v^{v_1} \bar{\oplus} d_v^{v_2, \dots, v_{k+1}}, \\
u_v^{v_1, \dots, v_{k+1}} &= u_v^{v_1, \dots, v_k} \bar{\oplus} u_v^{v_{k+1}} = u_v^{v_1} \bar{\oplus} u_v^{v_2, \dots, v_{k+1}}, \\
a_v^{v_1, \dots, v_{k+1}} &= a_v^{v_1, \dots, v_k} \bar{\oplus} a_v^{v_{k+1}} = a_v^{v_1} \bar{\oplus} a_v^{v_2, \dots, v_{k+1}}.
\end{aligned} \tag{24}$$

(i)

$$s_v^{v_1, \dots, v_k} \bar{\oplus} s_v^{v_{k+1}} = \frac{d_v^{v_1, \dots, v_k} u_v^{v_{k+1}} + d_v^{v_{k+1}} u_v^{v_1, \dots, v_k}}{u_v^{v_1, \dots, v_k} + u_v^{v_{k+1}} - u_v^{v_1, \dots, v_k} u_v^{v_{k+1}}} \tag{25}$$

For $\omega_v^{v_1, \dots, v_k} = \omega_v^{v_1, \dots, v_{k-1}} \bar{\oplus} \omega_v^{v_k} = \omega_v^{v_1} \bar{\oplus} \omega_v^{v_2, \dots, v_k}$, we can get the following:

$$(25) \Rightarrow \frac{d_v^{v_1} u_v^{v_2, \dots, v_k} u_v^{v_{k+1}} + u_v^{v_1} d_v^{v_2, \dots, v_k} u_v^{v_{k+1}} + u_v^{v_1} u_v^{v_2, \dots, v_k} d_v^{v_{k+1}}}{u_v^{v_1} u_v^{v_2, \dots, v_k} + u_v^{v_1} u_v^{v_{k+1}} + u_v^{v_2, \dots, v_k} u_v^{v_{k+1}} - 2u_v^{v_1} u_v^{v_2, \dots, v_k} u_v^{v_{k+1}}}, \tag{26}$$

(ii)

$$s_v^{v_1} \bar{\oplus} s_v^{v_2, \dots, v_{k+1}} = \frac{d_v^{v_1} u_v^{v_2, \dots, v_{k+1}} + d_v^{v_2, \dots, v_{k+1}} u_v^{v_1}}{u_v^{v_1} + u_v^{v_2, \dots, v_{k+1}} - u_v^{v_1} u_v^{v_2, \dots, v_{k+1}}}, \tag{27}$$

$$(27) \Rightarrow \frac{d_v^{v_1} u_v^{v_2, \dots, v_k} u_v^{v_{k+1}} + u_v^{v_1} d_v^{v_2, \dots, v_k} u_v^{v_{k+1}} + u_v^{v_1} u_v^{v_2, \dots, v_k} d_v^{v_{k+1}}}{u_v^{v_1} u_v^{v_2, \dots, v_k} + u_v^{v_1} u_v^{v_{k+1}} + u_v^{v_2, \dots, v_k} u_v^{v_{k+1}} - 2u_v^{v_1} u_v^{v_2, \dots, v_k} u_v^{v_{k+1}}}. \tag{28}$$

Equation (26) = (28); that is, $s_v^{v_1, \dots, v_{k+1}} = s_v^{v_1, \dots, v_k} \bar{\oplus} s_v^{v_{k+1}} = s_v^{v_1} \bar{\oplus} s_v^{v_2, \dots, v_{k+1}}$.

It is easy to know that the others (d , u , and a) can be proved as above. So (21) is true while $i = k + 1$.

The above procedure illustrates that (21) is true while i is no less than 2 and no more than m . That is, the theorem is proved to be true as follows:

$$\omega_v^{v_1, \dots, v_i} = \omega_v^{v_1, \dots, v_{i-1}} \bar{\oplus} \omega_v^{v_i} = \omega_v^{v_1} \bar{\oplus} \omega_v^{v_2, \dots, v_i} \quad (2 \leq i \leq m) \tag{29}$$

\square

Given m neighbors v_1, v_2, \dots, v_m of node v , their opinions about the data of v are $\omega_v^{v_1} = (s_v^{v_1}, d_v^{v_1}, u_v^{v_1}, a_v^{v_1}), \dots, \omega_v^{v_m} = (s_v^{v_m}, d_v^{v_m}, u_v^{v_m}, a_v^{v_m})$, the consensus of all the neighbors' opinions can be got through the computation of Theorem 4, then the expectation of consensus is $E(\omega_v) = s_v + a_v u_v$, where $s_v = s_v^{v_1, \dots, v_m}$, $u_v = u_v^{v_1, \dots, v_m}$, $a_v = a_v^{v_1, \dots, v_m}$. The anomaly score of the node v 's data is defined according to the expectation $E(\omega_v)$.

Definition 5. Suppose that the consensus of all the neighbors' opinions about node v 's data is ω_v and the expectation of the consensus is $E(\omega_v)$, then the anomaly score of node v is defined as follows:

$$AS_v = E(\omega_v). \tag{30}$$

TABLE 1: Notations used in the algorithms.

Notation	Description
m	Number of node v 's neighbors
V_{neighbor}	Node v 's neighbors set, $\{v_1, \dots, v_m\}$
x	Suspicious sensing data of node v
X	Suspicious vector data of node v
r	Current round
D_{neighbor}	Sensing data of V_{neighbor} at round r , $\{x_1, \dots, x_m\}$
VD_{neighbor}	Vector data of V_{neighbor} from $r - l + 1$ to r rounds, $\{X_1, \dots, X_m\}$
X'	Historical vector data of node v
VD'_{neighbor}	Historical vector data of V_{neighbor} , $\{X'_1, \dots, X'_m\}$
F_x	Indication of whether x is normal, faulty, or informational data, $F_x = 0$: x is normal; $F_x = 1$: x is faulty data; $F_x = 2$: x is informational data
AS_v	Anomaly score of node v
θ , thre	Predefine thresholds, discussed in Section 6
$\text{Corr}(x, x_i)$	Spatial correlation between x and x_i can be computed using extended Jaccard coefficient or correlation coefficient and so forth

There are some to be said. In the scenario that node v has one neighbor, Lemma 2 is not able to deal with it. To do with that, we suppose an imaginary neighbor who holds the opinion $\omega = (0, 0, 1, 0.5)$ and the neighbor takes part in the consensus with the real neighbor. Thus, we still get the consensus according to Lemma 2.

In the following sections, we present two algorithms to further determine whether suspicious data are normal or anomalous. The notations used to describe the algorithms are shown as in Table 1.

5.4. SLB Algorithm. The process of subjective logic-based algorithm (SLB) is as follows with discussion above. This process is executed among the node and its neighbors. Supposing node v has m neighbors v_1, v_2, \dots, v_m . According to the suspicious data of node v whether it is changing smoothly or nonsmoothly, each neighbor node v_i gives the opinion $\omega_v^{v_i}$ about the data of node v (Line 1–10). Utilizing Theorem 4 to compute, we get the consensus opinion ω_v of all the neighbors of node v (Line 11). The expectation of consensus opinion is obtained through the equation $E(\omega_v) = s_v + a_v u_v$ (Line 12). And then, the anomaly score AS_v can be get through Definition 5 (Line 13). If the anomaly score is less than the predefined threshold θ , the suspicious data of node v is considered as normal, or it is thought of as anomalous (Line 14–18) (Algorithm 1).

5.5. ESLB Algorithm. SLB algorithm fuses the opinions of all the neighbors about the data of the node to decide whether the data is normal or anomalous. However, it has the following disadvantages. (1) In the process of judgement among the node and its neighbors, the opinions of the neighbors whose data are suspicious are also included so as to affect the performance of anomaly detection. It is more severely affected especially when the proportion of anomalous data is ascending. (2) It does not distinguish the faulty data from the informational data. (3) The base rate

a of all the neighbors' opinions is set to 0.5 which is not reasonable. It does not take the historical information of the node and its neighbors into account.

To overcome the disadvantages of SLB, we extend SLB to ESLB. For the first point, ESLB removes the opinions of those neighbors whose data are suspicious. To solve the second point, ESLB employ the correlations of anomalous data. If those data are spatial correlated, they are the informational data or else the faulty data. Thirdly, we define a as follows in considering the historical information.

Suppose that X' and X'_i are the latest l rounds of historical data of node v and neighbor v_i in the pre-processing phase, the historical opinion of neighbor v_i about node v 's data is $\omega_v^{v_i'} = (s_v^{v_i'}, d_v^{v_i'}, u_v^{v_i'}, a_v^{v_i'})$. We set base rate $a_v^{v_i'}$ of historical opinion is 0.5; that is, $a_v^{v_i'} = 0.5$. Then we have the following definition.

Definition 6. Given the historical opinion of neighbor v_i about node v 's data is $\omega_v^{v_i'}$, base rate a of current opinion $\omega_v^{v_i}$ of v_i about v 's data is defined as follows:

$$a_v^{v_i} = E(\omega_v^{v_i'}). \quad (31)$$

Theorem 7. Suppose that historical opinion of neighbor v_i about node v 's data is $\omega_v^{v_i'}$, then base rate a of current opinion of v_i about v 's data is $a_v^{v_i} = s_v^{v_i'} + 0.5u_v^{v_i'}$.

Proof. From the definition of the expectation, we know that

$$E(\omega_v^{v_i'}) = s_v^{v_i'} + a_v^{v_i'} u_v^{v_i'}, \quad (32)$$

$$\left. \begin{array}{l} (31) \\ (33) \\ a_v^{v_i'} = 0.5 \end{array} \right\} \Rightarrow a_v^{v_i} = s_v^{v_i'} + 0.5u_v^{v_i'}. \quad (33)$$

□

We extend SLB to ESLB algorithm as follows. If the data x is suspicious, node v turns to its neighbors set V_{neighbor} to

```

Input:  $V_{\text{neighbor}}, x, X, D_{\text{neighbor}}, VD_{\text{neighbor}}$ ;
Output:  $F_x$ ;
(1) if  $r$  is at the time of data changing smoothly
(2) for  $1 \leq i \leq m$ 
(3) compute the opinion  $\omega_v^{v_i} = (s_v^{v_i}, d_v^{v_i}, u_v^{v_i}, a_v^{v_i})$  of neighbor  $v_i$  about  $v$  by use of (4)
(4) end for
(5) end if
(6) if  $r$  is at the time of data changing nonsmoothly
(7) for  $1 \leq i \leq m$ 
(8) compute the opinion  $\omega_v^{v_i} = (s_v^{v_i}, d_v^{v_i}, u_v^{v_i}, a_v^{v_i})$  of  $v_i$  about  $v$  by use of (5)
(9) end for
(10) end if
(11) get the consensus opinion  $\omega_v$  of all the neighbors  $v_1, v_2, \dots, v_m$  about node  $v$ 
(12) compute the expectation  $E(\omega_v)$  of the consensus opinion
(13) get the anomaly score  $AS_v$  of node  $v$ 
(14) if  $AS_v \leq \theta$ 
(15)  $x$  is normal data,  $F_x = 0$ ;
(16) else
(17)  $x$  is anomalous data,  $F_x = 1$  //here we do not distinguish faulty data from informational data
(18) end if
(19) return  $F_x$ ;

```

ALGORITHM 1: Subjective logic-based (SLB) algorithm.

further determine (Line 1–3). If the data of some neighbors are suspicious, they do not provide their opinions about the suspicious data of node v . We exclude the neighbors from the candidate neighbors set V_{neighbor} and get the neighbors set V_{neighbor} which provides the opinions about the data of node v (Line 4–8). For each node in V_{neighbor} , it computes its historical opinion $\omega_v^{v_k}$ of neighbor v_k about v 's data by use of X' and X'_k , and $a_v^{v_k}$ is set to 0.5 (Line 11). We compute the current opinion $\omega_v^{v_i}$ according to SLB algorithm excluding $a_v^{v_i}$ which is computed through Theorem 7 (Line 12). Then we get the result whether x is normal according to calling SLB algorithm (Line 15). If x is not normal, it sends message M_x to sink, which includes node v , current round l , data x , and flag F_x (Line 21). Sink receives all the messages at round r and further analyzes neighbors who hold on faulty data at this round. If x and x_i are all faulty at the same time and are spatial correlated, they are informational data or else faulty data (Line 24–32) (Algorithm 2).

6. Discussion

There are some problems to be explained further. First, authors in [8, 9] point out that voting algorithms cannot deal with the situation, in which the events are detected by sensor nodes which are not adjacent. However, our framework can do with the situation after minor revision. For example, suppose that node v_i and v_j are not within the radio range of each other and they detect the same event at some time. Suppose that the impact range of events is IR , radio range is CR , $h = \lceil IR/CR \rceil$. Our framework can still detect the event by computing the spatial correlation among h -hop neighbors. For the computation is executed on sink, it does not increase the energy consumption.

Second, in order to reduce the energy consumption, we use the idea proposed by [13] to construct and maintain

AR models. (1) It avoids unnecessary data transmission. While the data of nodes are normal, it does not transmit data in the network but estimates the data according to AR models by sink. (2) It reduces the computational complexity of constructing and maintaining AR models. The main computation is executed on sink and not sensor nodes. Please refer to [13] for more detail.

Third, although the thresholds, like θ and thre , are vital to SLAD, we do not pay much attention to them. We focus on how to more realistically quantize the opinion of the neighbors to special sensor node. In this paper, we set them with the historical experience. However, excellent methods are not excluded to improve SLAD further.

7. Simulation Results

7.1. Experimental Setup. We implement our simulation experiments in OMNET++ platform [20]. The topology and the sensing data come from Intel Berkeley research lab data set [21]. 54 sensors are deployed in the Lab of $400 * 700$, and the locations of sensor nodes are known in advance. In the experiments of Section 7.2, radio range is set to 150. Section 7.3 shows the impact of different radio ranges on the detection performance. All the experiments suppose that the radio links are reliable and do not fail. The sensing data have four attributes, yet only temperature is selected in our experiments. We use 1000 rounds of data as experimental data, and use the initial 100 rounds of data to construct AR models.

While using $AR(p)$ models to predicate the sensing data in WSNs, $AR(3)$ model can get good estimation and low cost of maintenance [13, 14]. So we use $AR(3)$ as the models constructed on the nodes. If p is set to 3, AR models can express as $X_t = \varphi X_{t-1} + \varphi X_{t-2} + \varphi X_{t-3} + \varepsilon$. In the beginning, we use the first $100(N)$ rounds as the training data, among

```

Input:  $V_{\text{neighbor}}, x, X, D_{\text{neighbor}}, VD_{\text{neighbor}}, X', VD'_{\text{neighbor}}$ ;
Output:  $F_x, F_{x_i}$ ;
(1) for each node  $v$ 
(2) if  $x$  is suspicious data
(3) node  $v$  turns to its neighbors  $V_{\text{neighbor}}$  to further determine
(4) for  $1 \leq i \leq m$ 
(5) if  $x_i$  is suspicious data
(6)  $v_i$  does not provide its opinion to node  $v$ ,  $V_{\text{neighbor}} = V_{\text{neighbor}} - \{v_i\}$ 
(7) end if
(8) end for
(9) for  $1 \leq k \leq m$ 
(10) if  $v_k \in V_{\text{neighbor}}$ 
(11) compute historical opinion  $\omega_{v_k}^{v_k'}$  of  $v_k$  about  $v$  by use of  $X$  and  $X'_k$ ,  $a_{v_k}^{v_k'} = 0.5$ 
(12) call SLB Algorithm (Line 1–10) to compute current opinion  $\omega_{v_k}^{v_k}$  excluding  $a_{v_k}^{v_k}$ , and  $a_{v_k}^{v_k} = s_{v_k}^{v_k'} + a_{v_k}^{v_k'} u_{v_k}^{v_k'}$ 
(13) end if
(14) end for
(15) call SLB Algorithm (Line 11–18) to get the result whether  $x$  is normal
(16) if  $x$  is normal
(17)  $F_x = 0$ 
(18) else
(19)  $F_x = 1$ 
(20) end if
(21) send message  $M_x$  to sink,  $M_x = \{v, r, x, F_x\}$ 
(22) end if
(23) end for
(24) sink receives all the messages at round  $r$ , and analyzes neighbors holding on faulty data at this round
//following executes on sink node
(25) if  $x$  and  $x_i$  are faulty at the same time
(26) if  $\text{Corr}(x, x_i) > \text{thre}$ 
(27)  $x$  and  $x_i$  are informational data,  $F_x = 2, F_{x_i} = 2$ 
(28) else
(29)  $x$  and  $x_i$  are faulty data,  $F_x = 1, F_{x_i} = 1$ 
(30) end if
(31) end if
(32) return  $F_x, F_{x_i}$ ;

```

ALGORITHM 2: Extended subjective logic-based (ESLB) algorithm.

which the first 90 (M) rounds of data are used to estimate the coefficients of AR model and the last 10 ($N-M$) rounds of data are used to determine the threshold τ .

If the sensing data are changing nonsmoothly, we would use the vector data to compute neighbors' opinions. To compute the base rate of neighbors to the node (historical information), it also needs to utilize the vector data. So, it needs to select the appropriate length of vector data (l). If l is set too small, it cannot express the data trends. Otherwise, it consumes too much energy to exchange sensing data. Figure 1 shows the detection rate of SLAD framework under the condition of different lengths of vector data. While l is not more than 5, the detection rate increases obviously with the increase of l . Once l achieves 5, the detection rate varies not obviously with the increase of l . Consequently, we set the length of vector data (l) to 5.

We randomly change some of normal data as faulty data and define the faulty rate as the proportion of faulty data to the whole data. In the experiments, we compare the performance of different algorithms at various faulty rate, and the results are mean of 20 times of executions.

7.2. Comparison of Detection Performance. In order to compare the anomaly detection performance of different algorithms, we define detection rate, false detection rate, and undetection rate. Among these definitions, the whole experimental data set is denoted as W_D , the real faulty data set is expressed as F_D , and the identified faulty data set which is determined by anomaly detection algorithms is marked as I_D .

Definition 8 (detection rate). It is defined as the faulty data which are determined as faulty in the proportion of the real faulty data:

$$\text{Detection_rate} = \frac{|F_D \cap I_D|}{|F_D|}. \quad (34)$$

Definition 9 (false detection rate). It is defined as those normal data which are determined as faulty in the proportion of the real faulty data:

$$\text{FalseDetection_rate} = \frac{|(W_D - F_D) \cap I_D|}{|F_D|}. \quad (35)$$

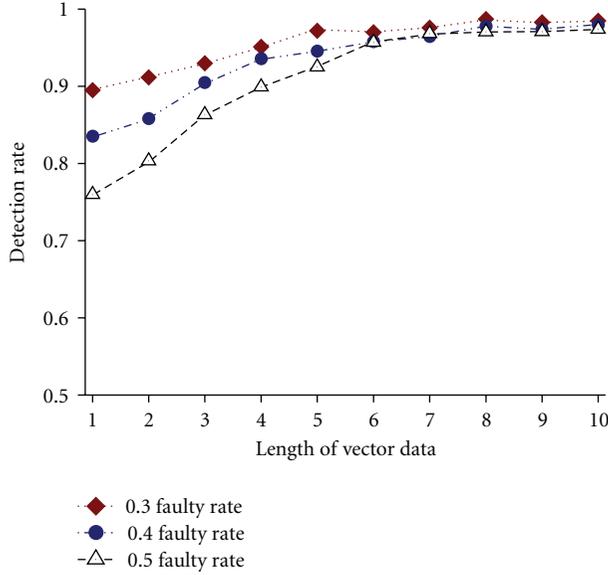


FIGURE 1: Impact of length on detection rate.

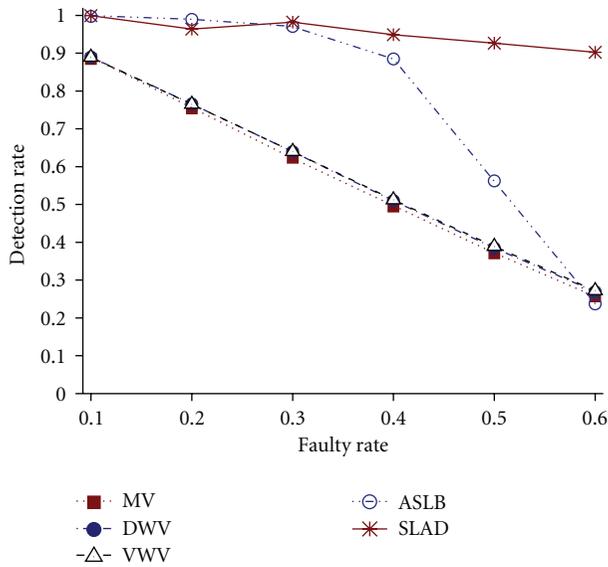


FIGURE 2: Detection rate of different algorithms.

Definition 10 (undetection rate). It is defined as those faulty data which are determined as normal in the proportion of the real faulty data:

$$\text{UnDetection_rate} = \frac{|F_D - (F_D \cap I_D)|}{|F_D|}. \quad (36)$$

In this section, we compare the performance of different algorithms. These algorithms are listed as follows. (1) MV (majority voting algorithm) [6]. (2) DWV (distance weight voting algorithm) [7]: it use the Euclidean distance of sensor nodes as the weight, and the weight is smaller with the distance being farther. Please refer to Section 2 about the details of MV and DWV algorithms. (3) VWV(value weight

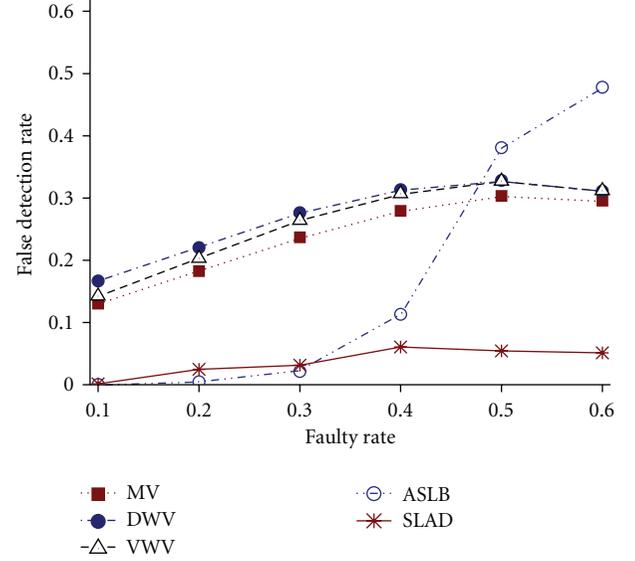


FIGURE 3: False detection rate of different algorithms.

voting algorithm): it is different from DWV, and it uses the distance of the data of node and its neighbors as the weight, that is, the difference of the data. It considers that the neighbors whose data are closer to that of the node should have greater weights. (4) ASLB (autoregressive model and SLB): it combines autoregressive models with subjective logic-based algorithm. (5) SLAD (subjective logic-based anomaly detection framework): it integrates autoregressive model and extended subjective logic-based algorithm (ESLB algorithm).

Figure 2 shows the detection rate of five algorithms at different faulty rate. It indicates that detection rates of all the algorithms are greater than 0.8 when faulty rate is low. The performances of ASLB and SLAD are better than MV, DWV, and VWV. The detection rates of MV, DWV, and VWV decrease sharply as faulty rate increases. ASLB keeps the high detection rate when faulty rate is less than 0.4, which decreases sharply once faulty rate reaches 0.4 and holds this trend with the increase of faulty rate. However, the detection rate of SLAD is still greater than 0.9 even though faulty rate increases, which shows the best performance compared with the other algorithms.

Figure 3 presents the detailed comparison results of these algorithms at different faulty rate. The false detection rate of all the algorithms increases as faulty rate becomes larger. The false detection rate of MV, DWV, and VWV keeps in some specified scope as faulty rate increases, and ASLB increases suddenly once faulty rate achieves 0.4. SLAD holds the false detection rate within limits which is no greater than 0.1. The false detection rate of SLAD is much less than the others.

We then study the impact of different faulty rate on undetection rate of these algorithms. The undetection rate of MV, DWV, and VWV decreases as faulty rate increases. The undetection rate of ASLB increases abruptly while faulty rate achieves 0.4, and it keeps the rising trend with the increase of faulty rate. SLAD preserves very low undetection rate which

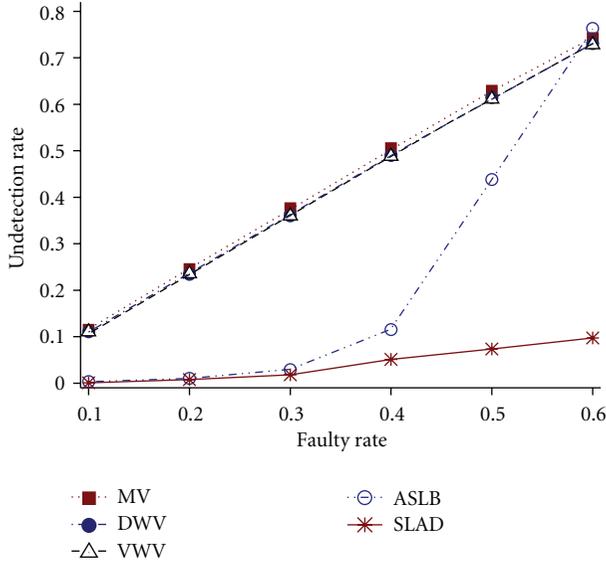


FIGURE 4: Undetection rate of different algorithms.

does not exceed 0.1 even though faulty rate is high. The undetection rate of SLAD is much less than other algorithms though it increases as faulty rate increases.

From the above figures, we note that ASLB suddenly changes its trends of detection performance when faulty rate is 0.4. The reason is presented as follows. When faulty rate is 0.4, the number of neighbors whose sensing data are right is more than those data being anomalous on average. It results that the detection performance does not decline too much. However, once faulty rate is more than 0.4, the number of neighbors whose data are faulty is no less than that whose data are normal. It is hard to decide whether the suspicious data is normal for ASLB, and it results to the poor detection performance significantly.

We also draw the following conclusion according to Figures 2, 3, and 4. The overall performance of SLAD is much better than the other algorithms, and the performance of ASLB is better than MV, DWV, and VWV when faulty rate is low. The cause is the combination of subjective logic. Using subjective logic, ASLB and SLAD fuses the quantitative opinions of neighbors which avoid the problems other algorithms are facing. Because MV, DVW, VWV, and ASLB use the opinions of all the neighbors, the number of faulty data of neighbors may be rising along with faulty rate increasing, which shows the bad impact on the detection rate, false detection rate, and undetection rate. However, SLAD has removed the opinions of the neighbors whose data are suspicious before providing their opinions and takes historical spatial correlations of the nodes and their neighbors into account. So, SLAD holds significantly superior performance than other algorithms, especially when faulty rate is high.

The above experiments discuss the cases that the network are only involving the faulty data, and not including the informational data. In the monitoring area, some events randomly arise. The anomalous data of sensor nodes detecting the events are spatial correlations (i.e., informational

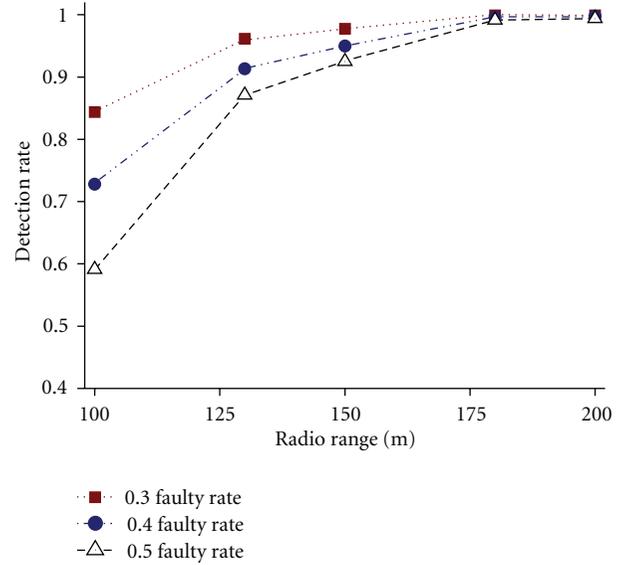


FIGURE 5: Impact of radio range on detection rate.

data). The faulty rate, which is defined as the number of informational data in proportion of the whole data, is set to 0.2. The experiment shows that detection rate of SLAD framework for informational data reaches more than 0.9, and MV, DWV, VWV are only about 0.7. The reason is that SLAD framework utilizes subjective logic to fuse the quantitative opinions of neighbors so as to improve the detection performance obviously.

7.3. Impact of Radio Range on Detection Performance. In this section, we analyze the impact of radio range on detection rate, false detection rate, and undetection rate at different faulty rate. The number of neighbors affects the detection performance of the algorithm. Different radio range of the nodes leads to different number of neighbors. Thereby, we discuss the detection performance of SLAD framework under the condition of different radio ranges.

We conduct the experiments to compare the detection performance of SLAD framework under different radio ranges. We set faulty rate to 0.3, 0.4, and 0.5 in the experiments. Figures 5, 6, and 7 show the detection rate, false detection rate, and undetection rate of SLAD, respectively. These figures indicate that detection rate decreases; false detection rate and undetection rate increase with the increase of faulty rate. They also show that detection rate increases; false detection rate and undetection rate decrease as the radio range becomes larger. The reason is that there are more neighbors providing the opinions with the radio range increasing.

8. Conclusions

In this paper, we present SLAD framework which considers the uncertainty of neighbors to the data of the node. It includes three phases: pre-processing, self-monitoring, and cooperant detecting. In the first phase, sink constructs AR

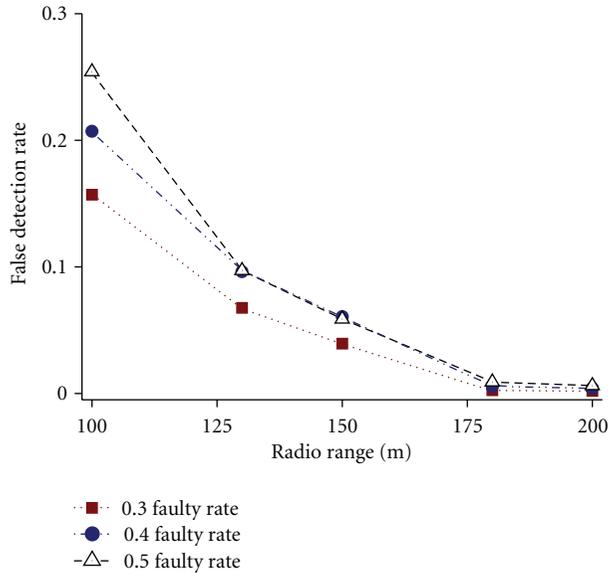


FIGURE 6: Impact of radio range on false detection rate.

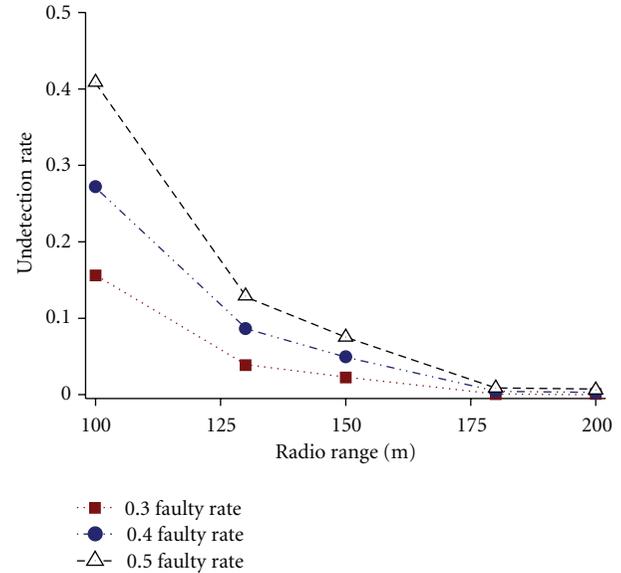


FIGURE 7: Impact of radio range on undetection rate.

model for each node. In the second phase, it uses AR models to check whether the sensing data are suspicious. In the third phase, it presents two novel algorithms SLB and ESLB. The third phase is the key of our framework. In SLB, each neighbor gives the quantitative opinion to the suspicious data involving with subjective logic theory. After fusing the opinions of all the neighbors, SLB gets the expectation of the consensus opinion and anomaly score, which demonstrates the degree of the suspicious data being considered as an anomaly. We extend SLB to ESLB in order to avoid the impact of those neighbors whose data are suspicious, effectively distinguish the faulty data from the informational data, and take the historical spatial correlations of the node and its neighbors into account. Simulation results show that SLAD framework improves the performance of anomaly detection effectively compared with previous works.

However, we find there is something to do for further improving SLAD. We believe that the opinion of the neighbor, who holds the higher historical spatial correlation with the node, should be paid more attention to. An example is given to demonstrate that. Suppose node *A* and node *B* are the neighbors of node *C* and node *A* and node *C* are located in the room while node *B* is out of the room. Generally, the historical spatial correlation between node *A* and node *C* is higher than that between node *B* and node *C*. Thus, the opinion of node *A* to node *C* should be given more attention. Unfortunately, the subjective logic, which works as the foundation of SLAD, treats the opinions equally and has no capability to deal with it. As the preparatory work, we proposed an operator for subjective logic which is capable of making the consensus on several neighbors' opinions with their weights in a fair way [22]. With the support of the new operator, we can map the historical spatial correlation to the weight of the opinion to improve SLAD. In theory, we believe it will improve the performance of anomaly detection for SLAD. It is our future work.

Acknowledgments

This work is supported by the National Science Foundation (61070056, 61033010), the National 863 High-tech Plan (2008AA01Z120), Program for New Century Excellent Talents in University, and the Research Funds of the Renmin University of China (10XNI018).

References

- [1] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: mobile networking for "smart dust"," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 271–278, Seattle, Wash, USA, August 1999.
- [2] D. Cruller, D. Estrin, and M. Sivastava, "Overview of sensor networks," *Computer*, vol. 37, pp. 41–49, 2004.
- [3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.
- [4] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 159–170, 2010.
- [5] S. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom, "Declarative support for sensor data cleaning," in *Proceedings of the 4th International Conference on Pervasive Computing*, pp. 83–100, Dublin, Ireland, May 2006.
- [6] B. Krishnamachari and S. Iyengar, "Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 241–250, 2004.
- [7] M. Krasniewski, P. Varadharajan, B. Rabeler, S. Bagchi, and Y. C. Hu, "TIBFIT: trust index based fault tolerance for arbitrary data faults in sensor networks," in *Proceedings of the International Conference on Dependable Systems and Networks*, pp. 672–681, Yokohama, Japan, July 2005.
- [8] Y. Kotidis, A. Deligiannakis, and V. Stoumpos, "Robust management of outliers in sensor network aggregate queries," in

- Proceedings of 6th International ACM Workshop on Data Engineering for Wireless and Mobile Access*, pp. 17–24, Beijing, China, June 2007.
- [9] A. Deligiannakis, Y. Kotidis, V. Vassalos, V. Stoumpos, and A. Delis, “Another outlier bites the dust: computing meaningful aggregates in sensor networks,” in *Proceedings of the 25th IEEE International Conference on Data Engineering (ICDE '09)*, pp. 988–999, Shanghai, China, April 2009.
- [10] N. Giatrakos, Y. Kotidis, A. Deligiannakis, V. Vassalos, and Y. Theodoridis, “TACO: tunable approximate computation of outliers in wireless sensor networks,” in *Proceedings of the International Conference on Management of Data (SIGMOD '10)*, pp. 279–290, Indianapolis, Ind, USA, June 2010.
- [11] X. Y. Xiao, W. C. Peng, C. C. Hung, and W. C. Lee, “Using sensor ranks for in-network detection of faulty readings in wireless sensor networks,” in *Proceedings of the 6th International ACM Workshop on Data Engineering for Wireless and Mobile Access*, pp. 1–8, Beijing, China, June 2007.
- [12] N. Giatrakos, Y. Kotidis, and A. Deligiannakis, “PAO: power-efficient attribution of outliers in wireless sensor networks,” in *Proceedings of the 7th International Workshop on Data Management for Sensor Networks*, pp. 33–38, Singapore, September 2010.
- [13] D. Tulone and S. Madden, “PAQ: time series forecasting for approximate query answering in sensor networks,” in *Proceedings of the European Conference on Wireless Sensor Networks*, pp. 21–37, Zurich, Switzerland, February 2006.
- [14] D. Tulone, “A resource—efficient time estimation for wireless sensor networks,” in *Proceedings of the Joint Workshop on Foundations of Mobile Computing (DIALM-POMC '04)*, pp. 52–59, Philadelphia, Pa, USA, October 2004.
- [15] A. Jøsang, “A logic for uncertain probabilities,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 3, pp. 279–311, 2001.
- [16] A. Jøsang, “Fission of opinions in subjective logic,” in *Proceedings of the 12th International Conference on Information Fusion*, pp. 1911–1918, Seattle, Wash, USA, July 2009.
- [17] O. Obst, X. R. Wang, and M. Prokopenko, “Using echo state networks for anomaly detection in underground coal mines,” in *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '08)*, pp. 219–229, St. Louis, Mo, USA, April 2008.
- [18] M. Wälchli, “Efficient signal processing and anomaly detection in wireless sensor networks,” in *Proceedings of the EvoWorkshops on Applications of Evolutionary Computing: Evo-COMNET, EvoENVIRONMENT, EvoFIN, EvoGAMES, EvoHOT, EvoIASP, EvoINTERACTION, EvoOmUSART, EvoNUM, EvoSTOC, EvoTRANSLOG*, pp. 81–86, Tübingen, Germany, April 2009.
- [19] M. Chang, A. Terzis, and P. Bonnet, “Mote-based online anomaly detection using echo state networks,” in *Proceedings of the 5th IEEE International Conference on Distributed Computing in Sensor Systems*, pp. 72–86, Marina Del Rey, Calif, USA, June 2009.
- [20] A. Varga, “The OMNET++ discrete event simulation system,” in *Proceedings of the European Simulation Multiconference*, pp. 319–324, Prague, Czech, June 2001.
- [21] Intel Berkeley Research Lab, <http://berkeley.intel-research.net/labdata/>.
- [22] H. Zhou, W. Shi, Z. Liang, and B. Liang, “Using new fusion operations to improve trust expressiveness of subjective logic,” *Wuhan University Journal of Natural Sciences*, vol. 16, no. 5, pp. 376–382, 2011.

Research Article

A Self-Organized and Smart-Adaptive Clustering and Routing Approach for Wireless Sensor Networks

Kyuhong Lee and Heesang Lee

Department of Industrial Engineering, Sungkyunkwan University, Suwon 440-746, Republic of Korea

Correspondence should be addressed to Heesang Lee, leehee@skku.edu

Received 15 July 2011; Revised 1 October 2011; Accepted 6 October 2011

Academic Editor: Yuhang Yang

Copyright © 2012 K. Lee and H. Lee. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Efficient energy consumption is a critical factor for the deployment and operation of wireless sensor networks (WSNs). In general, WSNs perform clustering and routing using localized neighbor information only. Therefore, some studies have used self-organized systems and smart mechanisms as research methods. In this paper, we propose a self-organized and smart-adaptive clustering (SOSAC) and routing method, which performs clustering in WSNs, operates the formed clusters in a smart-adaptive way, and performs cluster-based routing. SOSAC is comprised of three mechanisms, which are used to change the fitness value over time, to back up routing information in preparation for any potential breakdown in WSNs, and to adapt to the changes of the number of sensor nodes for a WSN. We compared the performance of the proposed SOSAC with that of a well-known clustering and routing protocol for WSNs. Our computational experiments demonstrate that the network lifetime, energy consumption, and scalability of SOSAC are better than those of the compared method.

1. Introduction

A wireless sensor network (WSN) is an infrastructure composed of wireless sensor nodes, which perform sensing tasks and transmit the data to a base station (BS) that is the final processing node for the WSN. According to the features of the wireless network, WSNs are widely used in dynamic and hazardous regions as well as in many industries including production, logistics, distribution, transportation, and health [1].

WSNs have many characteristics that differ from conventional wireless networks. For example, WSNs have several inherent constraints that are unique to this type of wireless networks [2, 3]. First, a WSN consists of hundreds or thousands of wireless sensor nodes but each node in a WSN is constrained in terms of processing capability and storage capacity. Second, the power device of the sensor node in WSNs cannot be recharged or replaced so that energy efficiency is very important [4].

Routing protocols in a WSN should be developed and proposed with consideration for these unique characteristics. The routing decides the transmission route for a data packet from the sensor node to the final destination BS.

The communication channels of WSNs can be configured through multihop mesh network called “flat routing”. In contrast, a “cluster-based routing” method divides a WSN into several clusters. In this cluster-based routing, each cluster is comprised of a “cluster head” (CH) and several “cluster members” (CMs). After formation of the cluster, hierarchical routing is performed, where CMs transmit data packets to its CH, and the CH integrates these data packets and transmits them to the BS directly or via other CHs [2].

Along with the development of communication networks, the routing paradigm has changed from a centralized system to a distributed system due to the demand for better scalability and simple installation. In WSNs, tens or thousands of sensor nodes may be needed to create one network, so when selecting the processor, memory, and power devices, economical devices with limited functions are used [1]. Accordingly, the distributed routing method is used rather than the centralized routing method for WSNs considering the processing ability of the sensor nodes [1]. When distributed processing is used, each sensor node is not allowed to use the information of the entire WSNs and must therefore perform clustering and routing using the localized neighbor information only. When the power device

for sensor nodes is not rechargeable or replaceable, it is also important to consider energy efficiency in order to maximize the network lifetime of the sensor nodes by minimizing the energy consumption [4].

Self-organization, which is a further development of distributed processing, is a system that uses local information only through a distributed and peer-to-peer method. Many studies on self-organized systems are being conducted using only local information and simple rules because of the associated advantages such as reduction of overheads in communication traffic, scalability, and robustness. Also, self-organization has been applied to many areas in the wireless network field such as ad-hoc networks, WSNs, wireless LAN, in routing, forming clusters, MAC protocol design, and radio resources management. Some studies have been conducted on self-organization with the features of WSNs [5].

Recently “smart” characteristics of routing have also been suggested in telecommunication networking [6–8]. Here smart means the capability to describe and analyze a situation, taking decisions based on the available data in a predictive or adaptive manner, and thereby performing smart actions.

In this paper, we propose a self-organized and smart-adaptive clustering (SOSAC) and routing method comprised of three mechanisms for a WSN. First, in order to select an appropriate CH, which is the most important factor influencing the clustering performance, two types of performance measure are used. The proposed mechanism adjusts the weights of these two performance measures automatically to reflect the changes in the WSN. Second, the broadcasting range, which is another important factor influencing the performance of clustering in SOSAC, is predetermined as a function of the number of sensor nodes. Third, to overcome problems caused by possible breakdown, damage, or failure of sensor nodes in WSNs, a smart backup mechanism is established to monitor the state of the CH without heavy overhead and to restore the system automatically in the case of such problems.

This paper is organized as follows. Section 2 reviews previous studies on clustering, self-organization, and smart telecommunication networks and their implications for this research. Section 3 presents the radio model that is used and the basic assumptions used in this study. Section 4 explains the process of forming self-organized clusters and routing. Section 5 describes how to operate clustering in a smart-adaptive way by using three proposed mechanisms. In this section, we also compare the performance of the proposed SOSAC with that of another well-known model. The measures for comparison are network lifetime, residual energy after a certain time, and scalability of the model. The last section presents the conclusions of this study and proposes future research directions.

2. Related Works

Grouping sensor nodes into clusters has been widely pursued by the research community in order to achieve the network scalability objective. The objective of clustering is mainly to generate stable clusters in environments with sensor nodes.

In addition to supporting network scalability, clustering has numerous advantages. It can localize the route set up within the cluster and thus reduce the size of the routing table stored at the individual sensor node [3, 9]. Clustering can also conserve communication bandwidth since it limits the scope of intercluster interactions to CHs and avoids redundant exchange of messages among sensor nodes [10]. Moreover, clustering can stabilize the network topology at the level of sensor nodes, and thus cut down on topology maintenance overhead. Sensor nodes are only affected by the connection with their CHs and not by changes at the level of inter-CH tier [11]. The CH can also implement optimized management strategies to further enhance the network operation and prolong the battery life of the individual sensor nodes and the network lifetime [10].

“Self-organization” is defined as the process where a structure or pattern appears in a system without intervention by external directing influences. It organizes through direct interaction in a peer to peer method [5]. The advantages of using the self-organized system are as follows [5, 12]. First, one of the most important characteristics of self-organization is the completely distributed control. Each participating system component acts on local decisions, that is, it is not possible to review the current global state and act accordingly. Second, an inherent feature of self-organizing systems is their capability to adapt to changing environmental conditions. This is a direct result of the distributed peer to peer working principle. Third, the robustness of the system prevents any problems due to breakdown, damage, or failure of individual elements. Fourth, scalability protects the system from degradation by increasing the number of individual elements in the system.

Several cluster-based and self-organized protocols for WSNs were studied as follows.

In the LEACH protocol [13], the basic idea is to select sensor nodes randomly as CHs. Random selection of a CH is good for self-organization of the cluster configuration. Cluster configuration is repeated at each round, and the round is divided into two phases: the “set-up phase” and the “steady phase”. In the set-up phase, LEACH selects a CH candidate with a threshold for a random number. In this phase, each sensor node compares a random number with the threshold $T(n)$ to elect itself to a CH, and this process is performed independently for each cluster. In the steady phase, the sensor nodes sense environment and transmit data, and the CHs aggregate the data before sending the data to the BS.

LEACH-Energy Distance (LEACH-ED) [14] is another self-organized protocol that is based on LEACH. It uses a different threshold from LEACH. The ratio between the residual energy of a sensor node and the total current energy of all of the sensor nodes in the network is used for the first threshold of LEACH-ED. It also uses the distance threshold as the second threshold. If the distance between a sensor node and an existing CH is less than the distance threshold, the sensor node cannot be elected as a CH.

The hybrid-energy efficient distributed protocol (HEED) [15] selects CHs by combining the residual energy of the sensor and the communication cost for selection of the CH

in a WSN. The communication cost is calculated using the average minimum reachability power or the neighbor sensor node degree as the secondary parameter. Once the CH is selected, HEED uses a hierarchical routing protocol with a 3-tier structure, where first each CM transmits data to its CH, second each CH sends data to one special CH by using the breadth-first search tree, and finally the special CH sends all data to the BS.

Robust energy efficient distributed clustering (REED) [16] is a self-organized clustering method which constructs k independent sets of CH overlays on the top of the physical network to achieve fault tolerance. Each sensor must reach at least one CH from each overlay. The method of selection of CH is same as HEED.

Distributed, energy-efficient, and dual-homed clustering (DED) [17] is a self-organized clustering method that achieves fault tolerance by providing an alternative route from sources to the BS. Each regular sensor node has a primary CH and a secondary backup CH. DEED and HEED are the same method for CH selection but use different parameters for the selection.

In telecommunications, although some researchers have studied smart systems for the formation or operation of communication networks, each study has used different methods and different definitions, as shown in the related studies described below.

In traditional networks, passive means were used to send packets or signals to each termination, but active networks have recently been suggested that allow nodes to customize computation on message flowing [18] or a new smart network framework with not only active services but also the concepts of contextawareness and userawareness [19]. In this network, the nodes of the network have the capability to sense, to reason and to be aware of the context and behavior of users, and automatically provide active services to users according to their situation and context knowledge [20].

Stone et al. [21] have suggested a smart network which requires three types of intelligence. First, it is an inference system for collecting information about the user's interaction with the network. Second, it is a framework associated with mobile agents for routing through the network of processing messages and servers. Third, it is a gateway to allow access to the network.

Gelenbe et al. [6, 7] have suggested a cognitive packet network which provides intelligent capabilities for routing and flow control to packets instead of the nodes or protocol in the wired network or wireless ad hoc network. This has enabled the realization of a smart routing system that allows the cognitive packets to find their own route between source and destination in the packet switched networks.

3. Network Modeling

3.1. First Order Ratio Model. A sensor node consumes energy when transmitting and receiving data packets in a WSN. In wireless data transmission, energy consumption is correlated to the data packet size and the distance between the two sensor nodes. Extensive research has been conducted in the area of low-energy radios. Different assumptions about

the radio characteristics, including energy dissipation in the transmission and received modes, will change the advantages of different protocols. In our work, we assume the following first-order radio model as our radio energy consumption model.

- (i) Transmitting the data packet: a sensor node consumes $\epsilon_{elec} = 50 \text{ nJ/bit}$ at the transmitter circuitry and $\epsilon_{amp} = (100 \text{ pJ/bit})/m^2$ at the amplifier.
- (ii) Receiving the data packet: a sensor node consumes $\epsilon_{elec} = 50 \text{ nJ/bit}$ at the receiver circuitry.
- (iii) A k -bit data packet is transmitted from sensor node to sensor node, and d_{ij} is the distance between the two sensor nodes i and j ; the energy consumption of sensor node i is given by $T_{ij} = \epsilon_{elec} \times k + \epsilon_{amp} \times d_{ij}^2 \times k$.
- (iv) The sensor node receives data packet: the energy consumption of sensor node is given by $R_i = \epsilon_{elec} \times k$.

3.2. Assumptions and Definitions for the Model. We assign the following properties and make the following assumptions for modeling and simulation for the WSN.

- (i) The location of all sensor nodes and the BS are fixed.
- (ii) The location of the BS (25 m, 150 m) is known in advance in the $50 \text{ m} \times 50 \text{ m}$ sensor field.
- (iii) The data packet size is 1,000 bits, and signal packet size is 50 bits.
- (iv) All sensor nodes have an initial energy of 0.5 J.
- (v) Period: in the data transmission phase, a CM creates information sensed by itself into a data packet and transmits this packet to its CH. The CH aggregates the data packets received from its CMs and transmits them to the BS through the intercluster route. A cycle of this process is defined as a "period".
- (vi) Round: some number of periods carried out within a data transmission phase can be defined as a "round". In our experiments, we assume that 1 round is made up of 10 periods.
- (vii) Network lifetime: the periods until a certain number of sensor nodes drained of its energy can be defined as the "network lifetime".

4. Self-Organized and Smart-Adaptive Clustering (SOSAC): The Proposed Clustering and Routing Protocol

This section explains the procedure of the SOSAC proposed herein. SOSAC decides the CHs every round, and it is comprised of three phases, as set forth in Sections 4.2, 4.3, and 4.4. In Section 4.1, the state of each sensor node of the WSN is explained to help understand the SOSAC operations.

4.1. States of Sensor Nodes. Each sensor node of SOSAC performs its duties while being changed into the following five states depending on the roles. Figure 1 indicates the state transition diagram of SOSAC.

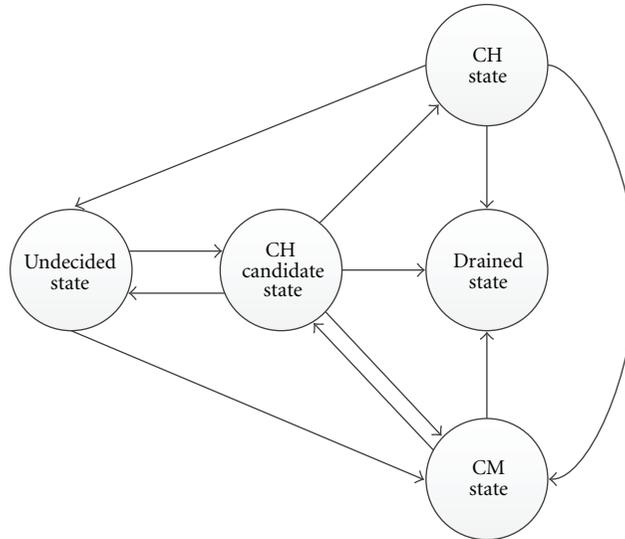


FIGURE 1: Transition diagram of sensor node states.

- (i) *Undecided state*: when a sensor node has been scattered first in the sensor field, after a CH has completed its duty, and after a sensor node has dropped out of the node with CH candidate state, the sensor node becomes a node with undecided state. A node with undecided state does not belong to any cluster, yet.
- (ii) *CH candidate state*: when the sensor nodes are scattered in the sensor field at period zero, all sensor nodes become a node with CH candidate state. Some of sensor nodes with a CM state also can become nodes with CH candidate state. Only a node with CH candidate state can compete for selection of CHs.
- (iii) *CH state*: one node with CH candidate state within a cluster becomes a node with CH state, that is, a CH in that period. As a CH, it collects and aggregates information from its CMs and transfers data packets to other CHs or the BS. Also, a CH decides its nearby nodes as the nodes with CH candidate state for the next round.
- (iv) *CM state*: when a CH is decided, all other sensor nodes in the same cluster become nodes with CM state, that is, CMs. A CM periodically sends the sensed information to its CH.
- (v) *Drained state*: when a sensor node cannot function anymore because all its energy has been drained or it has broken down, it becomes a node with drained state.

SOCAC is comprised of a clustering phase that forms the clusters, an intercluster routing phase that decides on the transmission route among the CHs, and a data transmission phase that sends/receives the data packets.

Figure 2 indicates the progress of SOSAC in a flowchart.

4.2. Clustering Phase. SOSAC needs a clustering phase for the hierarchical routing. The clustering phase commences if the sensor nodes are scattered first in the sensor field or after the “data transmission phase” has finished. The clustering phase is comprised of three steps: Broadcasting step, CH selection step, and clustering step as shown in Algorithm 1.

4.2.1. Broadcasting Step. When the sensor nodes are scattered in the sensor field at period zero, no sensor node belongs to any cluster, thus all sensor nodes are assumed in the same cluster at period zero. The CHs broadcast a CH-change-signal packet only within the broadcasting range in procedure 2.1 of Algorithm 1 because since a CH was at a good position when it was selected as the CH, it is highly likely that its neighbors are also at good location for being CHs. By limiting the broadcasting range, the overhead of the clustering phase can be reduced because of the proximity of the nodes with the CH candidate state.

4.2.2. CH Selection Step. In procedure 3.2 of Algorithm 1, we update the counter, which is the number of received CH-candidate-signal packets. This counter will be used for calculating the fitness value in procedure 5 of Algorithm 1. A CH-candidate-information-signal packet, which a node with CH candidate state broadcasts in procedure 4.1 of Algorithm 1, contains the energy state of the sending sensor node. Information of the received CH-candidate-signal packets is also used to calculate the fitness value in procedure 5 of Algorithm 1. The CH-candidate-information-signal packets are broadcast within $2 \times$ (broadcasting range) for the following reasons. All nodes with the CH candidate state of a cluster must share the neighborhood information with all nodes with CH candidate states within the same cluster. This can be achieved by sending the CH-candidate-information-signal packets within two times of the broadcasting range for

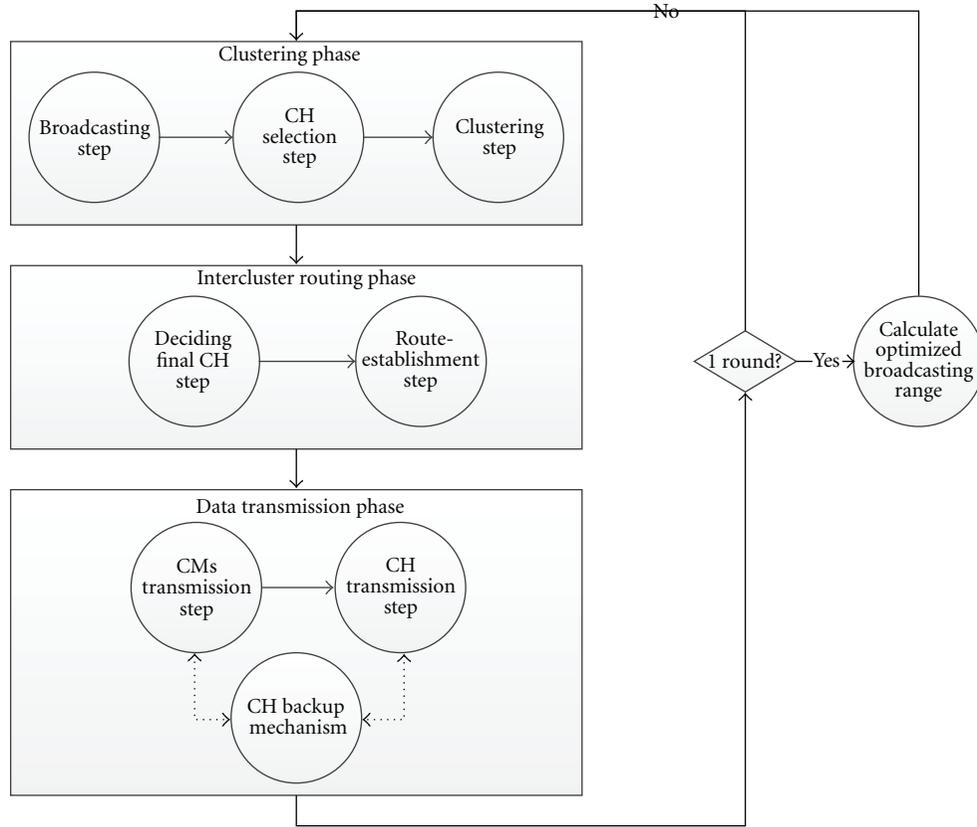


FIGURE 2: Flowchart of SOSAC.

two nodes with the CH candidate state that are farthest away from each other.

4.2.3. Clustering Step. A node with CH state in procedure 8.1 of Algorithm 1 broadcasts a CH-signal packet to the entire sensor field to form its cluster. The sensor nodes in all states except nodes with CH state or nodes with drained state that receive these signal packets select the nearest CH and become a CM of that cluster.

4.3. Intercluster Routing Phase. After the clustering phase, the CHs create an intercluster routing tree and select a CH as the “master CH”. Each CH sends packets to the master CH and the master CH transmits the packet to the BS directly. Transmission of data packets through the intercluster routing tree, and its master CH can save energy compared to the direct transmission of data packets from all CHs to the BS.

In the process of creating a tree for intercluster routing, the CHs perform their duties while changing into the following three states depending on the roles.

- (i) *Initial state*: if the intercluster routing phase starts, all CHs are initialized as the CHs with initial state. If a CH with initial state receives a route-broadcast-signal packet, it transmits ACK back to the CH that sent the route-broadcast-signal packet to the CH.
- (ii) *Route broadcasting state*: after finding the master CH, that CH becomes a CH with route broadcasting state.

The master CH sends route-broadcast-signal packets to CHs with initial state. A CH with initial state that received a route-broadcast-signal packet becomes a CH with route broadcasting state.

- (iii) *Route-established state*: the CH with route broadcasting state broadcasts route-broadcast-signal packet to establish an intercluster route and then becomes a CH with route-established state. When all CHs with initial state become CHs with route-established state, all CHs become CHs with route-established state.

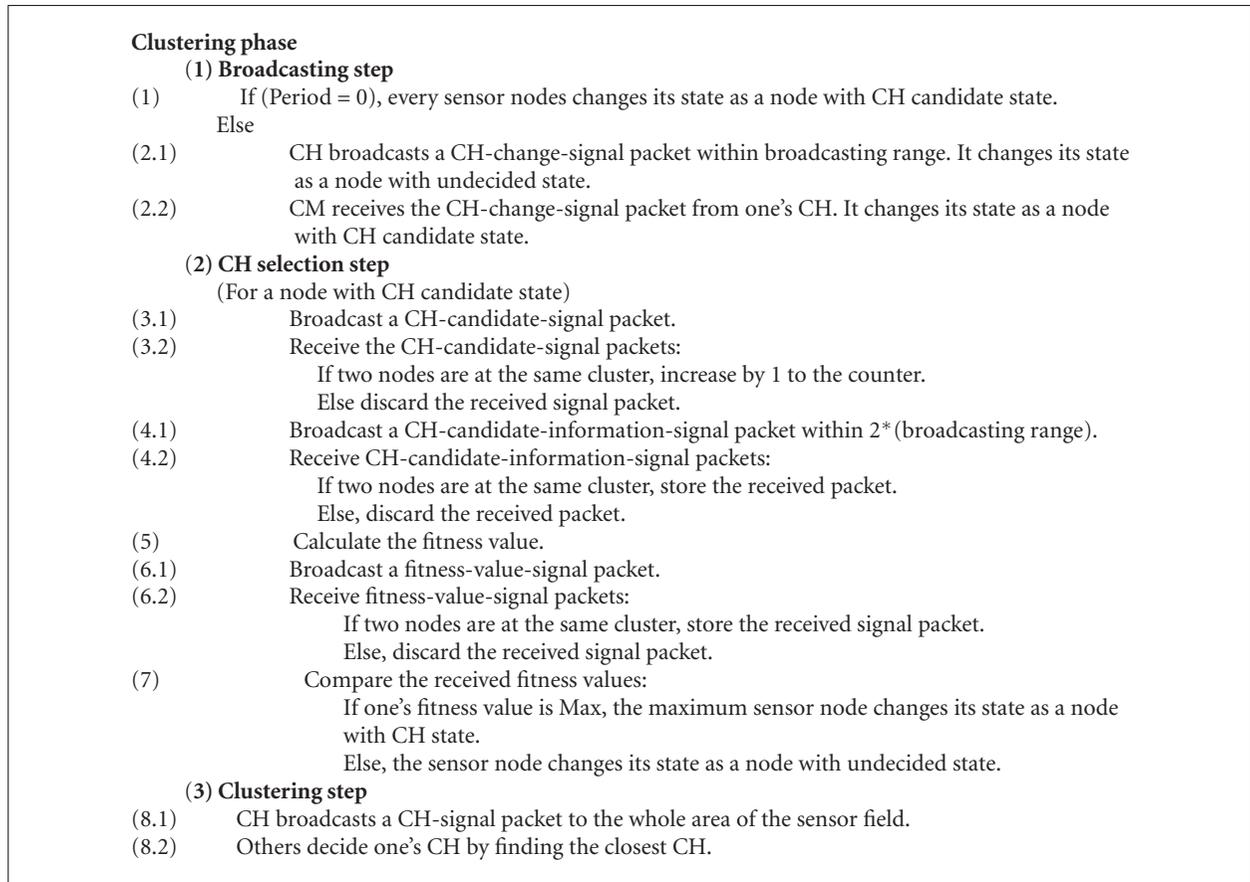
Figure 3 indicates the transition diagram of CHs in the intercluster routing phase.

The intercluster routing phase is divided into the deciding master CH step and the route establishment step.

4.3.1. Deciding Master CH Step. The expected residual energy in procedure 1 of Algorithm 2 is the expected remaining energy of each CH, which is calculated as the following.

$$\text{The expected residual energy} = \text{current energy} - \gamma, \quad (1)$$

where γ is the expected consuming energy which sends data packets to BS during the next round. CH broadcasts a CH-expected-residual-energy-signal packet to the whole area of the sensor field in procedure 2.1 of Algorithm 2. We use this simple broadcasting technique because any CH



ALGORITHM 1: The clustering phase algorithm.

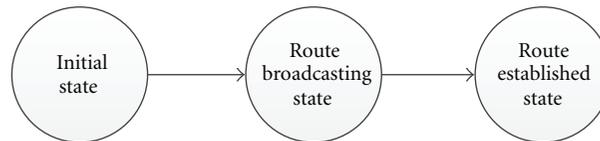


FIGURE 3: Transition diagram of CH states in the intercluster routing phase.

can get energy information for all CHs in the sensor field and compare it with its own energy information since each CH broadcasts CH-expected-residual-energy-signal packets to the whole area of the sensor field. This does not generate a heavy overhead since the signal packets are small and we assume that the sensor field is small (between 50 m*50 m and 150 m*150 m) in this study. When the sensor field is very large, we need to modify this intercluster routing procedure by using some flat routing techniques.

4.3.2. Route Establishment Step. A CH with initial state that receives the route-broadcast-signal packets sets the nearest CH with route established state as the first section of its transmission route. The CHs with route broadcasting state broadcast route-broadcast-signal packets in procedure 4 of Algorithm 2. The initial broadcasting range is δ m. (In our

computational implementation, we use $\delta = 25$.) If a CH with initial state receives a route-broadcast-signal packet, it transmits ACK to the CH that sent a route-broadcast-signal packet. If a CH with route broadcasting state does not receive any ACKs during given time interval, it extends the broadcasting range by δ m repeatedly until the broadcasting range is wider than (width of the sensor field * $\sqrt{2}$). Within this broadcasting range, if a CH with route broadcasting state does not receive any ACK, it becomes a leaf CH of the intercluster routing tree. This route establishment method yields a spanning tree that includes all CHs. This spanning tree is built as a breadth-first search tree while the CHs exchange the route-broadcast-signal packets and ACKs. While the breadth-first search tree is made, a CH tries to include its neighbor CHs by increasing the broadcasting range δ when the CH does not receive any ACKs, which is a kind of stop-and-wait method.

Intercluster routing phase**(1) Deciding master CH step**

(For a CH with initial state)

- (1) Calculate expected residual energy.
- (2.1) Broadcast a CH-expected-residual-energy-signal packet to the whole area of the sensor field.
- (2.2) Receive CH-expected-residual-energy-signal packets.
- (3) Compare CH expected residual energy:
If one's CH expected residual energy is Max, the maximum CH changes its state as a CH with route broadcasting state.

(2) Route establishment step**While** (All CHs with initial state become CHs with route established state)

(For a CH with route broadcasting state)

- (4) Broadcast a route-broadcast-signal packet within the intercluster broadcasting range. If it receives ACK, it changes its state as a CH with route established state.
If it does not receive any ACK during given time interval, it extends the broadcasting range. If it does not receive any ACK within maximum broadcasting range, that CH becomes a leaf CH.

(For a CH with initial state)

If it receives route-broadcast-signal packets:

- (5) Find the closest CH that sent route-broadcast-signal packets and establish intercluster route by sending ACK to the closest CH. It changes its state as a CH with route broadcasting state.

End while

ALGORITHM 2: The intercluster routing phase algorithm.

4.4. *Data Transmission Phase.* Once the clusters and the intercluster routing tree have been created, information sensed by each sensor node is transmitted to the BS using data packets in the data transmission phase. The data transmission phase is divided into 2 steps: CM transmission step and CH transmission step.

In Algorithm 3, a CM transmits data packets, which are created by sensing the surroundings in each period, to its CH. Each CH aggregates the received data packets and transmits the aggregated data packet to the BS through the route that is set in the intercluster routing phase. Sensing and data transmission of each sensor node are repeated during one round.

5. Smart Mechanisms and Computational Experiments of SOSAC

5.1. Smart Mechanisms in SOSAC

5.1.1. *Smart Fitness Value Adaptability Mechanism.* The most important factor affecting the clustering performance is the CH selection method. SOSAC uses the following fitness comparison procedure to select the appropriate CHs.

First, check the location of each node with CH candidate state for the fitness value. SOSAC uses the number of the received CH-candidate-signal packets to check the appropriateness of the location of a sensor node. In other words, each node with CH candidate state collects information on the locations of some neighboring nodes with CH candidate state (in the same cluster) as the number of received CH-candidate-signal packets and transmits it to the neighboring sensor nodes to compare the number of CH-candidate-signal packets. A CH with many neighboring nodes with

CH candidate state is considered as a good candidate for the CH of a cluster since it is located in the center of the cluster. Accordingly, SOSAC decides on the neighborhood degree fitness of each node with CH candidate state based on the ratio of the number of neighbors of ν to the maximum number of neighbors of the cluster where the sensor node belongs to. This fitness value indicates the location appropriateness of the node with CH candidate in the current cluster. We define "neighborhood degree fitness" of a sensor node ν that has CH candidate state of the cluster as follows. The "neighborhood degree fitness" consists of the first component of the fitness value.

Neighborhood degree fitness of ν

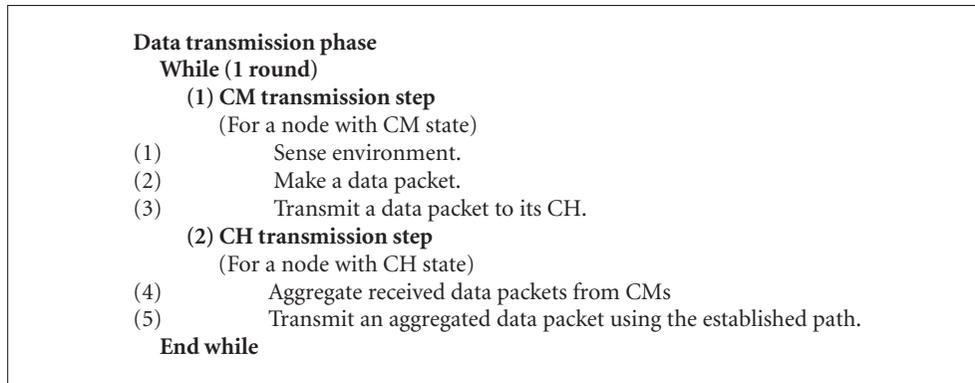
$$= \left(1 + \frac{\text{The number of neighbor of } \nu}{\text{Max. number of neighbor of cluster}(\nu)} \right)^\alpha \quad (2)$$

where $\text{cluster}(\nu)$ is the cluster that sensor node ν belongs to, and α is the average ratio of remaining energy to initial energy of $\text{cluster}(\nu)$.

Second, "energy state fitness" of a node with CH candidate state should be considered as the second component of the fitness value. Since CH consumes more energy than the CMs, SOSAC gives a priority to a node with CH candidate state that has more residual energy. Therefore, the energy state is used as the second component of the fitness value, and we define energy state fitness as follows:

Energy state fitness of ν

$$= \left(1 + \frac{\text{residual energy of } \nu}{\text{Max. residual energy of cluster}(\nu)} \right)^{1-\alpha} \quad (3)$$



ALGORITHM 3: The data transmission phase algorithm.

Since each component of the two fitness components has values between 1 and 2, the values of the two fitness components are inherently normalized. Also we use α and $1-\alpha$ as exponent parameters, each of which decides the weight of the respective fitness component. A total fitness value can be calculated by adding these two components together as follows.

$$\begin{aligned} \text{Total fitness value} = & \text{neighborhood degree fitness of } v \\ & + \text{energy state fitness of } v. \end{aligned} \quad (4)$$

If the total fitness value is to reflect the state of the networks, then it should be adjusted in a smart way to consider the change in the environment. In the beginning of networking, each sensor node has sufficient residual energy so that the energy state fitness is not very important. Therefore, the neighborhood degree fitness should be weighted more. This can be done by using the suggested total fitness value since the neighborhood degree fitness becomes more important than the energy state fitness as α remains relatively large. As time goes by, however, if the sensor nodes consume more energy, the scarcity of the energy state should be reflected. This can be done by using the suggested total fitness value since the energy state fitness becomes more important than the neighborhood degree fitness as $(1-\alpha)$ becomes large. Hence, SOSAC tries to achieve energy balance by increasing the weight of the energy state fitness.

For SOSAC implementation, we constructed a source code using Visual C++ of Visual Studio 2008 and conducted a computer simulation. The sensor field for the test was $50\text{ m} \times 50\text{ m}$ in size, and the simulation was carried out under the assumption that 100 sensor nodes with an initial energy of 0.5 J were distributed. The test used the average of the performances of the 10 different sensor distributions in the $50\text{ m} \times 50\text{ m}$ sensor field.

Figure 4 is an experiment to examine the effects of self-adjusting the weight α of the fitness value in a smart way. We compared the original SOSAC that self-adjusts the weights of the fitness value as time passes with three variations of SOSAC that are given with the fixed weights (0:1, 1:1, 1:0) of the two fitness value components. Here 0:1 means that

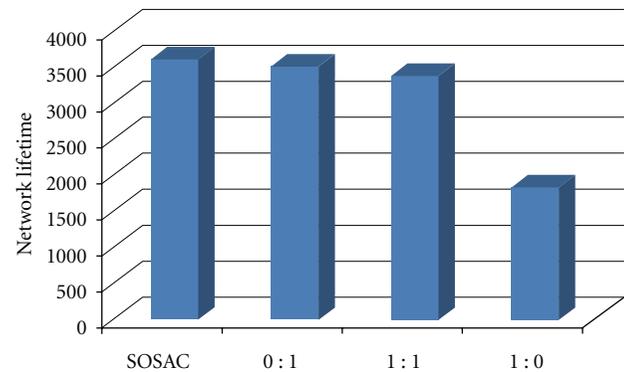


FIGURE 4: Effect of the fitness value with self-adjustment of its weights.

SOSAC uses energy state fitness only, 1:1 that equal weights of two fitness value components are used, and 1:0 that SOSAC uses neighborhood degree fitness only.

In Figure 4, the Y axis indicates the network lifetime and the X axis indicates the ratio of weights. The network lifetime of the automatically adjusted SOSAC is longer than those of the fixed weight (0:1, 1:1, and 1:0) SOSACs. This experiment shows that SOSAC adjusts suitable weights by itself in a smart way rather than providing weight parameters as inputs.

5.1.2. Smart Adjustment of Broadcasting Range. Along with the selection of an appropriate CH in the hierarchical routing, deciding on the number and size of the clusters is one of the important factors that determines the energy consumption of the sensor nodes. In this section, we suggest a method for deciding the appropriate number and size of clusters.

The number of clusters in SOSAC is decided by the broadcasting range of the sensor nodes. As the broadcasting range widens, the number of sensor nodes with CH candidate states, which participate in the election of a CH, increases. Consequently, a large cluster is formed with fewer clusters and CHs. With the fewer CHs, the energy consumption of the CHs for transmitting data packets over

a long distance is also reduced. However, because there are more CMs, the CH has to receive more data packets from them. Therefore, the energy consumption is greater when receiving the data packets in a large cluster. On the contrary, if the broadcasting range narrows, the number of clusters and CHs is increased due to the formation of small clusters. With an increasing number of CHs, the energy consumption for receiving is reduced, whereas the energy consumption of the CHs for transmission increases. Therefore, the broadcasting range of SOSAC must be optimized to increase the network lifetime of the WSNs.

Since we can enumerate many values for the broadcasting range of a sensor field in a computer simulation, it is not difficult to optimize the broadcasting range during computer simulation. However, it is impossible for the sensor nodes, which have poor calculation capability, to determine the optimized broadcasting range in real time in the sensor field. Hence we try to find the optimized broadcasting range as the function of the number of sensors that varies from 50 to 250 in the 50 m*50 m sensor field using computer simulations for SOSAC. Hence we make the following predictive formula that calculates the optimized broadcasting range. The formula (5) was made from the following nonlinear regression model that has the adjusted coefficient of determination = 0.977:

$$y = \frac{0.000034x^2 - 0.02783x + 9.677}{\text{an area of sensor field}/2500 (50 \text{ m} * 50 \text{ m})}, \quad (5)$$

where y is the broadcasting range(m) and x the number of sensors in the sensor field.

If the sensor nodes sense the number of sensor nodes, an optimal broadcasting range can be adjusted in a smart way using the above formula, and the number of sensor nodes can be computed after the data packets are transmitted in the first round. In other words, respective CHs can determine the number of their CMs using the number of data packets transmitted by their CMs in the first period. This information is transmitted to the master CH through the intercluster routing, and the master CH can determine the number of sensor nodes scattered in the sensor field by summing them. Therefore, in the clustering phase of the first round of SOSAC, clusters are formed in the broadcasting range with the initial value to perform routing. When the first round is over, the master CH calculates an optimized broadcasting range and broadcasts this information to the entire sensor field so that all the sensor nodes can be adjusted in the optimized broadcasting range at the start of the second round.

Figure 5 compares the optimized broadcasting range calculated by an enumerative method without the smart adjustment mechanism with the network lifetime calculated with the smart adjustment mechanism. In Figure 5, the Y axis indicates the network lifetime when 20% of the sensor nodes have become drained in periods, and the X-axis shows five different experimental sizes. Figure 5 shows that the network lifetimes when SOSAC uses an optimized broadcasting range by finding an enumerative method are longer than those when SOSAC uses the smart adjustment mechanism of formula (5). In fact, the smart adjustment of

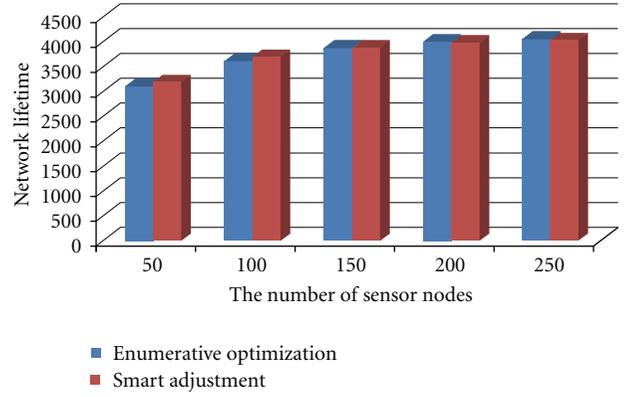


FIGURE 5: Effect of optimized broadcasting range.

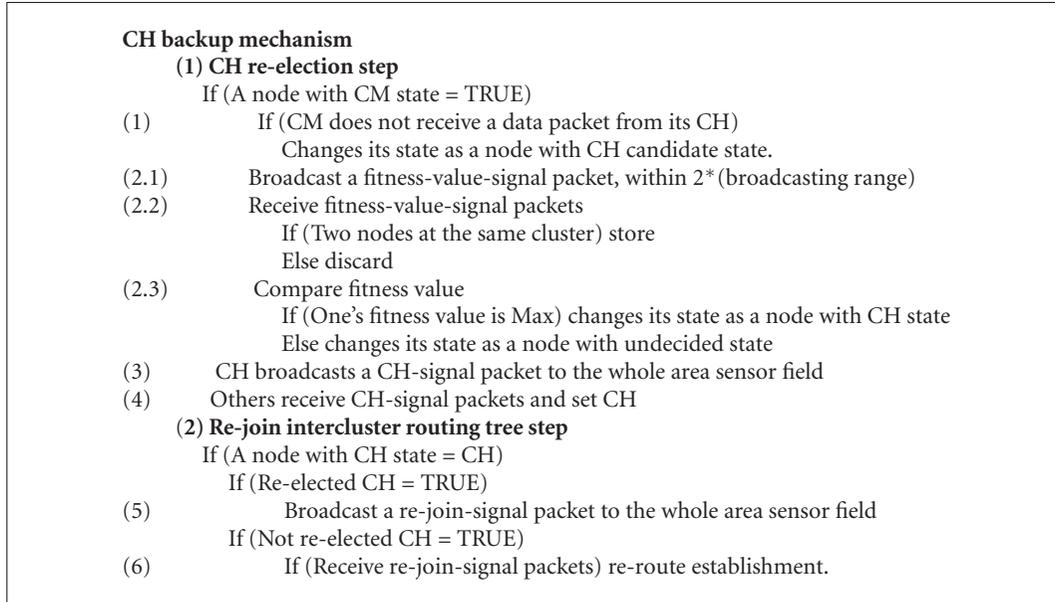
the broadcasting range of formula (5) shows almost identical performance with the enumerative optimum values of the broadcasting range, which cannot be easily implemented in real WSNs. This demonstrates that the smart adjustment mechanism of predictive formula (5), which can be easily implemented in real WSNs, can provided good estimation of the optimal range of broadcasting range.

5.1.3. Smart Backup Mechanism. A WSN can be used in inaccessible and dangerous areas such as battlefields and hazardous regions. Also some of the WSN sensor nodes can be broken down by vicious physical attack or cyber-attack before their energy is used up. Therefore, the robustness of WSNs is important for dealing with potential problems such as breakdown, failure, or attack of the sensor nodes.

When cluster-based routing is used, all information on the clusters may be lost if the sensor nodes in the node with CH state have become drained or failed due to breakdown, failure, or attack. Transmission of the entire network may fail depending on the location of the CH on the tree in the intercluster routing.

In order to avoid these risks, SOSAC has a “smart backup mechanism” that allows all the CMs of each cluster to recognize their CH’s states. If a CH losses its function unexpectedly, SOSAC has to select a new CH to minimize the transmission failures in the network. SOSAC can sense all CMs within its cluster range because it has to transmit the data packets that are broadcast by the CH to the CH of the neighboring cluster during intercluster routing. No additional overheads for sensing CHs or CMs are needed because the sensor nodes with CH state can be recognized depending on whether the data packets have been received or not. The smart backup mechanism of SOSAC selects a new CH when the CH has lost its function, and SOSAC needs overheads only to reconnect the network without using failed CHs. The smart backup mechanism is comprised of the following 2 steps: CH reelection step and rejoin intercluster routing tree step.

CH Reelection Step. The CH reelection step is carried out immediately when the CMs recognize a failure of the CH.



ALGORITHM 4: The CH backup mechanism algorithm.

In procedure 2.1 of Algorithm 4, the CMs broadcast the fitness values calculated in the previous CH selection step of clustering phase within 2^* (broadcasting range) to restore the system as soon as possible when a CH failure occurs. Each CM compares the received fitness values with its own fitness value to select a new CH. The selected CH broadcasts the CH-signal packets (procedure 3). When the CMs receive the CH-signal packets, they select this CH as their new CH (procedure 4).

Rejoin Intercluster Routing Tree Step. All the CMs of the cluster assume that they can obtain information on the neighboring CHs on the intercluster routing tree of their own CH during the clustering step in the clustering phase. The new CH elected in procedures 5 and 6 of Algorithm 4 broadcasts a rejoin-signal packet to its neighboring CHs on the tree to restore the network to a stable state again. The smart backup mechanism of SOSAC is able to restore an unstable network to a stable state by using unused information without incurring additional overhead to the recognition of nodes with CH state.

The smart backup mechanism should be able to restore the network, not only if an error has occurred in the CH, but also if several sensor nodes have lost their functions in a certain concentrated area or sporadic failures of each sensor node in scattered areas. In order to check if the smart backup function works properly, we can conduct two types of experiments. The basic setting for the experiments is the same as the experiment for smart adaptability of the fitness value.

The first experiment observes how long the network is able to maintain itself in the event that all sensor nodes have lost their function to sense any objects within 10 m from a certain point by a physical or cyber error in a certain area. This experiment assumes that the sensor nodes within 10 m

of the center of a coordinate which changes randomly have failed unexpectedly. In order to calculate an average value, we carried out the same experiments 10 times repeatedly and compared the average value with the normal state. To determine the occurrence of errors, we experimented with 3 cases of failure at the 501st, 1001st, and 1501st periods.

Figure 6 indicates that the earlier an error occurs, the shorter the network lifetime is. However, the maximum reduction rate of failure at the 501st period was only 1.8%, which reveals that the smart backup mechanism of SOSAC can cope with external errors in a smart way.

In the second experiment, we tested the result if 5, 10, or 15 sensor nodes stopped their functions sporadically in the WSNs in comparison with the result in the normal state.

Figure 7 implies that the occurrence of sporadic errors barely influenced the network lifetime by using a smart backup mechanism. The results of the two experiments show that the smart backup mechanism of SOSAC is a robust smart mechanism capable of coping with some changes in the external environment.

5.2. Performance Comparison with HEED. In this section, the performance of SOSAC is compared with that of HEED, which is a well-known self-organized clustering and routing method. HEED maintains or improves the network performance by comparing the previous dispersing type methods using the self-organized method. In constructing the source code and simulation for SOSAC and HEED, we used Visual C++ of Visual Studio 2008.

5.2.1. Network Lifetime Comparison. In this experiment, all the cases were compared, ranging from the case in which the network lifetime of the WSN ceases as the energy of the first sensor node is drained to the case in which the network

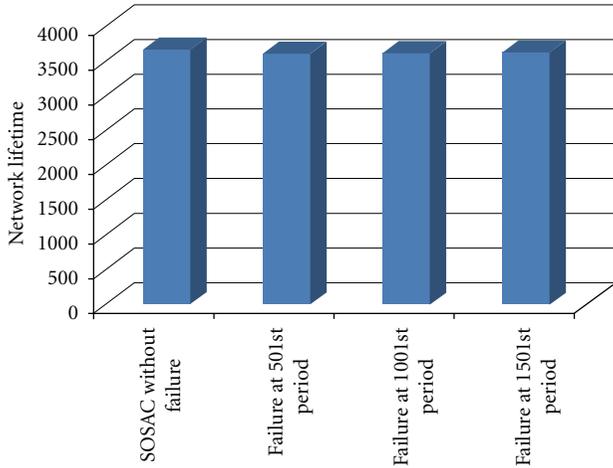


FIGURE 6: Experiment on concentrated failures.

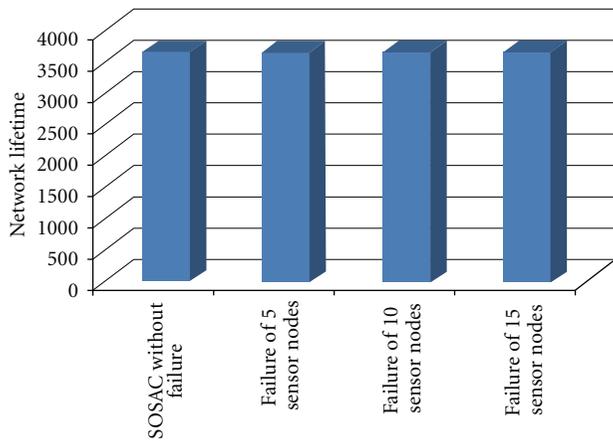


FIGURE 7: Experiment on sporadic failures.

lifetime of the WSN ceases as the energy of the 20th sensor node is drained.

Figure 8 shows that the network lifetime of SOSAC is between 61.7% (when the 1st sensor node is drained) and 8.5% (when the 20th sensor node is drained) longer than that of HEED. The network lifetimes until 20 sensor nodes had become drained were compared because it is impossible to observe the WSNs normally in the event that over 20% of the sensor nodes lose their functions due to the large vacuum in the sensing function.

We calculated the standard deviations of network lifetimes of SOSAC and HEED in Figure 9. On average, the standard deviation of SOSAC is 18.9% less than that of HEED, indicating that SOSAC shows more uniform performance than HEED.

5.2.2. Residual Energy Comparison. We compared the ratios of the residual energies in the WSNs to the network lifetimes of SOSAC and HEED.

In Figure 10, the Y-axis indicates the ratio of the residual energy in the WSN, and the X-axis indicates the number of drained sensor nodes. Figure 10 shows that the residual

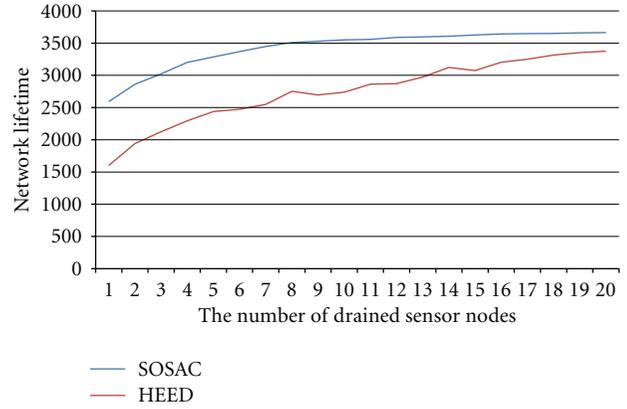


FIGURE 8: Comparison of network lifetime of SOSAC and HEED.

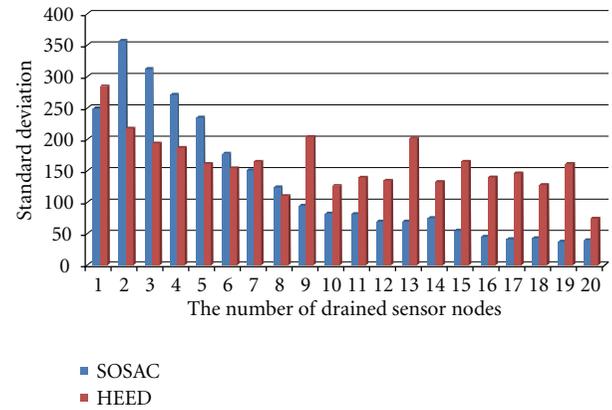


FIGURE 9: Comparison of standard deviations of network lifetimes for SOSAC and HEED.

energy of SOSAC is between 16.9% and 52.7% shorter than that of HEED. The comparison of the two experiments (Figures 8 and 10) indicates that SOSAC has a longer network lifetime than HEED, and that the sensor nodes of SOSAC consume energy more uniformly than those of HEED when the network lifetime of WSNs ceases. This implies that SOSAC controls certain sensor nodes so as not to consume energy quickly by balancing the energy consumptions of the sensor nodes based on the appropriate fitness value, which gives a relatively long network lifetime.

5.2.3. Scalability Comparison. We compared network lifetimes of SOSAC and HEED for different size sensor fields and different numbers of sensor nodes, while maintaining the same density of sensor nodes. For example, the 100 m* 100 m sensor field with 400 sensor nodes has four times as many as the 50 m* 50 m sensor field that has only 100 sensor nodes. Here the network lifetime of the WSN is defined for the 20% of sensor nodes that are drained.

In Figure 11, the Y axis indicates the network lifetime in periods, and the X axis is the size of the sensor fields. Figure 11 shows that the network lifetime of SOSAC was a maximum of 10.7% and a minimum of 2.2% longer than

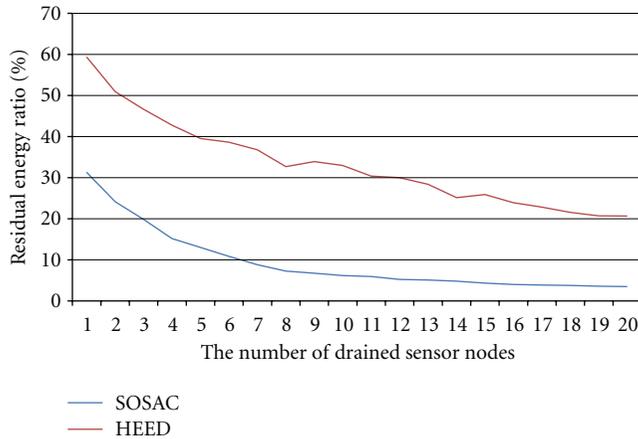


FIGURE 10: Experiment on residual energy ratios of SOSAC and HEED.

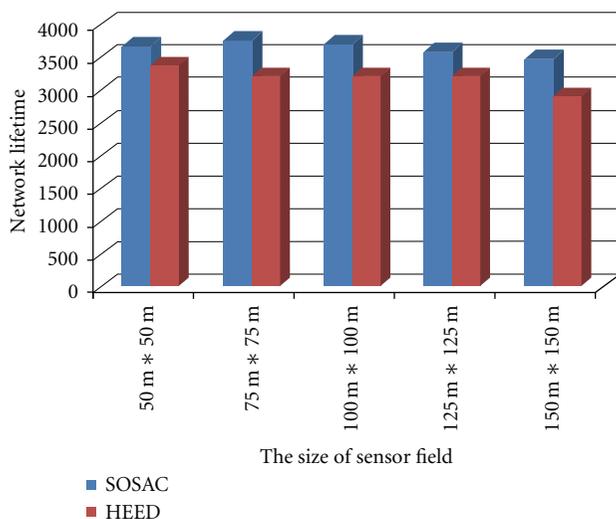


FIGURE 11: Experiment on scalability of SOSAC and HEED.

the network lifetime of HEED in different sensor field sizes and different numbers of sensor nodes. Hence, SOSAC shows good scalability in a smart way, which enables it to form clusters suitable for different sensor field sizes.

6. Conclusion

This paper has proposed a hierarchical clustering and routing model capable of maximizing the network lifetime through the decisionmaking of each sensor node based on local information by adopting a self-organized and smart-adaptive system in the design of the clustering and routing model of a WSN. The proposed method enables the sensor nodes to form clusters without a server or any external assistance, and the subsequent routing is performed based on it. The key advantage of this model is its ability to sense any environmental disturbances such as time changes, number of sensor nodes, and failures of sensor nodes using three smart adaptive mechanisms. The proposed method also

demonstrated superior performance compared to that of an existing self-organized clustering method.

A smart mechanism for WSNs that can cope with diverse changes in the environment needs to be developed in future study. Appropriate research examples are changes in the operation or mobile environment. In addition, research on self-organized and smart-adaptive communication methods for other communication networks holds the promise of valuable results in the near future.

Acknowledgment

This paper is the result of the research with the support of National Research Foundation of Korea (2009-0074081) using the funds from the Ministry of Education, Science and Technology in 2009.

References

- [1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [2] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.
- [3] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.
- [4] S. Mahfoudh and P. Minet, "Survey of energy efficient strategies in wireless ad hoc and sensor networks," in *Proceedings of the 7th International Conference on Networking (ICN '08)*, pp. 1–7, Cancun, Mexico, April 2008.
- [5] F. Dressler, *Self-Organization in Sensor and Actor Networks*, WILEY, New York, NY, USA., 2007.
- [6] E. Gelenbe, Z. Xu, and E. Seref, "Cognitive packet networks," in *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '99)*, pp. 47–54, November 1999.
- [7] E. Gelenbe, R. Lent, M. Gellman, P. Liu, and P. Su, "CPN and QoS driven smart routing in wired and wireless networks," *Lecture Notes in Computer Science*, vol. 2965, pp. 68–87, 2004.
- [8] M. D. Santo, A. Pietrosanto, P. Napoletano, and L. Carrubbo, "Knowledge based service systems," in *System Theory and Service Science: Integrating Three Perspectives in a New Service Agenda*, Social Science Electronic, New York, NY, USA, 2011.
- [9] N. Israr and I. U. Awan, "Multilayer cluster based energy efficient routing protocol for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 4, no. 2, pp. 176–193, 2008.
- [10] M. Younis, M. Youssef, and K. Arisha, "Energy-aware management for cluster-based sensor networks," *Computer Networks*, vol. 43, no. 5, pp. 649–668, 2003.
- [11] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff, "On energy provisioning and relay node placement for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2579–2590, 2005.
- [12] C. Prehofer and C. Bettstetter, "Self-organization in communication networks: principles and design paradigms," *IEEE Communications Magazine*, vol. 43, no. 7, pp. 78–85, 2005.
- [13] 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 Sciences (HICSS-33)*, p. 223, January 2000.
- [14] Y. Sun and X. Gu, "Clustering routing based maximizing lifetime for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 5, no. 1, p. 88, 2009.
 - [15] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.
 - [16] O. Younis, S. Fahmy, and P. Santi, "An architecture for robust sensor network communications," *International Journal of Distributed Sensor Networks*, vol. 1, no. 3-4, pp. 305–327, 2005.
 - [17] M. M. Hasan and J. P. Jue, "Survivable self-organization for prolonged lifetime in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 257156, 11 pages, 2011.
 - [18] D. L. Tennenhouse and D. J. Wetherall, "Towards an active network architecture," in *Proceedings of the Multimedia Computing and Networking (MMCN 96)*, and *Computer Communication Review*, vol. 26, no. 2, pp. 5–18, April 1996.
 - [19] E. Amir, S. McCanne, and R. Katz, "An active service framework and its application to real-time multimedia transcoding," *Computer Communication Review*, vol. 28, no. 4, pp. 178–189, 1998.
 - [20] A. Ren and G. Q. Maguire Jr., "A smart network with active services for wireless context-aware multimedia communications," in *Proceedings of the Wireless Communications and Systems, Emerging Technologies Symposium*, pp. 17.1–17.5, April 1999.
 - [21] S. Stone, M. Zyda, D. Brutzman, and J. Falby, "Mobile agents and smart networks for distributed simulations," in *Proceedings of the 14th Workshop on Standards Interoperability of Distributed Simulations*, pp. 909–917, Institute for Simulation and Training, Orlando, Fla, USA, March 1996.

Research Article

Distributed Algorithm for Real-Time Energy Optimal Routing Based on Dual Decomposition of Linear Programming

Jiří Trdlička and Zdeněk Hanzálek

Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University, Technická 2, 16627 Prague, Czech Republic

Correspondence should be addressed to Jiří Trdlička, jiri@trdlicka.cz

Received 14 June 2011; Revised 12 September 2011; Accepted 16 September 2011

Academic Editor: Yuhang Yang

Copyright © 2012 J. Trdlička and Z. Hanzálek. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work proposes a novel in-network distributed algorithm for real-time energy optimal routing in ad hoc and sensor networks for systems with linear cost functions and constant communication delays. The routing problem is described as a minimum-cost multicommodity network flow problem by linear programming and modified by network replication to a real-time aware form. Based on the convex programming theory we use dual decomposition to derive the distributed algorithm. Thanks to the exact mathematical derivation, the algorithm computes the energy optimal real-time routing. It uses only peer-to-peer communication between neighboring nodes and does not need any central node or knowledge about the whole network structure. Each node knows only the produced and collected data flow and the costs of its outgoing communication links. According to our knowledge, this work is the first, which solves the real-time routing problem with linear cost functions and constant communication delays, using the dual decomposition.

1. Introduction

Our work is focused on a distributed algorithm for data flow routing through multihop ad hoc and sensor networks, where all data has to be delivered to the destinations in time. An example of a target application is a network periodically sensing some consumption variables (like gas consumption, water consumption, etc.) in large objects. Each sensing device produces a data flow of a particular volume, which is supposed to be routed through the network. The objective is to optimize the energy consumption for the data transfer (minimal possible consumption in the whole network), while constrained by the communication capacities (maximum data volume which can be transferred by a device per time unit) and by maximum communication delay.

We assume a time division multiple access (TDMA) protocol (e.g., GTS allocation in IEEE 802.15.4 [1]) which ensures collision-free communication and causes constant communication delay. This approach is used quite often in industry and time-critical systems. Due to the TDMA mechanism assumed, the worst-case delay from the source node

to the sink node is the sum of the particular delays for each of the hops, assumed to be an integer (derived from the parameters like TDMA period, worst-case execution time of the communication stack. . .). In a particular setting, we may assume a unit hop delay (e.g., the same TDMA period). In this paper we assume the unit hop delay (the deadlines are expressed as the number of communication hops between devices) which is very transparent for the reader.

There are many communication protocols designed to find the exact energy optimal data routing in wireless sensor networks. However, to comply with the communication capacities, they usually need a central computational point with the knowledge of the actual network structure and parameters (e.g., [2]). The existence of such a computational point decreases the robustness of the system against the network damage. Furthermore, the routing of information about the actual network structure and parameters has to be solved in the case of the centralized algorithm.

In this paper, we propose a distributed algorithm, which computes the energy optimal real-time data routing without the need of any central computational or data point. The

algorithm supposes that each node knows only the cost (energy consumption per transmitted data unit) of its outgoing communication links and the data which is supposed to send and receive. The whole algorithm is mathematically derived using the convex programming theory. The main purpose of this paper is to present the principle of new distributed routing algorithms rather than to present an application-ready algorithm. We believe that the presented approach can lead to new, efficient, and highly adaptive routing algorithms for ad hoc and sensor networks. Moreover, the approach used in this work can be adapted for general distribution of convex optimization problems into the network.

2. Related Works

Traditionally, the routing problems for data networks are formulated as linear or convex multicommodity network flow routing problems, for example [2–4]. One of the advantages of this method is that several cost functions and constraints can be put together (e.g., different types of capacity and energy consumption constraints and real-time constraints). Using the same underlying model, we can easily combine the solution of different works focused on partial problems.

Several papers have been performed in the area of real-time routing in multihop wireless sensor networks. In [5], a well-known soft real-time communication protocol SPEED is presented. Several works use the relation between the message propagation speed and transmitting energy to balance the trade-off between the energy consumption and communication delay [6–8]. In [9] the authors deal with real-time communications over cluster-tree sensor networks, where they evaluate the end-to-end communication delay. However, none of these algorithms can ensure real-time and energy optimal data routing especially in high-loaded networks.

There are several works, which focus on the decomposition of network problems described by strictly convex optimization. A systematic presentation and classification of the decomposition techniques for network utility maximization (NUM) is presented in [10–12].

In [13–15], the authors use dual decomposition to decompose cross-layer optimization problems into the optimization of separated layers. The presented approaches lead to structural decomposition (e.g., to the routing layer, the capacity layer...) which is not suitable for the derivation of the in-network distributed algorithm. In [16], a general distributed algorithm for a strictly convex optimization problem with a common parameter for all nodes is presented. The decomposition of an optimal routing problem is presented, for example, in [17, 18], where the authors focus on the node-path formulation and based on dual decomposition derive an algorithm suitable for networks with a small number of communication paths.

In [19, 20], the authors use the duality theorem and node-link problem formulation to analytically derive the maximum network lifetime for linear networks. In [21], the authors use the node-link problem formulation and derive the distributed routing algorithm with an extension for the queuing delay. However, all these algorithms are limited to

strictly convex cost functions and fail in the case of linear cost functions.

This paper also continues our previous work. It joins and extends the results from [22, 23]. In [22], we have derived a distributed in-network routing algorithm based on dual decomposition. The algorithm consists of one iteration loop and is even able to solve problems with a linear objective function. In [23], we have presented a centralized algorithm for energy optimal real-time routing, which is represented as a min-cost multicommodity network flow routing problem with side constraints.

2.1. Contribution and Outline. The main contributions of this paper are

- (1) introduction of a new mathematically derived, distributed algorithm for energy optimal real-time routing based on network replication and dual decomposition;
- (2) introduction of a new approach for an in-network distribution of real-time routing problems with constant communication delay and linear objective function;
- (3) application of a distribution procedure for routing problems presented in [22] on a real-time routing problem [23].

The paper is organized as follows. Section 3 briefly describes the multicommodity network flow model. In Section 4 we define the real-time routing model. In Section 5, the distributed algorithm and its derivation are presented. An example and computational complexity experiments are given in Section 6. Section 7 concludes the paper and mentions the future work. The proof of the algorithm convergence is presented in the Appendix.

3. Multicommodity Network Flow Model

In this section, we briefly summarize the basic terminology and specify the multicommodity network flow model. For more details see, for example, [3, 4].

The network is represented by an oriented graph, where for each device able to send or receive data, a node of the graph exists. The nodes are labeled as $n = 1, \dots, N$. Directed communication links are represented as ordered pairs (n_1, n_2) of distinct nodes. The links are labeled as $l = 1, \dots, L$. We define the set of the links l leaving the node n as $\mathcal{O}(n)$ and the set of the links l incoming to node n as $\mathcal{I}(n)$. The network structure is described with two incidence matrices in node-link form. The matrix of the incoming links is denoted as A^+ , and the matrix of the outgoing links is denoted as A^- :

$$A_{n,l}^+ = \begin{cases} 1, & l \in \mathcal{I}(n) \text{ (link } l \text{ enters node } n), \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

$$A_{n,l}^- = \begin{cases} 1, & l \in \mathcal{O}(n) \text{ (link } l \text{ leaves node } n), \\ 0, & \text{otherwise.} \end{cases}$$

By m we denote an index of the communication demand and by \mathcal{M} we denote a set of all communication demands. The communication demands can be seen as the flow of various commodities incoming/leaving the network in some nodes. The flow of each communication demand has to satisfy the flow conservation law at each node (for a given commodity, the sum of the flow incoming to the node is equal to the sum of the flow leaving the node)

$$A^- \vec{x}^{(m)} + \vec{s}_{\text{out}}^{(m)} = A^+ \vec{x}^{(m)} + \vec{s}_{\text{in}}^{(m)} \quad \forall m \in \mathcal{M}, \quad (2)$$

where the column vector $\vec{s}_{\text{in}}^{(m)} \geq \vec{0}$ denotes the flow coming into the network, the $\vec{s}_{\text{out}}^{(m)} \geq \vec{0}$ denotes the flow leaving the network, and the $\vec{x}^{(m)} \geq \vec{0}$ denotes the flow routed through the network for demand m . Notice that a multisource multisink problem can be described in this way (e.g., the data gathering problem).

In this work, the node capacity constraints are used. They are the most common capacity constraint used for wireless networks. The total volume of the flow leaving one node has to satisfy the capacity constraint $D \sum_{m \in \mathcal{M}} \vec{x}^{(m)} \leq \vec{\mu}$, where $\vec{\mu} \geq \vec{0}$ is the column vector of the node capacities and matrix D describes the constraint structure (i.e., $D = A^-$).

In summary, the network flow model imposes the following constraints on the network flow variables $\vec{x}^{(m)}$:

$$\begin{aligned} A^- \vec{x}^{(m)} + \vec{s}_{\text{out}}^{(m)} &= A^+ \vec{x}^{(m)} + \vec{s}_{\text{in}}^{(m)} \quad \forall m \in \mathcal{M}, \\ D \sum_{m \in \mathcal{M}} \vec{x}^{(m)} &\leq \vec{\mu}, \\ \vec{x}^{(m)} &\geq \vec{0} \quad \forall m \in \mathcal{M}. \end{aligned} \quad (3)$$

The task of the total energy minimization is to minimize the cost function $f_{\text{cost}} = \vec{c}^T \sum_{m \in \mathcal{M}} \vec{x}^{(m)}$ by setting the flow vector $\vec{x}^{(m)}$ for all $m \in \mathcal{M}$ subject to the system of inequalities (3). Vector $\vec{c} > 0$ is a column vector of the energy consumption per data unit transmitted. The components of vector \vec{c} correspond with the energy consumption for the individual links in the network. Their values are usually determined directly from the transmission energy level in the nodes, which is needed for a reliable connection. This is done by the MAC layer of the communication protocol.

Another target application of this approach is a network, where the energy supplies of the nodes can be recharged with different maintenance costs (e.g., depending on the node reachability). In such an application, the recharge price is included in vector \vec{c} , and we minimize the network long-time maintenance cost.

4. Real-Time Multicommodity Network Flow Model

In this section, we extend the multicommodity flow model by real-time constraints, which guarantee a controllable routing delay through the network. Each communication demand has its own deadline, and the communication delay of each demand has to be shorter than its deadline. We model

the hop delay as an integer value associated with each communication link. For transparent model derivation, we assume the same communication delay over the entire network (i.e., each communication hop causes a delay equal to one). However, the model can be extended to a more general form, where the communication delays are integers.

4.1. Mathematical Model of Real-Time Routing. Let vector $\vec{x}^{(m,w)} \in R_+^L$ denote the flow of the communication demand m with an integer communication delay w in the network. Then the flow vector $\vec{x}^{(m)}$, independent of the flow delay of demand m , is equal to the sum of the flow vectors over all acceptable delays: $\vec{x}^{(m)} = \sum_{w=1}^{d^{(m)}} \vec{x}^{(m,w)}$, where $d^{(m)}$ denotes the deadline of the communication demand m . Using this equation, we can rewrite the capacity constraint from (3) into a new form:

$$D \sum_{m \in \mathcal{M}} \sum_{w=1}^{d^{(m)}} \vec{x}^{(m,w)} \leq \vec{\mu}. \quad (4)$$

Vector $\vec{s}^{(m,w)} \in R_+^N$ stands for the flow of the demand m leaving the network with the communication delay w , and vector $\vec{s}_{\text{in}}^{(m,w)} \in R_+^N$ denotes the flow of demand m coming into the network with the initial delay w . The flow of each demand may come into the network and leave it in more nodes. If all flow fragments of one demand coming into the network in different nodes have the same initial delay, then $\vec{s}_{\text{in}}^{(m,0)} = \vec{s}_{\text{in}}^{(m)}$, and $\vec{s}_{\text{in}}^{(m,w)} = 0$ for $w > 0$. The flow of demand m leaving the network prior to the deadline is

$$\vec{s}_{\text{out}}^{(m)} = \sum_{w=0}^{d^{(m)}} \vec{s}^{(m,w)} \quad \forall m \in \mathcal{M}. \quad (5)$$

Through (4) and (5), we have converted the real-time constraint (i.e., the delay has to be shorter than the deadline) to the structural constraint. Only the flow, whose delay is shorter than the deadline, is represented. The flow, which does not meet the deadline, violates the flow conservation law, and then the network flow constraints are not satisfied; that is, this solution is not feasible.

If the flow is sent through the network, the flow delay is increased by each communication hop. The flow of demand m coming into node n with communication delay w has to either leave the network in node n with the same delay w or reach the neighbor node with delay $w + 1$. The flow conservation law from (2) can be rewritten in the delay awareness form as

$$\begin{aligned} A^- \vec{x}^{(m,w+1)} + \vec{s}^{(m,w)} &= A^+ \vec{x}^{(m,w)} + \vec{s}_{\text{in}}^{(m,w)}, \\ \forall m \in \mathcal{M}, \quad 0 &\leq w \leq d^{(m)}. \end{aligned} \quad (6)$$

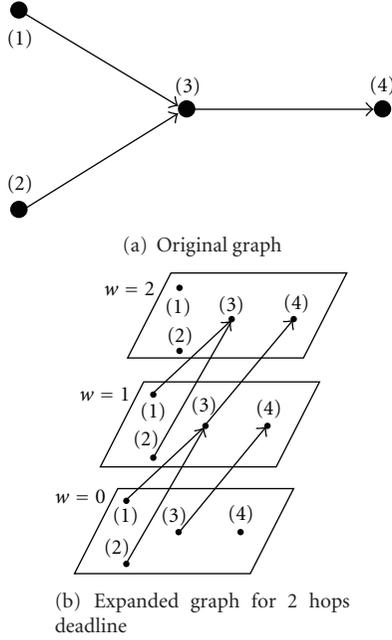


FIGURE 1: Example for the intuitive presentation.

In summary, the real-time energy optimal multicommodity flow routing problem can be written as

$$\begin{aligned}
 & \min_{\vec{x}, \vec{s}} c^T \sum_{m \in \mathcal{M}} \sum_{w=0}^{d^{(m)}} \vec{x}^{(m,w)} \\
 \text{subject to: } & A^- \vec{x}^{(m,w+1)} + \vec{s}^{(m,w)} = A^+ \vec{x}^{(m,w)} + \vec{s}_{\text{in}}^{(m,w)}, \\
 & \forall m \in \mathcal{M}, \quad 0 \leq w \leq d^{(m)}, \\
 & D \sum_{m \in \mathcal{M}} \sum_{w=1}^{d^{(m)}} \vec{x}^{(m,w)} \leq \vec{\mu}, \\
 & \sum_{w=0}^{d^{(m)}} \vec{s}^{(m,w)} = \vec{s}_{\text{out}}^{(m)} \quad \forall m \in \mathcal{M}, \\
 & \vec{x}^{(m,d^{(m)+1})} = \vec{0}, \\
 & \vec{x}^{(m,0)} = \vec{0}, \\
 & \vec{x}^{(m,w)} \geq \vec{0}, \\
 & \vec{s}^{(m,w)} \geq \vec{0}.
 \end{aligned} \tag{7}$$

Problem (7) describes the problem in a form, which can be solved in centralized way by any linear programming solver.

4.2. Intuitive Presentation of Extended Graph. In this section, we illustrate, in an intuitive way, the graph transformation, which has been discussed in Section 4.1 by mathematical equations. New variables have appeared for each communication link as well as new constraining equations for each

node. These variables and constraints can be seen as virtual layers of the network where each layer represents a different communication delay w . The number of the network layers is equal to the integer deadline of the demand m plus one (the number of allowed communication hops plus a zero layer). As consistent with the structure of (6), all communication links are redirected to the nodes in the higher layer, which means that the flow is routed not only in node space but also in delay space. Because the number of layers is limited by the deadline and the flow can leave the network only in virtual nodes of the sink nodes, all possible routings through this transformed network hold the deadlines. An easy example of the graph replication is shown in Figure 1.

5. Decomposition of the Routing Problem

To decompose the routing algorithm, we use the gradient optimization method to solve its dual problem. However, the linearity of the cost function of the problem (7) would cause oscillations in the gradient algorithm and prevents us finding the optimal solution. Therefore, we use the proximal-point method (for details see [4]) to modify the problem into a strictly convex form, which allows the usage of the gradient method. Moreover, we rewrite the problem into an equality form for a more transparent presentation:

$$\begin{aligned}
 & \min_{\vec{x}', \vec{z}', \vec{s}', \vec{z}'}'} g(\vec{x}, \vec{x}', \vec{s}, \vec{s}', \vec{z}, \vec{z}') \\
 \text{subject to: } & A^- \vec{x}^{(m,w+1)} + \vec{s}^{(m,w)} = A^+ \vec{x}^{(m,w)} + \vec{s}_{\text{in}}^{(m,w)}, \\
 & \forall m \in \mathcal{M}, \quad 0 \leq w \leq d^{(m)}, \\
 & D \sum_{m \in \mathcal{M}} \sum_{w=1}^{d^{(m)}} \vec{x}^{(m,w)} = \vec{\mu} - \vec{z}, \\
 & \sum_{w=0}^{d^{(m)}} \vec{s}^{(m,w)} = \vec{s}_{\text{out}}^{(m)} \quad \forall m \in \mathcal{M}, \\
 & \vec{x}^{(m,d^{(m)+1})} = \vec{0}, \\
 & \vec{x}^{(m,0)} = \vec{0}, \\
 & \vec{x}^{(m,w)} \geq \vec{0}, \\
 & \vec{s}^{(m,w)} \geq \vec{0},
 \end{aligned} \tag{8}$$

where the objective function $g(\vec{x}, \vec{x}', \vec{s}, \vec{s}', \vec{z}, \vec{z}')$ is

$$\begin{aligned}
 & g(\vec{x}, \vec{x}', \vec{s}, \vec{s}', \vec{z}, \vec{z}') \\
 & = c^T \sum_{m \in \mathcal{M}} \sum_{w=1}^{d^{(m)}} \vec{x}^{(m,w)} \\
 & \quad + \varepsilon (\vec{z} - \vec{z}')^T (\vec{z} - \vec{z}')
 \end{aligned}$$

$$\begin{aligned}
& + \varepsilon \sum_{m \in \mathcal{M}} \sum_{w=1}^{d^{(m)}} (\vec{x}^{(m,w)} - \vec{x}^{\prime\prime(m,w)})^T (\vec{x}^{(m,w)} - \vec{x}^{\prime\prime(m,w)}) \\
& + \varepsilon \sum_{m \in \mathcal{M}} \sum_{w=0}^{d^{(m)}} (\vec{s}^{(m,w)} - \vec{s}^{\prime\prime(m,w)})^T (\vec{s}^{(m,w)} - \vec{s}^{\prime\prime(m,w)}).
\end{aligned} \tag{9}$$

We have added slack variables $\vec{z} \geq \vec{0}$ into problem (8) to convert the problem into the equality form. The variables \vec{x}' , \vec{s}' , and \vec{z}' in (9) are the proximal-point variables, which have been added to convert the objective function from a linear form to a strictly convex one. Please notice that the set of optimal solutions for problem (8) is the same as for the problem (7). For the optimal solution of problem (8) it holds $\vec{x} = \vec{x}'$, $\vec{s} = \vec{s}'$, $\vec{z} = \vec{z}'$. In this way the routing problem has been separated into two nested subproblems. The internal subproblem is the minimization over the variables \vec{x} , \vec{s} , \vec{z} , and it is strictly convex. The outer subproblem minimizes the internal one by the proximal-point variables \vec{x}' , \vec{s}' , \vec{z}' .

5.1. Dual Problem. To solve the internal subproblem of (8) (minimization over the variables \vec{x} , \vec{z} , \vec{s}) we present its dual problem, which allows us to derive the distributable gradient algorithm. According to Slater's conditions (see e.g., [24]) the optimal solutions of the dual and primal problems have the same optimal values in this case.

The Lagrangian function of the problem (8) is

$$\begin{aligned}
L(\vec{x}, \vec{x}', \vec{s}, \vec{s}', \vec{z}, \vec{z}', \vec{\theta}, \vec{\lambda}, \vec{\gamma}) \\
& = g(\vec{x}, \vec{x}', \vec{s}, \vec{s}', \vec{z}, \vec{z}') \\
& + \sum_{m \in \mathcal{M}} \sum_{w=0}^{d^{(m)}} \vec{\theta}^{(m,w)T} (A^- \vec{x}^{(m,w+1)} - A^+ \vec{x}^{(m,w)} + \vec{s}^{(m,w)} \\
& \quad - \vec{s}_{\text{in}}^{(m,w)}) \\
& + \vec{\lambda}^T \left(D \sum_{m \in \mathcal{M}} \sum_{w=1}^{d^{(m)}} \vec{x}^{(m,w)} + \vec{z} - \vec{\mu} \right) \\
& + \vec{\gamma}^{(m)T} \left(\sum_{w=0}^{d^{(m)}} \vec{s}^{(m,w)} - \vec{s}_{\text{out}}^{(m)} \right),
\end{aligned} \tag{10}$$

where $\vec{x} \geq \vec{0}$, $\vec{s} \geq \vec{0}$, and $\vec{z} \geq \vec{0}$ are primal variables and $\vec{\theta}$, $\vec{\lambda}$ and $\vec{\gamma}$ are dual variables. The dual function W is

$$W(\vec{x}', \vec{s}', \vec{z}', \vec{\theta}, \vec{\lambda}, \vec{\gamma}) = \min_{\vec{x}, \vec{s}, \vec{z} \geq \vec{0}} L(\vec{x}, \vec{x}', \vec{s}, \vec{s}', \vec{z}, \vec{z}', \vec{\theta}, \vec{\lambda}, \vec{\gamma}). \tag{11}$$

The minimizers of the dual function (11) are

$$\begin{aligned}
\vec{x}_{\text{min}}^{(m,w)} & = \left[\vec{x}^{\prime\prime(m,w)} \right. \\
& \quad \left. - \frac{1}{2\varepsilon} (A^{-T} \vec{\theta}^{(m,w-1)} - A^{+T} \vec{\theta}^{(m,w)} + D^T \vec{\lambda} + \vec{c}) \right]^+, \\
\vec{z}_{\text{min}} & = \left[\vec{z}' - \frac{1}{2\varepsilon} \vec{\lambda} \right]^+, \\
\vec{s}_{\text{min}}^{(m,w)} & = \left[\vec{s}^{\prime\prime(m,w)} - \frac{1}{2\varepsilon} \vec{\gamma}^{(m)} \right]^+,
\end{aligned} \tag{12}$$

where symbol $[\cdot \cdot \cdot]^+$ denotes a positive or zero value in each component of vector $[\cdot \cdot \cdot]^+ = \max(\vec{0}, \dots)$ and $\vec{x}^{(m,0)} = \vec{x}^{(m,d^{(m)+1})} = \vec{0}$.

The dual problem of the internal subproblem of (8) is

$$U(\vec{x}', \vec{s}', \vec{z}') = \max_{\vec{\theta}, \vec{\lambda}, \vec{\gamma}} W(\vec{x}', \vec{s}', \vec{z}', \vec{\theta}, \vec{\lambda}, \vec{\gamma}). \tag{13}$$

The gradients of the dual function (11) are

$$\begin{aligned}
\nabla_{\theta} W^{(m,w)} & = A^- \vec{x}_{\text{min}}^{(m,w+1)} - A^+ \vec{x}_{\text{min}}^{(m,w)} + \vec{s}_{\text{min}}^{(m,w)} - \vec{s}_{\text{in}}^{(m,w)}, \\
\nabla_{\lambda} W^{(m)} & = D \sum_{m \in \mathcal{M}} \sum_{w=1}^{d^{(m)}} \vec{x}_{\text{min}}^{(m,w)} + \vec{z}_{\text{min}} - \vec{\mu}, \\
\nabla_{\gamma} W & = \sum_{w=0}^{d^{(m)}} \vec{s}_{\text{min}}^{(m,w)} - \vec{s}_{\text{out}}^{(m)}.
\end{aligned} \tag{14}$$

The gradients of the dual problem (13) are

$$\begin{aligned}
\nabla_{x'} U^{(m,w)} & = -2\varepsilon (\vec{x}_{\text{min}}^{(m,w)} - \vec{x}^{\prime\prime(m,w)}), \\
\nabla_{z'} U & = -2\varepsilon (\vec{z}_{\text{min}} - \vec{z}'), \\
\nabla_{s'} U^{(m,w)} & = -2\varepsilon (\vec{s}_{\text{min}}^{(m,w)} - \vec{s}^{\prime\prime(m,w)}).
\end{aligned} \tag{15}$$

5.2. Dual Gradient Algorithm. Using the dual problem (13) and the dual function (11) we rewrite the routing problem (8) into the form:

$$\min_{\vec{x}', \vec{z}', \vec{s}'} \max_{\vec{\theta}, \vec{\lambda}, \vec{\gamma}} \min_{\vec{x}, \vec{s}, \vec{z} \geq \vec{0}} L(\vec{x}, \vec{x}', \vec{s}, \vec{s}', \vec{z}, \vec{z}', \vec{\theta}, \vec{\lambda}, \vec{\gamma}). \tag{16}$$

A gradient algorithm created from problem (16) consists of 2 nested loops, and it is described as (17). The internal loop solves the dual problem (13) using the gradients (14). The

outer loop minimizes problem (16) over the proximal-point variables using the gradients (15):

LOOP 1

LOOP 2

Compute the \vec{x}_{\min} , \vec{z}_{\min} , $\vec{s}_{\min}^{(m,w)}$

according to Equations (13), and

$$\begin{aligned}\vec{\theta}^{(m,w)} &= \vec{\theta}^{(m,w)} + \alpha \nabla_{\theta} W^{(m,w)} \\ \vec{\gamma}^{(m)} &= \vec{\gamma}^{(m)} + \alpha \nabla_{\gamma} W^{(m)} \\ \vec{\lambda} &= \vec{\lambda} + \alpha \nabla_{\lambda} W\end{aligned}\quad (17)$$

END 2

$$\begin{aligned}\vec{x}^{\prime\prime(m,w)} &= \vec{x}^{\prime(m,w)} + \alpha \nabla_{x'} U^{(m,w)} \\ \vec{z}^{\prime} &= \vec{z}^{\prime} + \alpha \nabla_{z'} U \\ \vec{s}^{\prime(m,w)} &= \vec{s}^{\prime(m,w)} + \alpha \nabla_{s'} U^{(m,w)}\end{aligned}$$

END 1

The $\alpha > 0$ is a constant step size of the algorithm.

The important property of algorithm (17) is that it can be distributed as an in-network algorithm. Each node is responsible for the computation of the flow volume of the links leaving the node and for computation of the other corresponding variables. All the components of the variable vectors are a function of the node local variables and of the variables of the neighboring nodes that are within one hop communication distance. Only peer-to-peer communication during the algorithm is needed.

However, the distributed version of such an algorithm would have problems with the termination of the internal loop and with the synchronization of the loops between the nodes. Moreover the nested loops would cause an increase in the iterations.

To overcome these problems, we join both loops of the gradient algorithm into one loop, where we update all the variables simultaneously. Using (12), (14), and (15) we derive the iterative algorithm which is presented in Table 1, where k denotes the iteration number.

The correctness of such an algorithm is not seen directly from its derivation and has to be proven. The proof of the algorithm convergence is not a trivial problem, and its simplified version is presented in the Appendix. A necessary condition for the algorithm convergence assumed in the proof is $\alpha < 1/2\epsilon$. Moreover, we have performed several simulation experiments to test the algorithm convergence in Section 6.

The initial variables $\vec{x}_0^{\prime(m,w)}$, $\vec{s}_0^{\prime(m,w)}$, $\vec{z}_0^{\prime(m,w)}$, $\vec{\theta}_0^{(m,w)}$, $\vec{\lambda}_0$, $\vec{\gamma}_0^{(m)}$ are set to arbitrary initial values. The closer the values are to the final solution, the faster the algorithm converges. This property can be used in the case of minor changes of the network structure during its operation or in case of a precomputed routing, for example, based on Dijkstra's algorithm.

5.3. *Variables Initialization.* As we have mentioned in Section 5.2, the number of iterations depends on the initial

TABLE 1: Distributed routing algorithm.

(1) Initialize variables $\vec{x}_0^{\prime(m,w)}$, $\vec{s}_0^{\prime(m,w)}$, $\vec{z}_0^{\prime(m,w)}$, $\vec{\theta}_0^{(m,w)}$, $\vec{\lambda}_0$, $\vec{\gamma}_0^{(m)}$.
(2) Compute primal variables \vec{x}_k , \vec{z}_k , \vec{s}_k according to: $\vec{x}_k = [\vec{x}_k^{\prime(m,w)} - (1/2\epsilon)(A^{-T} \vec{\theta}_k^{(m,w-1)} - A^{+T} \vec{\theta}_k^{(m,w)} + D^T \vec{\lambda}_k + \vec{c})]^+$ $\vec{z}_k = [\vec{z}_k^{\prime} - (1/2\epsilon) \vec{\lambda}_k]^+$ $\vec{s}_k^{(m,w)} = [\vec{s}_k^{\prime(m,w)} - (1/2\epsilon) \vec{\gamma}_k^{(m)}]^+$ $\vec{x}_k^{(m,0)} = \vec{x}_k^{(m,d^{(m)+1)}} = \vec{0}.$
(3) Send/Receive the primal variables \vec{x}_k , \vec{z}_k , \vec{s}_k to/from neighboring nodes.
(4) Compute dual variables $\vec{\theta}_{k+1}$, $\vec{\lambda}_{k+1}$, $\vec{\gamma}_{k+1}$: $\vec{\theta}_{k+1} = \vec{\theta}_k^{(m,w)} + \alpha(A^{-T} \vec{x}_k^{(m,w+1)} - A^{+T} \vec{x}_k^{(m,w)} + \vec{s}_k^{(m,w)} - \vec{s}_{\text{in}}^{(m,w)})$ $\vec{\lambda}_{k+1} = \vec{\lambda}_k + \alpha(D \sum_{m \in \mathcal{M}} \sum_{w=1}^{d^{(m)}} \vec{x}_k^{(m,w)} + \vec{z}_k - \vec{\mu})$ $\vec{\gamma}_{k+1}^{(m)} = \vec{\gamma}_k^{(m)} + \alpha(\sum_{w=0}^{d^{(m)}} \vec{s}_k^{(m,w)} - \vec{s}_{\text{out}}^{(m)}).$
(5) Compute proximal-point variables \vec{x}_{k+1}^{\prime} , \vec{s}_{k+1}^{\prime} , \vec{z}_{k+1}^{\prime} $\vec{x}_{k+1}^{\prime(m,w)} = \vec{x}_k^{\prime(m,w)} + \alpha(-2\epsilon(\vec{x}_k^{(m,w)} - \vec{x}_k^{\prime(m,w)}))$ $\vec{z}_{k+1}^{\prime} = \vec{z}_k^{\prime} + \alpha(-2\epsilon(\vec{z}_k - \vec{z}_k^{\prime}))$ $\vec{s}_{k+1}^{\prime(m,w)} = \vec{s}_k^{\prime(m,w)} + \alpha(-2\epsilon(\vec{s}_k^{(m,w)} - \vec{s}_k^{\prime(m,w)})).$
(6) Send/Receive the dual variables $\vec{\theta}_{k+1}$, $\vec{\lambda}_{k+1}$, $\vec{\gamma}_{k+1}$ to/from the neighboring nodes.
(7) Set $k = k + 1$ and start new iteration in step 2.

variables $\vec{x}_0^{\prime(m,w)}$, $\vec{s}_0^{\prime(m,w)}$, $\vec{z}_0^{\prime(m,w)}$, $\vec{\theta}_0^{(m,w)}$, $\vec{\lambda}_0$, $\vec{\gamma}_0^{(m)}$. The closer the initial variables are to the final solution the less number of iterations are needed. In this work, we focus only on the dual variables. The proximal-point variables $\vec{x}_0^{\prime(m,w)}$, $\vec{s}_0^{\prime(m,w)}$, $\vec{z}_0^{\prime(m,w)}$ are set to zero. The variables $\vec{\lambda}_0$, $\vec{\gamma}_0^{(m)}$ cannot be estimated without the knowledge of the final routing. We set them to zero.

Let us suppose some initial variable $\vec{\theta}_0$ which satisfies a condition:

$$A^{-T} \vec{\theta}_0^{(m,w-1)} - A^{+T} \vec{\theta}_0^{(m,w)} + D^T \vec{\lambda}_0 + \vec{c} \geq \vec{0}. \quad (18)$$

According to (12), such initial variables do not cause any changes of the algorithm variables on its own (i.e., no variable changes if $\vec{s}_{\text{out}}^{(m)} = \vec{s}_{\text{in}}^{(m)} = 0$).

According to the complementary slackness condition (for details see, e.g., [4]) for the optimal solution holds for two neighboring nodes: if $(\theta_{k,l^+}^{(m,w-1)} - \theta_{k,l^-}^{(m,w)}) < c_l$ then $x_{k,l}^{(m,w)} = 0$, if $(\theta_{k,l^+}^{(m,w-1)} - \theta_{k,l^-}^{(m,w)}) = c_l$ then $x_{k,l}^{(m,w)} \geq 0$, and the link l is a part of the shortest path for the demand $m \in \mathcal{M}$. (We use the index l to denote the vector component corresponding to the link l , l^+ to denote end node of the link, and l^- to denote the start node of the link.) This fact leads us directly to Dijkstra's algorithm which can be used in a distributed way. If for all sink nodes n_{out} we set $\theta_{0,n_{\text{out}}}^{(m,w)} = 0$ for all $m \in \mathcal{M}$, $0 \leq w \leq d^{(m)}$ and for the other nodes n we set $\theta_{0,n}^{(m,w)} = \text{the shortest distance to the sink node}$, we get an initial setting, which satisfies condition (18). Moreover, this initial setting is much closer to the optimal solution than the setting $\theta_{0,n}^{(m,w)} = 0$ for all n, m, w . In Section 6.3, we present several experiments to evaluate the heuristic behavior in comparison with the zero initial setting. Please notice that due to the capacity constraints, the heuristic

solution does not need to be equal to the final optimal solution. According to our experiments it rapidly decreases the number of iterations.

6. Experiments

To demonstrate the behavior and the correctness of the distributed routing algorithm, we have performed several experiments in Matlab. We have focused on a basic problem, where for each communication demand one node sends data flow to one sink node (i.e., a multicommodity mono-source, mono-sink problem). Therefore, $s_{in,n_1}^{(m)} = 1$ for the source node n_1 and $s_{out,n_2}^{(m)} = 1$ for the sink node n_2 of the communication demand m .

The random networks for the experiments have been constructed as follows. We consider a square field of size $[size \times size]$, where the *size* is changing during the experiments. The field is divided into subsquares of size $[1 \times 1]$. One node is randomly placed into each subsquare, and the communication distance is set to 1.7 (i.e., node n_1 can communicate with node n_2 , if and only if their Euclidean distance is less than 1.7). Please notice that our network is close to the “unit-disk network” [2]. The communication costs \vec{c} per transmitted data flow unit have been set as the square power of the distance between the nodes. The link capacities have been set to two $\mu_n = 2$, and the maximum number of communication hops to $d^{(m)} = 6$ for all $m \in \mathcal{M}$. The constants of the algorithm have been set as $\alpha = 0.03$ and $\varepsilon = 0.3$. The initial values $\vec{x}_0^{(m,w)}$, $\vec{s}_0^{(m,w)}$, $\vec{\theta}_0^{(m,w)}$, $\vec{\lambda}_0$, $\vec{\gamma}_0^{(m)}$ have been set to 0 and $z_{0,n}'' = \mu_n$ for all experiments except for those in Sections 6.2 and 6.4. Only feasible problems are used.

During the experiments we evaluate k as a number of iterations needed to achieve less than 1% deviation of the cost function from the optimal value, less than 1% capacity violation, and less than 1% flow conservation law violation during the last 500 iterations. The 500 iterations are included in the statistics. (the optimal value was computed separately by a centralized algorithm for evaluation purposes only).

Unfortunately, as the presented approach of dual decomposition of the routing problems in node-link form is a new technique, there are only few works in this area at this moment. The most similar work can be found in [21]. However, it is focused on slightly different problems. It uses different energy consumption and communication delay models, and it is limited only on problems with strictly convex objective functions. Our work is focused on the problems with linear objective functions and presents different quantities than [21]. Due to the differences in the used models and algorithms derivations the experimental comparison would have a disputable contribution and would be strongly dependent on the chosen problems.

6.1. Example. To present the resulting optimal data flow routing in the network we have performed an experiment based on the network described above. The field size has been set to 9 (i.e., 81 nodes in the network), and the number of communication demands has been set to 8. The link

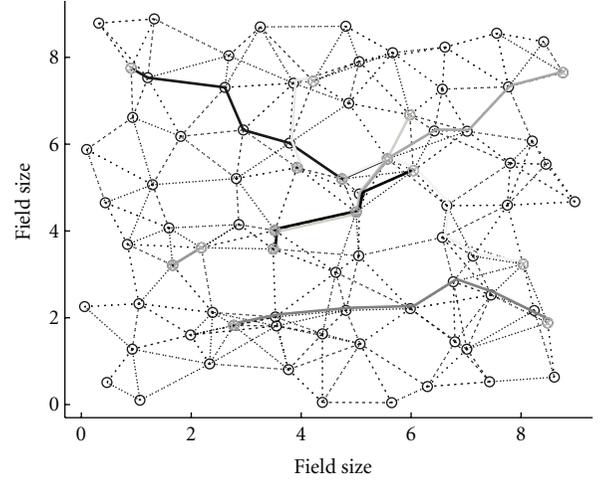


FIGURE 2: Optimal data flow routing (multicommodity, mono-source, mono-sink problem).

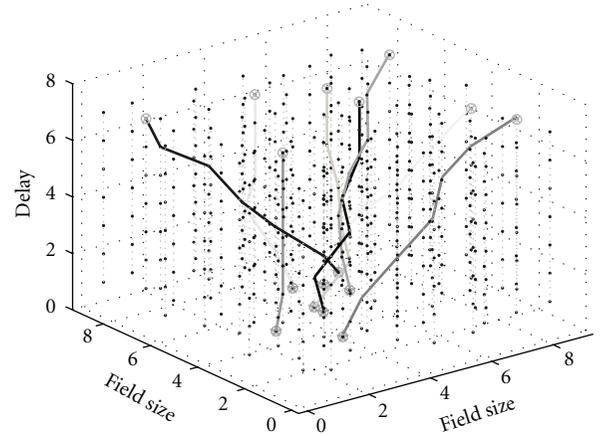


FIGURE 3: Optimal data flow routing in node-delay space.

capacities have been set to $\mu_n = 4$ for this problem. The initial variables $\vec{\theta}_0$ have been set according to Section 5.3.

The optimal data flow routing in the network is shown in Figure 2. The 3D routing model in the node-delay space is presented in Figure 3 where the vertical axis represents the communication delay. No communication demand routing has more than 6 communication hops.

The algorithm behavior can be seen in [25] on video. The flow routing in the network and in the node-delay space and the progress of the flow conservation law are presented.

6.2. Algorithm Convergence. We have performed a set of experiments with random initial variables on random networks, and we have evaluated the algorithm convergence. The field size has been set to 7. There are 8 communication demands in the network. The initial values have been set randomly from intervals: $\vec{x}_0^{(m,w)}$, $\vec{s}_0^{(m,w)}$, $\vec{z}_0^{(m,w)} \in \langle 0, 2 \rangle$ for proximal-point variables, and $\vec{\theta}_0^{(m,w)}$, $\vec{\lambda}_0$, $\vec{\gamma}_0^{(m)} \in \langle 0, 20 \rangle$ for dual variables. The intervals have been chosen as the double value of typical optimal values.

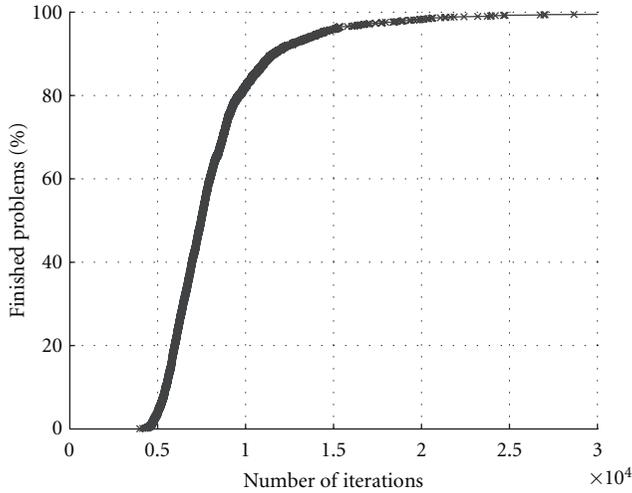


FIGURE 4: Algorithm convergence with random initial variables.

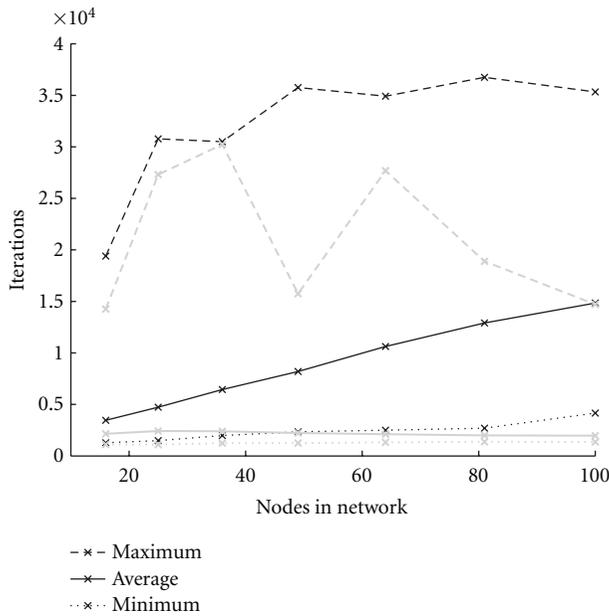


FIGURE 5: The number of iterations in relation to the number of nodes. In black, for zero initial variables. In gray, for initialization according to Section 5.3.

The algorithm has been run 2000 times on random networks. The results are presented in Figure 4. There, the number of iterations is placed on the horizontal axis, and the number of experiments, which has been finished before the number of iterations, is on the vertical axis.

This experiment provides an important practical verification of the theoretical proof of the algorithm convergence. 95% of the experiments have been finished before 14300 iterations.

6.3. Number of Iterations. To demonstrate the statistical behavior of the algorithm, we have gradually increased the number of nodes from 16 to 100 (i.e., the field size from

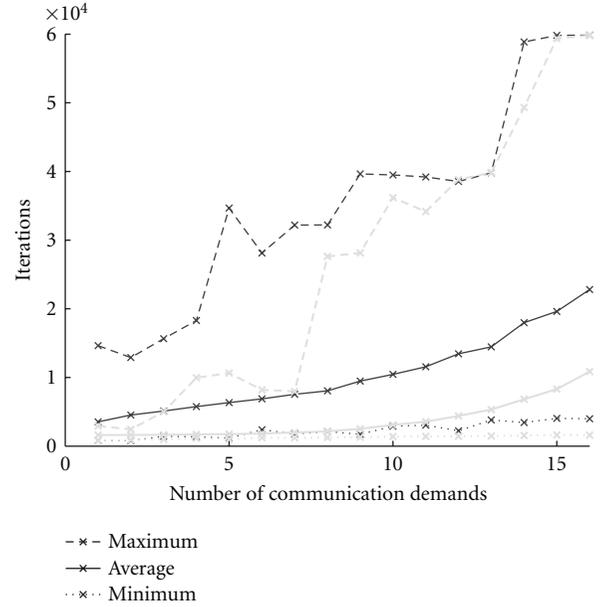


FIGURE 6: The number of iterations in relation to the number of communication demands. In black, for zero initial variables. In gray, for initialization according to Section 5.3.

4 to 10) for 8 communication demands. The computation has been repeated, on random networks 1000 times for each number of nodes.

The results have been evaluated as a maximum, average, and minimum number of iterations needed to achieve the optimal value, and it is presented in Figure 5. There, the experiment progress is presented for the basic algorithm without the initial heuristic in black (the initial variables have been set to zero) and for the algorithm with the initial heuristic according to Section 5.3 in gray.

Similar experiments have been performed for the iterations dependence on the number of communication demands. We have gradually increased the number of communication demands from 1 to 16 in networks with 49 nodes. The computation has been repeated, on random networks 1000 times for each number of communication demands. The results are presented in Figure 6 for both algorithms with and without the initial heuristic.

In Table 2, we present the percentages of infeasible problems for each number of communication demands. The values say how much percent of the generated problems have not been feasible due to the capacity and real-time constraints. From Table 2, it follows that the network has been close to saturation and that the constraints affect the final routings.

The important outcome of these experiments is the observation that the number of iterations is approximately linear. It follows that the algorithm is easily applied to networks with many nodes. We can see a significant improvement of the initial heuristic in the Figures.

6.4. Network Change. The advantage of the one-loop algorithm presented in this work is that it can automatically

TABLE 2: Percentage of infeasible problems in the experiment for the number of communication demands.

Number of comm. demands	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Infeasible problems [%]	0.5	2.0	3.3	4.0	3.9	5.8	7.0	8.6	12.0	13.8	17.5	20.7	27.1	33.1	39.3	48.1

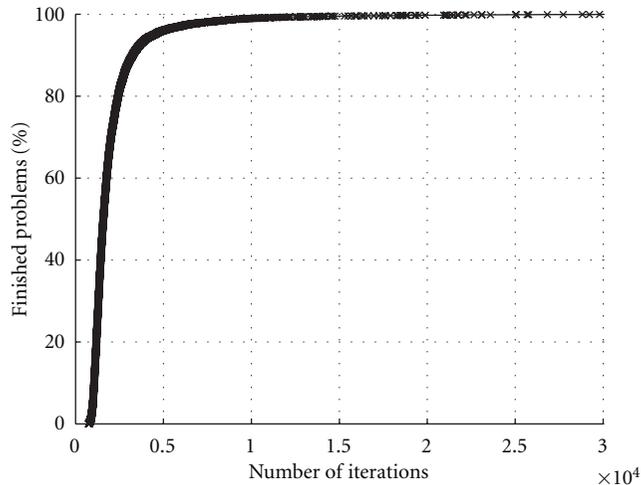


FIGURE 7: Algorithm convergence for network change simulation.

adjust the data routing in case of network changes. In order to evaluate the algorithm behavior in this case, we have simulated a dying node in the network as follow.

(1) We generated a random network with communication demands and found the optimal solution. (2) In the original problem from step 1, we removed one node and measured the number of iterations to find the new optimal solution. (3) We repeated step 2 for 15 different nodes with the largest data flow.

The simulation has been performed for 942 random original networks with field size set to 7 and with 8 communication demands (i.e., $942 \times 15 = 14130$ experiments).

The results are presented in Figure 7. There, the number of iterations is placed on the horizontal axis, and the number of experiments, which has been finished before the number of iterations, is on the vertical axis.

95% of the experiments have been finished before 4466 iterations. Moreover, less than 15.0% of the dual variables and 2.0% of the primal variables have been changed during the experiments, which significantly decrease the amount of transmitted data.

Video presentation can be seen in [25].

7. Conclusion

We have presented a distributed algorithm for the real-time energy optimal routing in ad hoc and sensor networks for systems with linear cost functions and constant communication delays. Such networks are used very often in the industrial environment, where TDMA-derived mechanisms or mechanisms with a significant stack computation delay are used. We have described the routing problem as a multicommodity network flow optimization problem, modified it into

a time aware form, and used the dual decomposition method to derive the distributed algorithm. The algorithm does not need any central computational node with knowledge about the whole network structure, and it only uses local communication between the neighboring nodes. This rapidly increases the robustness of the algorithm in the case of partial network damage. We have performed several simulations to evaluate the algorithm behavior and to test its convergence. The mathematical proof of the algorithm convergence is available in the Appendix.

The main purpose of this paper was to present the basic concept of a new in-network distributed routing algorithms for real-time data and its derivation. From the experimental section it is seen that the algorithm is not application-ready because of the high number of iterations, which leads to high number of communications. However, the main strength of the algorithm is not to find the whole optimal routing in an unknown network, but to adapt an existing routing in case of local network changes (dead/new node, loss of connection, communication costs, or capacities change, etc.) where the number of data communications could be significantly decreased. Moreover, the algorithm can easily adapt to slow network changes. The algorithm is suitable for static networks with sporadic changes or for networks with slow continuous changes, so the routing can be adjusted.

According to our preliminary experiments the number of iterations can be significantly decreased in future work. The algorithm can be extended by heuristics based on the partial knowledge about the network structure (e.g., node geographical position) and heuristics based on Newton's method. The results of our preliminary testing indicate that Newton's method-based heuristics can decrease the number of iterations more than 3 times.

Considering the fact that the algorithm is based on Linear programming formulation, we believe that the principle of the algorithm and the approach used to its derivation can be used to solve many different problems in the sensor networks area, like resource sharing, network localization, object tracking, and so forth.

Appendix

Proof of the Algorithm Convergence

We prove the convergence of the algorithm presented in Table 1 of the paper for $\alpha \rightarrow 0$ and for $\alpha < 1/(2\epsilon)$ as follows. First, we rewrite the equations of the algorithm into a more transparent form, which significantly simplifies the proof presentation. Then we define a merit function P_k such that $P_k \geq 0$ and $P_k = 0$ for the optimal solution, and we show that P_k is nonincreasing during the algorithm computation. Next, we assume the merit function P_k to be nondecreasing for all $k \geq k_0$ and show, for feasible problems, that for all

$k \geq k_0$ we have the optimal solution. We use the marking $x_{k,i}$, $\nabla_x L_{k,i}$, $[A^T(A\vec{x}_k - \vec{b})]_i$ to denote the i th component of the vectors. We simplify the notation of L_k , P_k , d_k , and so forth, instead of $L_k(x_k, y_k, \theta_k)$, $P_k(x_k, y_k, \theta_k)$, and so forth, for a more compact and transparent description. Let us remind the reader that $\varepsilon > 0$, $\vec{x}_k \geq \vec{0}$, $\vec{c} > \vec{0}$ and that according to Slater's conditions [24] the optimal solutions of the dual and primal problems have the same optimal values in this case.

We rewrite the problem (8) into a more transparent form:

$$\begin{aligned} \min_{\vec{y}} \min_{\vec{x}} \quad & \vec{c}^T \vec{x} + \varepsilon (\vec{x} - \vec{y})^T (\vec{x} - \vec{y}) \\ \text{subject to:} \quad & \tilde{A} \vec{x} = \vec{b}, \\ & \vec{x} \geq \vec{0}. \end{aligned} \quad (\text{A.1})$$

An example of the matrix and vectors transformation is

$$\begin{aligned} \vec{x}_k = \begin{bmatrix} \vec{x}_k^{-(1,0)} \\ \vdots \\ \vec{x}_k^{-(1,d^{(1)})} \\ \vdots \\ \vec{x}_k^{-(M,d^{(M)})} \\ \vec{s}_k^{-(1,0)} \\ \vdots \\ \vec{s}_k^{-(1,d^{(1)})} \\ \vdots \\ \vec{s}_k^{-(M,d^{(M)})} \\ \vec{z}_k \end{bmatrix}, \quad \vec{y}_k = \begin{bmatrix} \vec{x}_k^{\prime(1,0)} \\ \vdots \\ \vec{x}_k^{\prime(1,d^{(1)})} \\ \vdots \\ \vec{x}_k^{\prime(M,d^{(M)})} \\ \vec{s}_k^{\prime(1,0)} \\ \vdots \\ \vec{s}_k^{\prime(1,d^{(1)})} \\ \vdots \\ \vec{s}_k^{\prime(M,d^{(M)})} \\ \vec{z}_k^{\prime} \end{bmatrix}, \\ \vec{c} = \begin{bmatrix} \vec{c} \\ \vdots \\ \vec{c} \\ \vdots \\ \vec{c} \\ \vec{0} \\ \vdots \\ \vec{0} \\ \vdots \\ \vec{0} \\ \vec{0} \end{bmatrix}, \quad \vec{\theta}_k = \begin{bmatrix} \vec{\theta}_k^{(1,0)} \\ \vdots \\ \vec{\theta}_k^{(1,d^{(1)})} \\ \vdots \\ \vec{\theta}_k^{(M,d^{(M)})} \\ \vec{\gamma}_k^{(1)} \\ \vdots \\ \vec{\gamma}_k^{(M)} \\ \vec{\lambda}_k \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} \vec{s}_{\text{in}}^{-(1,0)} \\ \vdots \\ \vec{s}_{\text{in}}^{-(1,d^{(1)})} \\ \vdots \\ \vec{s}_{\text{in}}^{-(M,d^{(M)})} \\ \vec{s}_{\text{out}}^{-(1)} \\ \vdots \\ \vec{s}_{\text{out}}^{-(M)} \\ \vec{\mu} \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} \tilde{A} &= \begin{pmatrix} \tilde{B} & \tilde{I} & 0 \\ 0 & \tilde{S} & 0 \\ \tilde{D} & 0 & I \end{pmatrix}, \quad \tilde{B} = \begin{pmatrix} \tilde{B}^1 & 0 & 0 \\ & \ddots & \\ 0 & 0 & \tilde{B}^M \end{pmatrix}, \\ \tilde{B}^m &= \begin{pmatrix} A^+ & A^- & 0 & 0 \\ 0 & A^+ & A^- & 0 \\ & & \ddots & \\ 0 & 0 & A^+ & A^- \end{pmatrix}, \\ \tilde{S} &= \begin{pmatrix} \tilde{S}^1 & 0 & 0 \\ & \ddots & \\ 0 & 0 & \tilde{S}^M \end{pmatrix}, \quad \tilde{S}^m = (I \ I \ \dots \ I), \\ & \quad \tilde{D} = (D \ D \ \dots \ D), \\ \tilde{I} &= \begin{pmatrix} I & 0 & 0 \\ & \ddots & \\ 0 & 0 & I \end{pmatrix}, \end{aligned} \quad (\text{A.2})$$

where I is an identity matrix. If $M = |\mathcal{M}|$ the sizes of new vectors and matrices are

$$\begin{aligned} \vec{x}_k, \vec{y}_k, \vec{c} &: \left[N \sum_{m \in \mathcal{M}} d^{(m)} + MN + N \times 1 \right], \\ \vec{\theta}_k, \vec{b} &: \left[L \sum_{m \in \mathcal{M}} d^{(m)} + N \sum_{m \in \mathcal{M}} d^{(m)} + N \times 1 \right], \\ \tilde{A} &: \left[L \sum_{m \in \mathcal{M}} d^{(m)} + N \sum_{m \in \mathcal{M}} d^{(m)} + N \times N \sum_{m \in \mathcal{M}} d^{(m)} \right. \\ & \quad \left. + MN + N \right], \\ \tilde{B}^m &: \left[Nd^{(m)} \times Ld^{(m)} \right], \\ \tilde{B} &: \left[N \sum_{m \in \mathcal{M}} d^{(m)} \times L \sum_{m \in \mathcal{M}} d^{(m)} \right], \\ \tilde{S}^m &: \left[N \times Nd^{(m)} \right], \\ \tilde{S} &: \left[NM \times N \sum_{m \in \mathcal{M}} d^{(m)} \right], \\ \tilde{D} &: \left[N \times L \sum_{m \in \mathcal{M}} d^{(m)} \right], \\ \tilde{I} &: \left[N \sum_{m \in \mathcal{M}} d^{(m)} \times N \sum_{m \in \mathcal{M}} d^{(m)} \right]. \end{aligned} \quad (\text{A.3})$$

The Lagrangian function of the problem (A.1) is

$$L_k = \vec{c}^T \vec{x}_k + \varepsilon (\vec{x}_k - \vec{y}_k)^T (\vec{x}_k - \vec{y}_k) + \vec{\theta}_k^T (\tilde{A} \vec{x}_k - \vec{b}) \quad (\text{A.4})$$

and the algorithm equations from Table 1 can be expressed as

$$\begin{aligned} \vec{x}_k &= \left[\vec{y}_k - \frac{1}{2\varepsilon} (\vec{c} + \tilde{A}^T \vec{\theta}_k) \right]^+, \\ \vec{\theta}_{k+1} &= \vec{\theta}_k + \alpha \nabla_{\theta} L_k, \\ \vec{y}_{k+1} &= \vec{y}_k - \alpha \nabla_y L_k. \end{aligned} \quad (\text{A.5})$$

It can be easily verified, that (A.5) are identical with the equations from Table 1. We define two diagonal matrices I'_k and I''_k as

$$\begin{aligned} I'_{k,i,j} &= \begin{cases} 1, & i = j \text{ and } \left[\vec{y}_k - \frac{1}{2\varepsilon} (\vec{c} + \tilde{A}^T \vec{\theta}_k) \right]_i > 0, \\ 0, & \text{otherwise,} \end{cases} \\ I''_{k,i,j} &= \begin{cases} 1, & i = j \text{ and } \left[\vec{y}_k - \frac{1}{2\varepsilon} (\vec{c} + \tilde{A}^T \vec{\theta}_k) \right]_i \leq 0, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (\text{A.6})$$

Now we can write $\vec{x}_k = I'_k (\vec{y}_k - (1/2\varepsilon)(\vec{c} + \tilde{A}^T \vec{\theta}_k))$. Then by differentiation of the Lagrangian function (A.4) we get

$$\nabla_y L_k = -2\varepsilon (\vec{x}_k - \vec{y}_k) = I'_k (\vec{c} + \tilde{A}^T \vec{\theta}_k) + I''_k 2\varepsilon \vec{y}_k, \quad (\text{A.7})$$

$$\nabla_{\theta} L_k = \tilde{A} \vec{x}_k - \vec{b}, \quad (\text{A.8})$$

$$\nabla_{yy}^2 L_k = I'_k 2\varepsilon, \quad \nabla_{\theta y}^2 L_k = \tilde{A} I'_k, \quad (\text{A.9})$$

$$\nabla_{y\theta}^2 L_k = I'_k \tilde{A}^T \quad \nabla_{\theta\theta}^2 L_k = -\frac{1}{2\varepsilon} \tilde{A} I'_k \tilde{A}^T.$$

We define the merit function P_k as

$$P_k = 0.5 \left| \nabla_y L_k \right|^2 + 0.5 \left| \nabla_{\theta} L_k \right|^2. \quad (\text{A.10})$$

According to the Karush-Kuhn-Tucker conditions (see e.g., [24, 26]) if $P_k = 0$ the solution in the k th iteration of the algorithm is the optimal solution of the problem (A.1). The gradient of the merit function P_k is:

$$\begin{aligned} \nabla P_k &= \begin{bmatrix} \nabla_{yy}^2 L_k \nabla_y L_k + \nabla_{y\theta}^2 L_k \nabla_{\theta} L_k \\ \nabla_{\theta y}^2 L_k \nabla_y L_k + \nabla_{\theta\theta}^2 L_k \nabla_{\theta} L_k \end{bmatrix} \\ &= \begin{bmatrix} I'_k 2\varepsilon L_k \nabla_y L_k + I'_k \tilde{A}^T (\tilde{A} \vec{x}_k - \vec{b}) \\ \tilde{A} I'_k \nabla_y L_k - \frac{1}{2\varepsilon} \tilde{A} I'_k \tilde{A}^T (\tilde{A} \vec{x}_k - \vec{b}) \end{bmatrix}. \end{aligned} \quad (\text{A.11})$$

We can define a column change vector as $d_k = \alpha [-\nabla_y L_k^T, (\tilde{A} \vec{x}_k - \vec{b})^T]^T$. For $\alpha \rightarrow 0$ we can express one iteration step of the merit function (A.10):

$$\begin{aligned} d_k^T \nabla P_k &= \alpha \left(-\nabla_y L_k^T 2\varepsilon I'_k \nabla_y L_k - \nabla_y L_k^T I'_k \tilde{A}^T (\tilde{A} \vec{x}_k - \vec{b}) \right. \\ &\quad \left. + (\tilde{A} \vec{x}_k - \vec{b})^T \tilde{A} I'_k \nabla_y L_k - \frac{1}{2\varepsilon} (\tilde{A} \vec{x}_k - \vec{b})^T \right. \\ &\quad \left. \cdot \tilde{A} I'_k \tilde{A}^T (\tilde{A} \vec{x}_k - \vec{b}) \right) \\ &= -\alpha \left(\nabla_y L_k^T 2\varepsilon I'_k \nabla_y L_k + \frac{1}{2\varepsilon} (\tilde{A} \vec{x}_k - \vec{b})^T \right. \\ &\quad \left. \cdot \tilde{A} I'_k \tilde{A}^T (\tilde{A} \vec{x}_k - \vec{b}) \right) \\ &\leq 0. \end{aligned} \quad (\text{A.12})$$

From (A.12), it results that $d_k^T \nabla P_k \leq 0$, and then the merit function P_k is nonincreasing during the algorithm. Let us assume that there is a k_0 , such that the merit function P_k is nondecreasing for all $k \geq k_0$ (i.e., $d_k^T \nabla P_k = 0$ for all $k \geq k_0$).

From (A.12), we can write for all $k \geq k_0$:

$$\begin{aligned} I''_k \nabla_y L_k &= \vec{0}, \\ I'_k \tilde{A}^T (\tilde{A} \vec{x}_k - \vec{b}) &= \vec{0}. \end{aligned} \quad (\text{A.13})$$

And it follows that

$$I''_k \vec{y}_k = \vec{0}. \quad (\text{A.14})$$

Let us suppose for some $k \geq k_0$ that $I'_k \neq I'_{k+1}$ and for some index i , it holds:

$$\begin{aligned} i : \left[\vec{y}_k - \frac{1}{2\varepsilon} (\vec{c} + \tilde{A}^T \vec{\theta}_k) \right]_i &> 0, \\ \left[\vec{y}_{k+1} - \frac{1}{2\varepsilon} (\vec{c} + \tilde{A}^T \vec{\theta}_{k+1}) \right]_i &\leq 0. \end{aligned} \quad (\text{A.15})$$

Then we can write from (A.15)

$$\begin{aligned} \left[\vec{y}_{k+1} - \frac{1}{2\varepsilon} (\vec{c} + \tilde{A}^T \vec{\theta}_{k+1}) \right]_i \\ = \left[\vec{y}_k - \frac{1}{2\varepsilon} (\vec{c} + \tilde{A}^T \vec{\theta}_k) - \alpha (\vec{c} + \tilde{A}^T \vec{\theta}_k) \right]_i \leq 0. \end{aligned} \quad (\text{A.16})$$

Under the condition from (A.16) follows

$$\left[\vec{c} + \tilde{A}^T \vec{\theta}_k \right]_i > 0. \quad (\text{A.17})$$

Under the condition $\alpha < 1/(2\varepsilon)$ and (A.17) we can write from (A.15), (A.7), and (A.5)

$$\begin{aligned} 0 < \left[\vec{y}_k - \frac{1}{2\varepsilon} (\vec{c} + \tilde{A}^T \vec{\theta}_k) \right]_i < \left[\vec{y}_k - \alpha (\vec{c} + \tilde{A}^T \vec{\theta}_k) \right]_i \\ = \left[\vec{y}_k - \alpha \nabla_y L_k \right]_i = \vec{y}_{k+1,i}. \end{aligned} \quad (\text{A.18})$$

It means $\vec{y}_{k+1,i} \neq 0$ which is in contradiction with the conditions (A.14) and (A.13). It follows for all $k \geq k_0$ that

$$\left[\vec{c} + \tilde{A}^T \vec{\theta}_k \right]_i \leq \vec{0}. \quad (\text{A.19})$$

And according to (A.7) for all $k \geq k_0$, it holds:

$$\nabla_y L_k \leq \vec{0}. \quad (\text{A.20})$$

From (A.18), (A.14), and (A.15) follows

$$\text{rank} \left| I'_k \right| \leq \text{rank} \left| I'_{k+1} \right|. \quad (\text{A.21})$$

The result of (A.21) is that there are a finite number of changes of the matrices I'_k and I''_k such as $I'_k \neq I'_{k+1}$ and $I''_k \neq I''_{k+1}$ for $k \geq k_0$. So for some $k_1 \geq k_0$, it holds $I'_k = I'_{k+1}$ for all $k \geq k_1$. Based on this fact we can express from (A.5) \vec{x}_{k+1} using (A.13) for all $k \geq k_1$:

$$\begin{aligned} \vec{x}_{k+1} &= I'_{k+1} \left[\vec{y}_{k+1} - \frac{1}{2\varepsilon} \left(\vec{c} + \tilde{A}^T \vec{\theta}_{k+1} \right) \right] \\ &= I'_k \left[\vec{y}_k - \frac{1}{2\varepsilon} \left(\vec{c} + \tilde{A}^T \vec{\theta}_k \right) - \alpha \nabla_y L_k - \frac{\alpha}{2\varepsilon} \tilde{A}^T \left(\tilde{A} \vec{x}_k - \vec{b} \right) \right] \\ &= \vec{x}_k - \alpha \nabla_y L_k. \end{aligned} \quad (\text{A.22})$$

and from (A.13) and (A.22) follows for all $k \geq k_1$

$$\begin{aligned} 0 &= \nabla_y L_k^T \tilde{A}^T \left(\tilde{A} \vec{x}_{k+1} - \vec{b} \right) \\ &= \nabla_y L_k^T \left(\tilde{A}^T \left(\tilde{A} \vec{x}_k - \vec{b} \right) - \alpha \tilde{A}^T \tilde{A} \nabla_y L_k \right) \\ &= \alpha \nabla_y L_k^T \tilde{A}^T \tilde{A} \nabla_y L_k. \end{aligned} \quad (\text{A.23})$$

From (A.23) we get a condition, which says that for all $k \geq k_1$ the change of variable \vec{y}_k is a circulation in the network:

$$\tilde{A} \nabla_y L_k = \vec{0}. \quad (\text{A.24})$$

Based on (A.24) and (A.13) we write:

$$0 \leq \nabla_y L_k^T \nabla_y L_k = \nabla_y L_k^T \left(\vec{c} + \tilde{A}^T \vec{\theta}_k \right) = \nabla_y L_k^T \vec{c} \leq 0. \quad (\text{A.25})$$

And the result of (A.25) for all $k \geq k_1$ is

$$\nabla_y L_k = \vec{0}. \quad (\text{A.26})$$

To express $(\tilde{A} \vec{x}_k - \vec{b})$ we use the problem (A.1). According to the Karush-Kuhn-Tucker conditions and (A.26) we can write a dual function of this problem. Please notice that $\vec{x}_k \nabla_x L_k = 0$ for all $k \geq k_1$ according to (A.26) and (A.7)

$$\begin{aligned} g(\vec{\theta}_k) &= \min_{\vec{x} \geq \vec{0}, \vec{y}} L(\vec{x}, \vec{y}, \vec{\theta}_k) \\ &= \vec{c}^T \vec{x}_{k_1} + \varepsilon (\vec{x}_{k_1} - \vec{y}_{k_1})^T (\vec{x}_{k_1} - \vec{y}_{k_1}) + \vec{\theta}_k^T (\tilde{A} \vec{x}_{k_1} - \vec{b}). \end{aligned} \quad (\text{A.27})$$

Then for $k + 1$ iteration it holds:

$$g(\vec{\theta}_{k+1}) = g(\vec{\theta}_k) + \alpha (\tilde{A} \vec{x}_{k_1} - \vec{b})^T (\tilde{A} \vec{x}_{k_1} - \vec{b}). \quad (\text{A.28})$$

From (A.28) it follows that for $(\tilde{A} \vec{x}_{k_1} - \vec{b}) \neq 0$ it holds: if $k \rightarrow \infty$ then $g(\vec{\theta}_k) \rightarrow \infty$. According to the duality gap theorem (see, e.g., [26]) it holds: $\max_{\vec{\theta}} g(\vec{\theta}) \leq \min_{\vec{x}, \vec{y} \in S} f(\vec{x}, \vec{y})$, where $f(\vec{x}, \vec{y})$ represents the primal function and S a set of feasible solutions. It follows that if $(\tilde{A} \vec{x}_{k_1} - \vec{b}) \neq 0$ and the merit function (A.10) is nondecreasing then the original problem is not feasible. It follows for feasible problems

$$(\tilde{A} \vec{x}_k - \vec{b}) = 0 \quad \forall k \geq k_1. \quad (\text{A.29})$$

We have presented a merit function P_k which is nonincreasing during the algorithm for some $\alpha \rightarrow 0$ and which is equal to zero $P_k = 0$ for an optimal feasible solution. Next, we have shown for feasible problems that if the merit function P_k is nondecreasing for all $k \geq k_1$ then according to (A.26) and (A.29) the merit function $P_k = 0$, and the solution $(\vec{x}_k, \vec{y}_k, \vec{\theta}_k)$ is an optimal solution of the original problem (A.1).

Acknowledgment

This work was supported by the Ministry of Education of the Czech Republic under the Project P103/10/0850 and the Project ME 10039.

References

- [1] A. Koubaa, M. Alves, and E. Tovar, "GTS allocation analysis in IEEE 802.15.4 for real-time wireless sensor networks," in *Proceedings of the 14th International Workshop on Parallel and Distributed Real-Time Systems*, vol. 2006, Rhodes, Greece, 2006.
- [2] M. G. C. Resende and P. M. Pardalos, *Handbook of Optimization in Telecommunications*, Springer, New York, NY, USA, 2006.
- [3] D. P. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, New York, NY, USA, 2004.
- [4] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*, Athena Scientific, Belmont, Mass, USA, 1998.
- [5] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: a stateless protocol for real-time communication in sensor networks," in *Proceedings of the International Conference on Distributed Computing Systems (ICDCS '03)*, pp. 46–55, May 2003.
- [6] O. Chipara, Z. He, G. Xing et al., "Real-time power-aware routing in sensor networks" in *Proceedings of the 14th IEEE International Workshop on Quality of Service (IWQoS '06)*, pp. 83–92, June 2006.
- [7] G. Bravos and A. G. Kanatas, "Integrating power control with routing to satisfy energy and delay constraints in sensor networks," *European Transactions on Telecommunications*, vol. 20, no. 2, pp. 233–245, 2009.
- [8] X. Kai and Y. Zeng, "A distributed cross-layer real-time routing in wireless sensor networks," in *Proceedings of the 2nd International Conference on Signal Processing Systems (ICSPS '10)*, vol. 1, pp. V159–V162, Dalian, China, 2010.

- [9] P. Jurčík, R. Severino, A. Koubáa, M. Alves, and E. Tovar, "Real-time communications over cluster-tree sensor networks with mobile sink behaviour," in *Proceedings of the 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA '08)*, pp. 401–412, Kaohsiung, Taiwan, 2008.
- [10] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, Article ID 1664999, pp. 1439–1451, 2006.
- [11] D. P. Palomar and M. Chiang, "Alternative decompositions for distributed maximization of network utility: framework and applications," in *Proceedings of the 25th IEEE International Conference on Computer Communications*, pp. 1–13, Barcelona, Spain, April 2006.
- [12] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: a mathematical theory of network architectures," *Proceedings of the IEEE*, vol. 95, no. 1, Article ID 4118456, pp. 255–312, 2007.
- [13] B. Johansson, P. Soldati, and M. Johansson, "Mathematical decomposition techniques for distributed cross-layer optimization of data networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1535–1547, 2006.
- [14] B. Johansson and M. Johansson, "Primal and dual approaches to distributed cross-layer optimization," in *Proceedings of the 16th IFAC World Congress*, pp. 113–118, Prague, Czech Republic, 2005.
- [15] H. Nama, M. Chiang, and N. Mandayam, "Utility-lifetime trade-off in self-regulating wireless sensor networks: a cross-layer design approach," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, vol. 8, pp. 3511–3516, June 2006.
- [16] B. Johansson, C. M. Carretti, and M. Johansson, "On distributed optimization using peer-to-peer communications in wireless sensor networks," in *Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '08)*, pp. 497–505, San Francisco, Calif, USA, June 2008.
- [17] J. N. Tsitsiklis and D. P. Bertsekas, "Distributed asynchronous optimal routing in data networks," *IEEE Transactions on Automatic Control*, vol. 31, no. 4, pp. 325–332, 1986.
- [18] S. H. Low and D. E. Lapsley, "Optimization flow control—I: basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.
- [19] W. Hui, Y. Yuhang, M. Maode, and W. Xiaomin, "Network lifetime optimization by duality approach for multi-source and single-sink topology in wireless sensor networks," *IEEE International Conference on Communications*, pp. 3201–3206, 2007.
- [20] H. Wang, Y. Yang, M. Ode, and D. Wu, "Network lifetime optimization by duality approach for single-source and single-sink topology in wireless sensor networks," in *Proceedings of the 4th IEEE and IFIP International Conference on Wireless and Optical Communications Networks (WOCN '07)*, pp. 1–7, Singapore, July 2007.
- [21] M. Zheng, W. Liang, H. Yu, and Y. Xiao, "Cross layer optimization for energy-constrained wireless sensor networks: joint rate control and routing," *Computer Journal*, vol. 53, no. 10, pp. 1632–1642, 2010.
- [22] J. Trdlička and Z. Hanzálek, "Distributed algorithm for energy optimal multi-commodity network flow routing in sensor networks," in *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP '10)*, pp. 1–6, Suzhou, China, October 2010.
- [23] J. Trdlička, Z. Hanzálek, and M. Johansson, "Optimal flow routing in multi-hop sensor networks with real-time constraints through linear programming," in *Proceedings of the IEEE Symposium on Emerging Technologies and Factory Automation (ETFA '07)*, pp. 924–931, September 2007.
- [24] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
- [25] J. Trdlička, "Video presentation," 2011, <http://www.trdlicka.cz/research/distributed-routing>.
- [26] D. P. Bertsekas, *Nonlinear Programming*, Massachusetts Institute of Technology, Boston, Mass, USA, 1999.

Research Article

MAC Protocols Used by Wireless Sensor Networks and a General Method of Performance Evaluation

Joseph Kabara¹ and Maria Calle²

¹ School of Information Sciences, University of Pittsburgh, Pittsburgh, PA 15260, USA

² Department of Electrical and Electronics Engineering, Universidad del Norte, Barranquilla, Colombia

Correspondence should be addressed to Joseph Kabara, jkabara@ieee.org

Received 15 June 2011; Revised 13 September 2011; Accepted 16 September 2011

Academic Editor: Yuhang Yang

Copyright © 2012 J. Kabara and M. Calle. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many researchers employ IEEE802.15.4 as communication technology for wireless sensor networks (WSNs). However, medium access control (MAC) layer requirements for communications in wireless sensor networks (WSNs) vary because the network is usually optimized for specific applications. Thus, one particular standard will hardly be suitable for every possible application. Two general categories of MAC techniques exist: contention based and schedule based. This paper explains these two main approaches and includes examples of each one. The paper concludes with a unique performance analysis and comparison of benefits and limitations of each protocol with respect to WSNs.

1. Introduction

Wireless sensor networks (WSNs) have hundreds or potentially thousands of nodes, each of which are small computers capable of measuring physical characteristic(s) of the surrounding environment and transmitting the information using a radio link. WSNs can be used in monitoring applications such as weather, crops, surveillance, human health care, and structural health [1, 2]. However, WSNs are different from typical computer networks in that individual nodes have very limiting constraints in memory and processing power. Additionally, energy usage is a major limitation since nodes usually employ physically small hardware platforms and they are very likely to be battery powered. Once a battery is depleted, it is often very difficult, if not impossible, to recharge or replace it, so the node is considered dead [1]. One example application is structural health monitoring of a bridge. There may be hundreds of nodes measuring vibrations in the bridge and transmitting this information to a sink (main receiver) not located on the bridge. The information can be used by engineers to schedule maintenance or repairs. When batteries are depleted, nodes must be replaced or recharged. As an illustration, consider the application presented in [3], where a node transmits every

80 milliseconds and the hardware platform uses 120.12 joules in one hour. If the hardware employs two AA batteries with a capacity of 1200 mAh, the node can work 65.96 hours before someone must climb the bridge to replace hundreds of batteries. Another example application is leakage in an industrial plant with hazardous chemicals. People must evacuate, but a sensor network may be deployed by dropping nodes from a plane. In this case there is no control over the network topology and no way to recharge batteries either.

An additional complication is that individual monitoring applications have widely different requirements in throughput, delay, network topology, and so forth. Regarding physical topology, the bridge monitoring and the chemical leak monitoring are applications using nodes possibly located in random positions. In contrast, if the situation is patient monitoring in a medical facility, the network may need a specific layout in order to avoid interference with medical equipment. Regarding delay, human health monitoring may have a tighter delay requirement than the other two mentioned applications since vital signs of the patient may indicate the need of immediate treatment. Since different applications have different requirements, WSNs will employ a family of communication standards, each member designed to optimize the critical parameter(s).

2. Background

Since the terminology for wireless sensor networks is often used with different meanings in the literature, a single, common set of definitions is necessary to prevent confusion.

(i) *MAC Layer.* The IEEE802 LAN (local area network) and MAN (metropolitan area network) Reference Model [4] defines medium access control (MAC) as a sublayer of the data link layer presented in the OSI model. The MAC layer main functions are frame delimiting and recognition, addressing, transfer of data from upper layers, error protection (generally using frame check sequences), and arbitration of access to one channel shared by all nodes [4]. MAC layer protocols for WSNs must be energy efficient to maximize lifetime. Additionally, protocols must be scalable according to the network size and should adapt to changes in the network such as addition of new nodes, death of existing nodes, and transient noise on the wireless channel [5].

(ii) *Sleep.* Node state where the radio is turned off [6].

(iii) *Frame.* Data unit containing information from a MAC layer protocol and possibly from upper layers [4].

(iv) *Packet.* Data unit with information from a network layer protocol and possibly from upper layers [4].

(v) *Collision.* Event where two or more frames are received at the same time, damaging the resulting signal. All information is lost [5].

(vi) *Overhearing.* To receive a packet whose destination is any other node [6]. Overhearing results in wasted energy.

(vii) *Idle Listening.* Another source of wasting energy occurs when a node has its radio on, listening to the medium while there are no transmissions [6].

(viii) *Overemitting.* To transmit a message when the destination is not ready for receiving it. Energy for sending the message is wasted [5].

(ix) *Control Frames Overhead.* All frames containing protocol information and not application data. Energy for transmitting and receiving these frames is considered to be wasted [6].

(x) *Capture Effect.* Phenomenon present in some analog modulation schemes, such as frequency modulation (FM). Two signals with different amplitudes arrive at a receiver and go through the passband filter at the same time. The lower amplitude signal is greatly attenuated at the demodulator output, so the stronger signal is successfully received [7].

(xi) *Broadcast.* Sending a message to all nodes in the network [5].

(xii) *Clock Drift.* Most clocks in networking equipment use quartz oscillators, which change with age, temperature, magnetic fields, and mechanical vibration. As the oscillator changes, the time presented by the clock also changes and this is called clock drift [8].

3. Wireless Standards

Standards for wireless communications exist for different applications: cellular telephony, satellite communications, broadcast radio, local area networks, and so forth. Three well-known standards for wireless data communication have been proposed for use in WSNs, each with certain advantages. However, WSNs do not have widely accepted standard communication protocols in any of the layers in the OSI model sense. The following subsections describe standardized protocols which may match WSN requirements. The protocols provide wireless data transmission with appropriate data rates for a wide range of applications, they can be implemented in battery-powered devices, and they do not require complicated planning and setup. Several commercial products use these wireless standards, which could be an advantage for WSNs in cost and ease of implementation. The purpose of this section is to familiarize the reader with the standards, show their advantages and disadvantages, and discuss their use in WSNs.

3.1. *IEEE802.11.* IEEE802.11 is a family of standards for wireless data communications with definitions for characteristics in the Physical and MAC layers. IEEE802.11b, for example, uses direct sequence spread spectrum (DSSS) with varying modulation schemes to maximize the data rate in a given noise environment. Differential binary phase shift keying (DBPSK) is used for 1 Mbps, differential quadrature phase shift keying (DQPSK) for 2 Mbps, and complementary code keying (CCK) for 5.5 and 11 Mbps [9]. The MAC protocol has two modes [9].

(a) *DCF (Distributed Coordination Function).* Mode with no central device controlling the communication. DCF uses CSMA/CA in any of the following ways.

Carrier sensing: a node senses the medium. If it is idle, the node transmits the data frame. If the medium is busy, the node waits until it becomes idle again, waits for a random time and transmits. Upon frame reception, the receiver node answers with an ACK (acknowledgment) control frame. If a collision occurs, transmitting nodes wait a random time and try again later.

Virtual carrier sensing: a node with a frame to transmit senses the medium. If it is idle, the node sends a control frame called RTS (request to send), which contains the intended receiver address and the time required to send the information (transmission delay). If the destination node agrees to communicate, it will answer with a CTS (clear to send) control frame which also contains the delay. All nodes hearing RTS or CTS should refrain from transmission until the transmission delay has elapsed and the medium is

idle again. The receiver must respond with an ACK for each data frame received.

(b) *PCF (Point Coordination Function)*. A special node, the access point (AP), polls every node and controls the communication process. An AP periodically broadcasts a beacon control frame with parameters and invitations to join the network [9].

Advantages of IEEE802.11 include that it is widely used, so it is easy to find networks supporting the standard. Data rates are high for wireless end user transmission and radio ranges can be hundreds of meters. Also, as IEEE802.11 supports well-known protocols as TCP and IP, devices connected with this technology may have easy access to the Internet and this way they can send information anywhere in the world.

Disadvantages include the large overhead in control and data packets. 802.11 requires 34 bytes for the header and the checksum, TCP and IP require a minimum of 20 bytes for each header, so there is at least 74 bytes of overhead to send application information, which in WSNs may be only two bytes. Another possibility is using UDP which employs less overhead, 8 bytes for the header. However, UDP uses IP and 802.11 MAC headers add 62 bytes total to the application information. Perhaps the most important problem for using 802.11 in WSNs is energy consumption. Even though the standard has power saving mechanisms, according to Ferrari et al. "power consumption is rather high, and the short autonomy of a battery supply still remains the main disadvantage of the proposed IEEE802.11 sensor system" [10].

3.2. IEEE802.15.1, Bluetooth. The IEEE also defined MAC and physical layer characteristics for the 802.15.1 standard. In this standard, the physical layer uses 2.4 GHz, frequency hopping spread spectrum (FHSS) with Gaussian frequency shift keying (GFSK) as the modulation scheme. The result is a 1 Mbps data in the basic rate; however, much of the capacity is used for control purposes. The enhanced data rate provision has two data rates, 2 Mbps using $\pi/4$ -differential quadrature phase shift keying (DQPSK) and 3 Mbps using 8 DPSK [11]. IEEE802.15 defines wireless personal area networks (WPANs) allowing connectivity in a 10-meter range. However, some Bluetooth devices have 100-meter range [12].

An 802.15.1 master node controls up to 7 active slave and up to 255 nonactive slave nodes. These networks are referred to as piconets and several piconets may communicate using a bridge node, forming a scatternet. The MAC protocol uses polling with a time division multiplexing (TDM) scheme called time division duplex. In one time slot, the master will poll a single slave, inquiring if it has something to send. If the slave has data to transmit, it sends it to the master in the next time slot [13]. A master node must periodically transmit, even if there is no data to be exchanged, to keep slaves synchronized. Slaves cannot communicate directly; the information must go through the master node. Using the most reliable communication mode, a Piconet can support

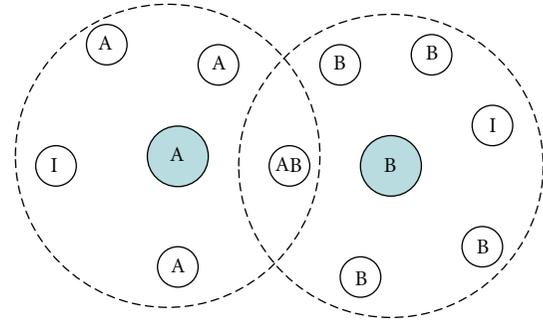


FIGURE 1: Bluetooth scatternet. Shaded circles are master nodes in two different piconets, A and B. White circles show slaves belonging to each piconet. Active slaves are labeled according to the network they belong to; inactive slaves have label I. AB is a slave belonging to both piconets. The figure shows coverage areas of both master nodes.

one full duplex channel with 64 kbps master-slave and another 64 kbps slave-master through the basic rate [13]. Figure 1 illustrates an example scatternet [14].

An advantage to using 802.15.1 for WSNs is that the hardware is designed to have low cost [11]. Disadvantages of 802.15.1 include that a WSN using Bluetooth requires that a group of nodes transmit to one master, located just one hop away. WSNs literature calls this organization cluster based, and the master node is referred to as cluster head [15]. Research shows that a problem in this approach is the master/cluster head becomes a single point of failure, which can isolate all other members of the network [16]. Another problem arises in applications with random deployment because it is not always possible to ensure that all slave nodes are within range of the master. Additionally, the periodic transmissions used for synchronization waste energy at both the transmitter and the receivers.

3.3. IEEE802.15.4. The IEEE defined physical and MAC layer characteristics for establishing connectivity between devices with low-power consumption, low cost, and low data rate. The standard is related to ZigBee technology since The ZigBee Alliance (association of several companies such as Samsung and Motorola) defines the other communication layers (above MAC) for 802.15.4 compliant devices. Frequency bands are 2.4 GHz and 868/915 MHz, both working with DSSS. The 2.4 GHz band has a 250 kbps data rate using offset quadrature phase shift keying (O-QPSK) modulation. The 868/915 MHz band has data rates up to 240 kbps using BPSK [17]. Typical radio range according to the standard is 10 meters. Maximum packet size is 128 bytes with payload of 104 bytes. 64-bit IEEE or 16-bit addresses can be used [17]. The 802.15.4 standard defines two types of devices.

FFD (Full Function Device): Supports all characteristics from the standard. One FFD can be a network coordinator, a router, or a gateway which connects the network to other networks. FFDs can communicate with any other device [17].

RFD (Reduced Function Device): It has very limited characteristics and it can only talk to a FFD [17]. RFDs have low-power consumption and low complexity.

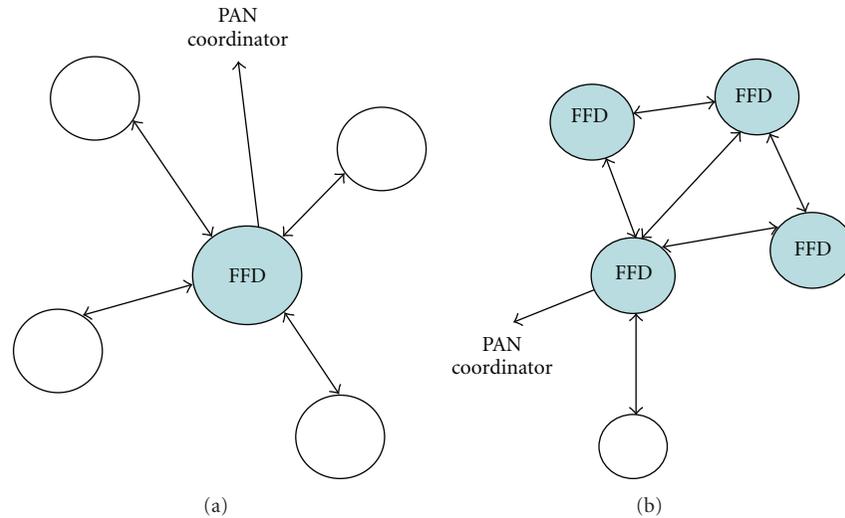


FIGURE 2: (a) Star topology with one full function device and 4 reduced function devices (white circles). (b) Peer-to-Peer topology.

Figure 2 presents two possible topologies using 802.15.4. Both topologies have a PAN (personal area network) coordinator, which is a FFD [17].

The 802.15.4 MAC layer has two modes [17]:

Nonbeacon mode employs CSMA/CA. A node checks the medium; if busy, it waits for a random period of time before trying to transmit. If idle, the node transmits.

Beacon mode employs two periods: active (divided in 16 time slots) and inactive (devices enter a low-power mode), as presented in Figure 3 [17]. At the beginning of the active period, the coordinator sends beacon frames with information regarding the period duration so the duty cycle can vary. The contention access period (CAP) follows the Beacon, allowing devices to send frames using slotted CSMA/CA. A node waits for a random time, then checks the medium and if the channel is clear, transmits. If the channel is busy, the device waits again. The first waiting time (before checking the medium) can be very small to minimize idle listening in low traffic. The node can sleep immediately after receiving an acknowledgement.

When the CAP ends, the collision free period (CFP) begins. The CFP uses guaranteed time slots (GTSs), in a TDMA fashion, to support devices requiring low latency or dedicated bandwidth. The coordinator cannot interact with the PAN during the inactive period and may sleep [17].

The major advantage to using 802.15.4 for WSNs is that the hardware for the nodes is designed to be inexpensive [18]. Disadvantages to using 802.15.4 for WSNs include that a star topology is only appropriate for the clustered model of WSNs since this model requires all RFDs to be close enough for their signal to be received by a FFD. However, like 802.15.1, a random deployment does not guarantee the position of any device. 802.15.4 energy usage may be another issue; the standard is designed to minimize usage but still some ZigBee radio devices consume more energy than devices using just FSK modulation and Manchester encoding [19]. In one example, the MICAz radio works at 250 kbps (802.15.4 compliant radio), requiring 19.7 mA for receiving

and 17 mA for transmitting with 1 mW transmission power. MICA2 uses the same microcontroller and memory but the radio works at 38.4 kbps, requiring 7 mA for receiving and 10 mA for transmitting with the same power as MICAz [19]. Transmitting a fixed size packet requires more energy in MICA2 than MICAz. However, if both platforms spend the same time in idle listening (receiving nothing), MICAz uses more energy than MICA2.

3.4. WirelessHART. The HART Communication Foundation extended the wired HART protocol for communication requirements in industrial plants, specifically compensating for electrically noisy environments and real-time delay constraints. WirelessHART was accepted as international electrotechnical commission (IEC) standard 62591. The protocol defines functionality in the physical, MAC, network, transport and application layers [20]. WirelessHART uses the physical layer from IEEE802.15.4, but the MAC layer uses TDMA and channel hopping, in order to minimize interference [21]. A blacklist feature blocks occupied channels, so hopping can take place at most in 16 different frequencies. Time division multiplexing employs 10 msec time slots which can be allocated to a single transmitter and receiver pair, or several devices in a contention access method similar to CSMA. A group of time slots is called a superframe with size defined by a network manager device, which also maintains synchronization and creates routes in the network. WirelessHART creates a mesh network with six different types of devices, as shown in Figure 4 [22].

NM (network Manager) It creates and manages the TDMA schedule and routes in the network.

FD (Field Device) communication devices directly connected to the monitored machines.

RD (Field Device) router devices not directly connected to the monitored machines. Router devices assist in the communication process and may be used to increase network coverage. RDs are optional in the standard.

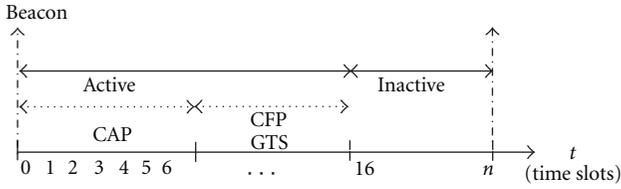


FIGURE 3: Beacon mode structure. Active period has contention (CAP) and noncontention (CFP) modes.

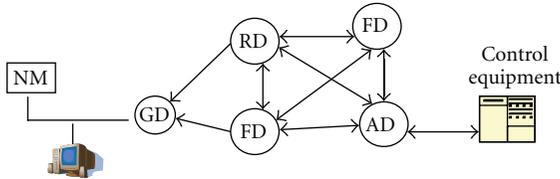


FIGURE 4: Different devices in a WirelessHART network.

AD (Adapter Device) machines with an integral wired HART protocol adapter can connect to the wireless network using an AD.

HD (Handheld Device) allows for mobile monitoring, configuration, and preservation of nodes in a WirelessHART network.

GD (Gateway Device) a gateway connects a WirelessHART compatible network to a network with a different technology, employed in the industrial plant. The network is often the plant automation system.

One advantage of WirelessHART is robustness to harsh industrial communication environment. The main disadvantage is that the NM is a single point of failure; although one backup NM can be implemented, only one active NM is allowed in the network [22].

3.5. ISA100. ISA100 is a family of standards for wireless communications created by the International Society of Automation (ISA) [23]. ISA100a corresponds to process automation, and it shares several features with WirelessHART. Both standards use the physical layer from IEEE802.15.4, and a MAC layer with TDMA, frequency hopping, CSMA, and channel blacklisting [23]. The majority of differences between ISA100 and WirelessHART are in network, transport, and application layers. However, one distinct feature of the ISA100 MAC layer is that it allows dedicated time slots or shared time slots. When sharing a time slot a CSMA-CA algorithm employing priorities is used to control access. A network using ISA100a requires a data link (DL) subnet, with input/output devices, routing and portable devices. There is a backbone network (BN) with routers and gateways (GW) and, finally, a manager network (MN) with a security manager and a system manager [24].

One advantage of ISA100a is that it allows direct connection with different industrial wired standards, such as HART, Fieldbus, and Profibus. One disadvantage is that it requires two manager nodes to control the network, increasing system complexity.

4. Categorization of MAC Protocols for Wireless Sensor Networks

MAC protocols presented in the literature can be classified in two groups according to the approach used to manage medium access: contention based and schedule based [25]. All protocols presented in this paper assume no mobility in the network, only one radio available in each sensor and bidirectional links (meaning if node A can listen to node B, node B can listen to node A).

4.1. Contention Based. Medium access is distributed; there is no need for central coordination for the nodes to use the medium. Examples include the following.

(a) Sensor MAC (S-MAC). S-MAC [6] operates by placing a node in a state that listens to the medium; if a node hears nothing it sends a SYNC packet with a schedule defining listen and sleep periods. All nodes hearing this packet will adopt the schedule. Nodes may adopt two or more schedules (if different neighbors have different schedules). Nodes keep tables with the schedules of their neighbors. During a listen period, a node with a packet to send executes a procedure similar to 802.11 virtual channel sensing, it will send a request to send (RTS) frame and the receiver node will answer with a clear to send (CTS) frame. All nodes not involved in the conversation will enter a sleep state while the communicating nodes send data packets and ACKs. Sleeping decreases energy consumption but introduces latency since communication with a sleeping node must wait until it wakes up [6]. Figure 5 shows an example of the sequence of events occurring in communication between four nodes using S-MAC.

Advantages of S-MAC include sleeping, which reduces energy consumption. The protocol adapts easily to changes in topology and has been tested in hardware. Additionally, there is no need for a central entity or for tight synchronization. Disadvantages of S-MAC include the need to maintain loose synchronization for the schedules to work properly. Clock drift in the nodes can result in nodes becoming unsynchronized. Control frames such as RTS and CTS generate overhead and increase energy usage. Idle Listening still occurs, as shown in Figure 5, where node D is not receiving any packet but must stay awake during the entire listening phase.

S-MAC has been extensively studied and several subsequent protocols include suggestions for performance improvement. Examples include timeout MAC (T-MAC) [26] and dynamic sensor-MAC (DS-MAC) [27]. The B-MAC protocol suggests a different approach which decreases the overhead generated by control frames and does not explicitly synchronize the transmitter and the receiver.

(b) Berkeley Media Access Control for Low-Power Sensor Networks (B-MAC). B-MAC [28] employs an adaptive preamble to reduce idle listening, a major source of energy usage in many protocols. When a node has a packet to send, it waits during a backoff time before checking the channel. If

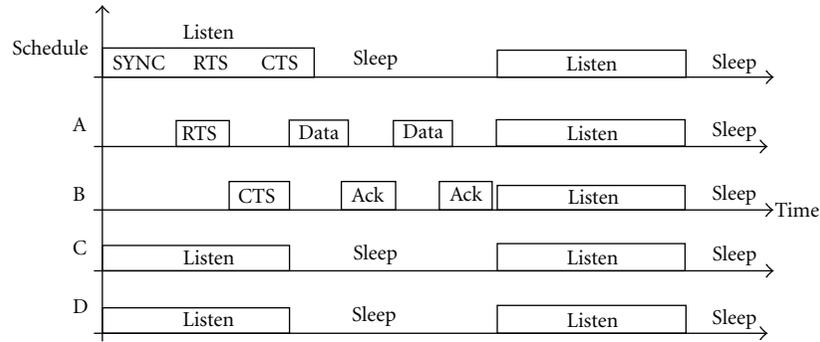


FIGURE 5: S-MAC example. Nodes A, B, and C are within range of each other. D is within range of C and A transmits to B.

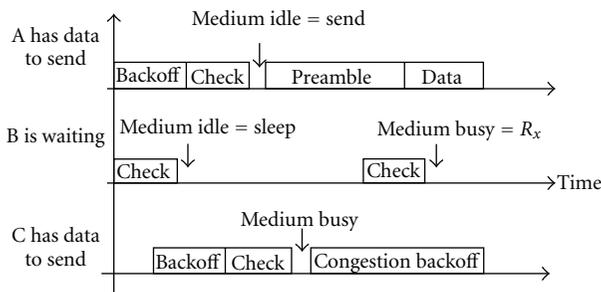


FIGURE 6: B-MAC communication example. All nodes are within range of each other.

the channel is clear, the node transmits; otherwise it begins a second (congestion) backoff. Each node must check the channel periodically using LPL (low-power listening); if the channel is idle and the node has no data to transmit, the node returns to sleep [28]. Figure 6 illustrates one example transmission using B-MAC.

The B-MAC preamble sampling scheme adjusts the interval in which the channel is checked to equal the frame preamble size. As an example, if the medium is checked every 100 ms, the preamble of the packet must last 100 ms as a minimum, in order for the receiver to detect the packet. Upper layers may change the preamble duration, according to the application requirements [28].

An advantage of using B-MAC in WSNs is that it does not use RTS, CTS, ACK, or any other control frame by default, but they can be added. Additionally, it is one of the few specialized MAC protocols whose implementation was tested in hardware. No synchronization is required, and the protocol performance can be tuned by higher layers to meet the needs of various applications. The main disadvantage is that the preamble creates large overhead. One example presents 271 bytes of preamble to send 36 bytes of data [28].

(c) *Predictive Wake-UP MAC (PW-MAC)*. PW-MAC [29] improves on protocols like S-MAC and B-MAC because it uses pseudo random schedules, thus not all nodes will wake up and transmit at the same time, avoiding collisions. A node that has just woke up sends a short beacon so other nodes know it is up. A sender can then transmit a data packet and

request more information from the receiver, such as current time and current seed for the pseudo random schedule used by receiver. By using the seed in a linear congruential generator (LCG), sender in PW-MAC can predict when a receiver will wake up; hence sender sleeps until a little bit before the receiver is awake.

However, there are hardware variations that generate errors in the sender prediction. PW-MAC uses a “sender wake-up advance time” [29], a compensating value particular to every platform, including clock drift, operating system delay, and hardware latency. The value helps correcting errors each node can do when predicting a receiver wake-up time.

One advantage of using PW-MAC is that sleeping until the receiver is up effectively decreases duty cycle in the sender. Additionally, the protocol has been tested on hardware, using MicaZ motes, and memory footprint is small.

Disadvantages of using PW-MAC include overhead created by beacons and idle listening, even if it is small [29] compared to other protocols such as WiseMAC [30], RI-MAC [31], and X-MAC [32].

4.2. *Schedule Based*. Protocols arbitrate medium access by defining an order (called schedule) for nodes to transmit, receive, or be inactive. Generally speaking, each node communicates during specific time slot(s) and can be inactive the rest of the time. Schedule-based protocols use a variety of approaches, as illustrated below.

(a) *Low-Energy Adaptive Clustering Hierarchy (LEACH)*. LEACH [33] includes application, routing, MAC, and physical characteristics for communication in WSNs. A specific application considered is remote monitoring where data gathered by neighboring nodes may be redundant. LEACH assumes all nodes are synchronized, they can control their transmission power, and they can reach one base station (BS, equivalent to the sink in other protocols) if needed. The nodes also have sufficient processing capabilities to implement different MAC protocols and perform signal processing functions, such that all information can be aggregated in only one message. The LEACH protocol works in rounds, as presented in Figure 7. Nodes organize in clusters, elect a cluster head (CH), and then start sending

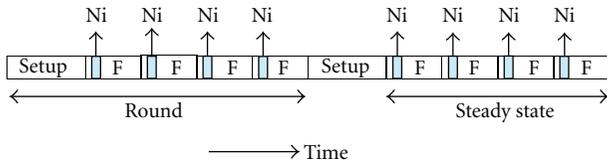


FIGURE 7: LEACH operation rounds. F are frames divided in time slots. Ni are slots assigned to node *i*.

information. Every cluster uses DSSS with a different code, to minimize interference [33].

During the setup phase, nonpersistent CSMA is used as the MAC protocol. Node *i* elects itself as a CH with probability $P_i(t)$. The probability is selected in such a way that every node can be a CH and those who have recently been elected have a smaller chance to be selected in the next round. Each elected node sends an advertisement message. Nonelected nodes receive several of these messages and decide which cluster to join, based on the received signal strength of the messages. The nodes inform the CH using a join-request message. The CH creates a TDMA schedule using this information and sends it to all nodes in the cluster. In normal, steady-state operation, every node uses only its assigned time slot to send data to the CH and sleeps the rest of the time. Cluster heads aggregate their cluster data and send it to the BS using CSMA [33].

Advantages of LEACH include saving energy through sleeping. CH rotation extends the lifetime of the network by balancing the rate of energy usage over all nodes, so any one node takes longer to exhaust its energy resources. Including several other networking layers in the protocol design benefits the whole communication scheme by reducing energy usage due to inefficiencies between layers. Disadvantages of LEACH include overhead associated with the death of a CH. When a CH dies, the whole cluster becomes inactive during the remaining steady-state phase, even if several nodes inside the cluster have enough energy to function. Also, LEACH assumes one-hop communication between the nodes and the CH and also among the cluster heads and the BS, something that is not easily achieved in a randomly deployed network. DSSS increases the complexity of the hardware. LEACH requires tight synchronization (for the TDMA schedule and for using DSSS) which is not included as part of the protocol and will require additional energy and overhead to accomplish.

(b) *Power-Efficient and Delay-Aware Medium Access Protocol (PEDAMACS)*. PEDAMACS [34] assumes one access point (AP, also called sink) with the ability to reach all sensor nodes in one hop. However, sensor nodes may employ more than one hop to reach the AP. There are three transmission power levels defined to reach three distances: P_l the maximum, P_m the medium, and P_s the minimum. The protocol has the following four phases, which are illustrated in Figure 8.

Topology learning: the AP broadcasts a packet with P_l to synchronize the nodes. After that, the AP sends another packet with P_m which will be retransmitted through the entire network, so all nodes receive the topology currently

held by the AP and can update it. Using the received signal strength and interference models, each node identifies its local neighbors (nodes able to decode a packet transmitted with P_s), its interferers (nodes unable to decode a packet transmitted with P_s , but with received signal strength high enough to interfere with other signals), and its parent node in the route to the AP. During this phase, the protocol employs a protocol similar to 802.11, with RTS and CTS, since there is no schedule yet.

Topology collection: each node sends topology information to the AP using P_s , so data may possibly go through several hops. The protocol also uses CSMA in this phase.

Scheduling phase: the AP broadcasts the schedule so every node adjusts its clock and knows the time slots allowed for it to transmit and receive. The rest of the time, the nodes sleep. A guard interval for each time slot compensates for synchronization errors. Nodes transmit data with P_s .

Adjustment: at the end of the scheduling phase, the AP requests and the nodes send adjustment topology packets indicating changes in neighbors or interferers. Nodes can also send this information during the scheduling phase inside data packets [34].

PEDAMACS considers characteristics from physical and network layers, to its advantage. Other advantages include that PEDAMACS can be used for sending periodic data or for event-driven sensing, using the assigned time slots only when the event happens; otherwise, the nodes keep on sleeping. The protocol can be extended to use more than one AP and to handle nodes outside the range of the AP. Delay results are bounded for different network sizes [34].

The disadvantages of PEDAMACS include considerable additional overhead beside RTS, CTS, and ACK packets. The protocol assumes an AP which can communicate to all nodes, with an infinite energy supply. Such an AP may not be possible in WSNs, especially with random deployment. Additionally, low transmission power levels save energy, but radio ranges decrease significantly. One example with Mica2 motes shows 25 cm radio range for -20 dBm which is the minimum transmission power [3], so nodes must be very close to each other to maintain connectivity in the network.

(c) *Priority-Based MAC Protocol for Wireless Sensor Networks (PRIMA)*. PRIMA [35] uses a similar procedure as LEACH [33] to create clusters and elect cluster heads (CHs) and to control communication and keep synchronization inside each cluster; CH will rotate every 15 minutes. PRIMA defines four priorities for information by making application layer to add two bits at the end of each packet. MAC layer uses two different protocols: classifier MAC (C-MAC) adds each packet to one of four different queues, according to each priority. The other protocol is channel access MAC (CA-MAC) which uses CSMA/CA and TDMA slots. Random access slots allow for different nodes to request a time slot and CH to broadcast schedules. Nodes send data according to schedule using TDMA slots without collisions. A similar situation happens when CHs want to transmit to the base station (main node, BS). There will be a CSMA phase to create schedules and a TDMA phase where each CH can transfer data without collisions.

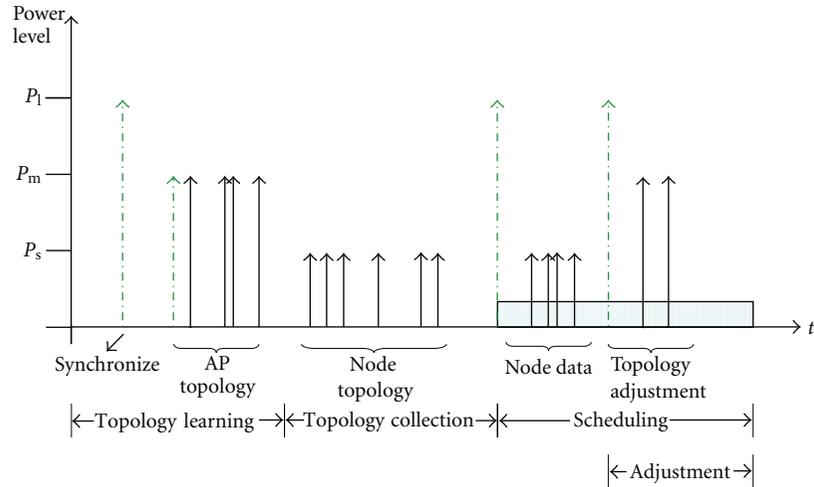


FIGURE 8: PEDAMACS phases: solid arrows are packets sent by nodes; dashed arrows show packets sent by the AP. Different power levels are used in different situations. Only the scheduling phase has defined time slots.

TABLE 1: WSNs MAC Protocol Comparison.

Name	Implemented	Applications	Synch. requirement	Overhead
S-MAC	Hardware	Event-driven, long idle periods, delay order of message time	Loose	RTS, CTS, ACK, SYNC
B-MAC	Simulation/hardware	Delay tolerant	None	Preamble
PW-MAC	Hardware	Low delay, long idle periods	None	Beacon
LEACH	Simulation	Periodic data collection and monitoring	Tight	ADV, Join-Req, schedule
Pedamacs	Simulation	Delay bounded	Tight	RTS, CTS, ACK, Synch, topology learning
PRIMA	Simulation	Different QoS	Tight	Synch, Schedule, CH election
IEEE 802.11	Simulation/hardware	High data rates, large energy source, smart terminals	None	RTS, CTS, ACK
IEEE 802.15.1	Simulation/hardware	Medium to low data rates, low-energy consumption	Tight	Synch transmissions, S, C
IEEE 802.15.4	Simulation/hardware	Medium to low data rates, low-energy consumption	Tight	Beacon, ACK
WirelessHART	Simulation/hardware	Process automation	Tight	Synch, schedule, routing, other
ISA100a	Simulation/hardware	Process automation	Tight	Synch, schedule, routing,

The main advantage of PRIMA is reducing packet delivery delay according to traffic requirements. PRIMA also shares with LEACH advantages in CH rotation, helping increase lifetime. However, if a CH dies, all nodes in the cluster become useless until a new CH election takes place, just as in LEACH. Additionally, overhead packets increase energy consumption.

5. MAC Protocol Summary

Table 1 summarizes the protocols presented in this paper, comparing some of their characteristics. Notice all contention-based protocols have been implemented in hardware, at least for tests shown in the particular cited study, while schedule-based ones have been implemented only in

simulations. Also notably, only PEDAMACS shows bounded delay for different network sizes.

The Applications column in Table 1 shows characteristics of applications that could benefit from the particular protocol. The Overhead column presents the type of control frames or other type of overhead used by each protocol. One example with no control frames is B-MAC where overhead is caused by the preamble size of the data frame. Regarding standards, control frames mentioned in the table are not the only ones used in each case: 802.15.1 uses supervisory (S) and control (C) frames, 802.11 uses control and management frames, and 802.15.4 has command frames. A detailed explanation of all control frames is in the standards presented in [9, 13, 17, 36, 37]. When using the Overhead column for comparison purposes, note each protocol has

TABLE 2: Performance comparison. Protocols in bold are the main subject in each reference mentioned. Others are the benchmarks considered in each case.

Protocol	Maximum energy consumption		Performance metric		Maximum latency	
	Value	Units	Comparison performed using	Platform/Tool	Value	Units
S-MAC [6]	6	Joules	Hardware	Mica	11	Seconds
S-MAC no sleep	29	Joules	Hardware	Mica	1	Second
B-MAC [28]	15	Milliwatts	Hardware	Mica2	1700	Milliseconds
S-MAC	35	Milliwatts	Hardware	Mica2	2700	Milliseconds
PW-MAC [29]	10	%duty cycle	Hardware	MicaZ	1	Second
WiseMAC	70	%duty cycle	Hardware	MicaZ	85	Second
RI-MAC	65	%duty cycle	Hardware	MicaZ	1	Second
X-MAC	70	%duty cycle	Hardware	MicaZ	77	Second
PEDAMACS [34]	13	Millijoules	Simulation	TOSSIM	0.2×10^6	Bit time
S-MAC	21	Millijoules	Simulation	TOSSIM	2.8×10^6	Bit time
IEEE802.11	19.5	Millijoules	Simulation	TOSSIM	0.45×10^6	Bit time
PRIMA-RT [35]	0.015	J/packet/node	Simulation	OMNeT++	15	Seconds
Q-MAC-RT	0.024	J/packet/node	Simulation	OMNeT++	5	Seconds

different control frame sizes and they are sent during different phases of communication, so the total overhead for a particular communication session varies and must be analyzed with respect to a particular application. However, as an illustration, consider a network with four nodes all within range of each other and only one node needs to send one packet. The communication procedure using S-MAC in that network is as follows: one node sends a SYNC frame, all nodes hear it and adopt the schedule and the node with a packet sends an RTS. The receiver answers with CTS, the data packet is transmitted, and the receiver sends one ACK. There are four control frames to send one packet. Now consider the same network using LEACH. One node sends an advertisement message (ADV) saying it is the cluster head, the other three nodes send join request messages, and the CH sends the TDMA schedule. After that, the node with data to transmit sends the packet. Total overhead in this case is five packets. So, even though Overhead column shows three types of control frames for LEACH and four types for S-MAC, the total overhead generated by each protocol may be smaller or larger depending upon the application and current state of the network.

Every protocol tries to improve on a particular metric, thus different performance variables are used to evaluate protocol usefulness. Table 2 shows detailed results presented in the papers as examples of the benefits of using each protocol. The Protocol column shows the main protocol presented in every study using bold characters and the protocol used as a benchmark in each paper with regular characters. The Maximum Energy Consumption column in Table 2 presents the highest value reported for each protocol. Not all papers used energy measurement units, so this column shows data for energy, power, or current consumption for comparison purposes, since the metrics are related. The Platform/Tool column shows the specific hardware or software used in

the experiments of each protocol, since not all protocols were tested using the same procedures. The Maximum Latency column illustrates the highest delay presented for each protocol. Tests are performed with different network sizes, topologies, and energy consumption models in each paper, making it difficult to directly compare protocols. Not all tests use the same units.

Results in Table 2 illustrate the performance comparison presented in each study; in all cases the main protocol has better performance than the protocols employed for comparison purposes. Note that PRIMA-RT means the real-time version of the protocol.

One of the protocols presented in Section 4 is not presented in Table 2, evaluation of the LEACH protocol employed the ns software package, but not for energy consumption or delay.

6. Conclusions

Previously there were no standard methods of comparing the performance of scheduled-based and contention-based protocols, or even for protocols belonging to the same category. The lack of standard evaluation metrics has made it difficult to evaluate and select a protocol, even if the requirements of a particular application are known. The number of wireless sensor network protocols is rapidly expanding so a set of protocols covering the widest possible breadth was selected for analysis. Using the analysis method and metrics presented in this paper suggests that contention-based approaches may be helpful when the network topology is random, application requirements are not delay constrained, and there is no mechanism to ensure tight synchronization. Analysis also shows that schedule-based approaches may be more energy efficient if deployment is not random and the base stations

include high-power transmitters and large energy stores which can be used to manage synchronization and schedules.

Protocol designers and users benefit from standard test methods that can be applied across all communication protocols for WSN, so that protocols can be measured using the same references and units, allowing for comparison and evaluation.

References

- [1] I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [2] X. Ning, R. Sumit, C. Krishna Kant et al., "A wireless sensor network for structural monitoring," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 13–24, ACM Press, November 2004.
- [3] M. Calle and J. Kabara, *Energy Consumption in Wireless Sensor Networks: Measuring Energy Consumption and Lifetime*, VDM Verlag, Saarbrücken, Germany, 2008.
- [4] IEEE, "IEEE Standards for local and metropolitan area networks: overview and architecture," *IEEE Std 802-2001 (Revision of IEEE Std 802-1990)*, 2001.
- [5] I. Demirkol, C. Ersoy, and F. Alagöz, "MAC protocols for wireless sensor networks: a survey," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 115–121, 2006.
- [6] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, 2004.
- [7] K. Leentvaar and J. H. Flint, "The Capture Effect in FM Receivers," *IEEE Transactions on Communications*, vol. 24, no. 5, pp. 531–539, 1976.
- [8] R. Tjoa, K. L. Chee, P. K. Sivaprasad, S. V. Rao, and J. G. Lim, "Clock drift reduction for relative time slot TDMA-based sensor networks," in *Proceedings of the 15th Personal, Indoor and Mobile Radio Communications, (PIMRC '04)*, pp. 1042–1047, September 2004.
- [9] IEEE, "IEEE standard for information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part II: wireless LAN medium access control (MAC) and physical layer (PHY) specifications IEEE Std 802.11g," pp. 67, 2003.
- [10] P. Ferrari, A. Flammini, D. Marioli, and A. Taroni, "IEEE802.11 sensor networking," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 2, pp. 615–619, 2006.
- [11] Bluetooth-SIG, *Specification of the Bluetooth System. Wireless Connections Made easy. Covered Core Package Version: 2.0+EDR*, 2004.
- [12] S. Rathi, "Bluetooth protocol architecture," *Dedicated Systems Magazine*, pp. 28–33, 2000.
- [13] IEEE, "IEEE Std 802.15.1—2005 IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)," *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)*, pp. 0_1-580, 2005.
- [14] A. Tanenbaum, *Computer Networks*, Prentice Hall, Upper Saddle River, NJ, USA, 4th edition, 2003.
- [15] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, 2000.
- [16] T.-Y. Lin, Y.-C. Tseng, K.-M. Chang, and C.-L. Tu, "Formation, routing, and maintenance protocols for the blueRing scatter-net of bluetooths," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, p. 10, 2003.
- [17] IEEE, "IEEE standard for information technology—telecommunications and information exchange between systems—local and metropolitan area networks specific requirements part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)," *IEEE Std 802.15.4-2003*, pp. 0_1-670, 2003.
- [18] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks," in *Proceedings of the 23rd IEEE International Performance, Computing, and Communications Conference (IPCCC '04)*, pp. 701–706, April 2004.
- [19] Crossbow, *MPR- Mote Processor Radio Board MIB- Mote Interface/Programming Board User's Manual*, Crossbow Technology, 2006.
- [20] J. Song, S. Han, A. K. Mok et al., "WirelessHART: applying wireless technology in real-time industrial process control," in *Proceedings of the 14th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '08)*, pp. 377–386, April 2008.
- [21] M. D. Biasi, C. Snickars, K. Landernäs, and A. J. Isaksson, "Simulation of process control with wirelessHART networks subject to packet losses," in *Proceedings of the 4th IEEE Conference on Automation Science and Engineering (CASE '08)*, pp. 548–553, August 2008.
- [22] T. Lennvall, S. Svensson, and F. Hekland, "A comparison of WirelessHART and ZigBee for industrial applications," in *Proceedings of the 7th IEEE International Workshop on Factory Communication Systems (WFCS '08)*, pp. 85–88, May 2008.
- [23] H. Hayashi, T. Hasegawa, and K. Demachi, "Wireless technology for process automation," in *Proceedings of the ICROS-SICE International Joint Conference*, pp. 4591–4594, Tokyo, Japan, August 2009.
- [24] N. Q. Dinh, S.-W. Kim, and D.-S. Kim, "Performance evaluation of priority CSMA-CA mechanism on ISA100.11a wireless network," in *Proceedings of the 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT '10)*, pp. 991–996, 2010.
- [25] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," *Wireless Networks*, vol. 12, no. 1, pp. 63–78, 2006.
- [26] T. V. Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, ACM Press, pp. 171–180, 2003.
- [27] P. Lin, C. Qiao, and X. Wang, "Medium access control with a dynamic duty cycle for sensor networks," *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '04)*, vol. 3, pp. 1534–1539, 2004.
- [28] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 95–107, ACM Press, November 2004.
- [29] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "PW-MAC: an energy-efficient predictive-wakeup MAC protocol

- for wireless sensor networks,” *Proceedings of the IEEE INFOCOM*, pp. 1305–1313, 2011.
- [30] A. El-Hoiydi and J. -D. Decotignie, “WiseMAC: an ultra low power MAC protocol for multi-hop wireless sensor networks,” in *Proceedings of 1st International Workshop, Algorithmic Aspects of Wireless Sensor Networks*, vol. 3121, pp. 18–31, 2004.
- [31] Y. Sun, O. Gurewitz, and D. B. Johnson, “RI-MAC: a receiver initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks,” in *Proceedings of the International Conference on Embedded Networked Sensor System (SenSys '08)*, 2008.
- [32] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks,” in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 307–320, November 2006.
- [33] 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.
- [34] S. C. Ergen and P. Varaiya, “PEDAMACS: power efficient and delay aware medium access protocol for sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 7, Article ID 1637439, pp. 920–930, 2006.
- [35] J. Ben-Othman, L. Mokdad, and B. Yahya, “An energy efficient priority-based QoS MAC protocol for wireless sensor networks,” in *Proceedings of the IEEE International Conference on Communications*, pp. 1–6, 2011.
- [36] H. C. Foundation, *Wireless Devices Specification, HCF SPEC 290 Revision 1.0*, 2007.
- [37] Isa 100a. W. Group, *Wireless Systems for Industrial Automation: Process Control and Related Applications*, 2009.

Research Article

Information Fusion-Based Storage and Retrieve Algorithms for WSNs in Disaster Scenarios

Zhe Xiao,¹ Ming Huang,² Jihong Shi,² Wenwei Niu,² and Jingjing Yang²

¹ School of Electrical and Electronic Engineering, Nanyang Technological University, Western Catchment Area, Singapore 639798

² School of Information Science and Engineering, Yunnan University, Kunming 650091, China

Correspondence should be addressed to Zhe Xiao, zxiao1@e.ntu.edu.sg

Received 12 July 2011; Revised 13 September 2011; Accepted 19 September 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Zhe Xiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sensor networks are especially useful in catastrophic or disaster scenarios such as abysmal sea, floods, fires, or earthquakes where human participation may be too dangerous. Storage technologies take a critical position for WSNs in such scenarios since the sensor nodes may themselves fail unpredictably, resulting in the loss of valuable data. This paper focuses on fountain code-based data storage and recovery solutions for WSNs in disaster scenarios. A review on current technologies is given on challenges posed by disaster environments. Two information fusion-based distributed storage (IFDS) algorithms are proposed in the “few global knowledge” and “zero-configuration” paradigm, respectively. Correspondingly, a high-efficient retrieve algorithm is designed for general storage algorithms using Robust Soliton distribution. We observe that the successful decoding probability can be provisioned by properly selecting parameters—the ratio of number of source node and total nodes, and the storage capacity M in each node.

1. Introduction

WSNs have attracted a lot of attention recently due to their broad applications. With the rapid development of microelectromechanical system, sensor nodes can be made much smaller with less cost. “Smart dust”, as a form of WSNs, will become one of the most potential applications in real world [1]. They can be deployed in tragedy, isolated, and obscured fields to monitor objects, detect fires, temperature, flood, and other disaster incidents such as earthquakes, landslides, and ice damage. Sensing networks are ideal for such scenarios since conventional sensing methods that involve human participation within the sensing region are often too dangerous. These scenarios offer a challenging design environment because the nodes used to collect and transmit data can fail suddenly and unpredictably as they may melt, corrode, or get smashed. Hence, it is necessary to design reliable storage strategies to collect sensed data from sensors before they disappear from the network.

In 2006, Kamra et al. [2] designed and analyzed techniques to increase “persistence” of sensed data based on growth code—a variant of fountain code. Later on, Lin et al.

[3] proposed an algorithm that uses random walks with traps to disseminate the source packets in the WSNs. They employed the Metropolis algorithm to specify transition probabilities of the random walks. However, the knowledge of the total number of sensors N , sources K , and the maximum node degree of the graph are required in their works. Recently, Aly et al. [4] proposed two new decentralized algorithms with limited or no knowledge of global information based on raptor codes. They afterward proposed two distributed flooding-based storage algorithms [5] for a WSNs wherein all nodes serve as sources as well as storage nodes, and the results demonstrated that it is required to query only 20–30% of the network nodes in order to retrieve the data collected by the N sensing nodes, in such a specific scenario where the buffer size is 10% of the network size. As a conclusion, a review on fountain code-based storage technologies is elaborated in Section 2.

The ultimate goal of any storage strategies is to get the maximum data recovering possibilities while encountering loss of data. A storage strategy needs to include two parts, that is, “how to store” and “how to retrieve”. The two parts should be well matched with each other just like a

decoding algorithm needs to be suitable for the encoding process. However, most of the previous works only focus on the storage part involving how to network the backup of sensing packages to each storage node and how to process the back-up packages in distributed way. In this paper, we extend the works to a complete solution including both storage and recovery sections. With respect to the storage part, two IFDS algorithms are proposed in the “limited global knowledge” and “zero-configuration” paradigm, respectively. For the recovery part, a belief propagation and Gaussian elimination-based recovery Algorithm (BGRA) is designed for data retrieve in close connection with the proposed storage algorithms, and it is suitable for any storage algorithms using Robust Soliton distribution. Moreover, the general scenarios with consideration of the percentage of source node number K among total number of nodes N and the storage capacity M of each node are studied. We analyze in detail how the three parameters affect the data retrieve. The results indicate that a WSN with designable successful decoding probability can be deployed by selecting proper N , K , and M , which lays some foundation for the application of WSNs in disaster scenarios.

2. Fountain Code-Based Storage in WSNs

Fountain codes are a new class of rateless codes with finite dimension and infinite block length. The first class of efficient universal fountain codes was invented by Luby [6] and is called LT codes. The codes are designed for channels with erasures such as internet, but many distinctive characteristics make the codes become an excellent solution in a wide variety of situations. MacKay [7] mentioned two major applications in his review of fountain code, one is for broadcast and the other is for storage. In storage applications, fountain codes can be used to spray encoded packets as backup of a file on more than one storage device so as to prevent data loss caused by catastrophic failures of unreliable storage device; and to recover the file, one simply needs to gather enough packets from any intact devices and skip over the corrupted packets on the broken devices. Actually, the distributed storage model in WSNs is very similar to the case, and it seems easier to implement in WSN since the communication network used to bridge nodes makes it convenient to spray the back-up packages. In a sensor network, the storage device is the node with storage units. In order to prevent data loss caused by unexpected failure of the storage node, a similar solution is to network the important sensed data to multiple storage nodes, encoding them distributedly using fountain code and to store them as a backup. The original sensed data can be retrieved as long as to query enough storage nodes with enough encoded packages.

Based on these points, many researchers follow closely with fountain code-based decentralized storage technology and give specialized solutions to the storage problems of WSNs in disaster environments [2–6, 8]. The basic approach is to achieve distributed encoding in each storage node using simple exclusive-or operations. Specific implementations

adopt the encoding process of growth code [2], LT code [3], or raptor code [4], respectively.

The way to disseminate the original sensed data to each storage node assumes crucial role in recovery performance of storage data in WSNs. Random walks [3, 4] and flooding [5] are two major ways to spray sensed data. The flooding dissemination adopts a very simple operation that each node floods the sensed data to all its neighbors and decides whether to store or discard the received packages according to the probability computed by random algorithms. Random walks employ Metropolis algorithm to disseminate the source packets. The number of random walks launched from each sensing node and the probabilistic forwarding tables for random walks are computed by the Metropolis algorithm. As long as a source block stops at a node at the end of the random walk, this node will store this source block. After all source blocks are disseminated, each storage node generates its encoded block. The basic features of the methods are summarized in Table 1.

According to global information requirements, the storage algorithm can be classified into two categories—“limited global knowledge” or “zero-configuration” algorithms [3–5]. If each node in the network knows the value of K —the number of sources, and the value of N —the number of storage nodes as a prerequisite for the designed storage algorithm, then the algorithm works in the “limited global knowledge” paradigm. However, in many scenarios, especially, when changes of network topologies may occur due to node joining-in or node failures, the exact value of N may not be available for all nodes. On the other hand, the number of sources K usually depends on the environment measurements or some events, and thus the exact value of K may not be known by each node either. As a result, to design a fully distributed storage algorithm which does not require any global information with “zero configuration” is very important and useful. In previous literatures, exact decentralized fountain codes (EDFC) and approximate decentralized fountain codes (ADFC) [3], distributed storage algorithms (DSA)-I [5] and raptor codes based distributed storage (RCDS)-I [4] are “limited global knowledge” based algorithms, and distributed storage algorithms (DSA)-II [5] and raptor codes based distributed storage (RCDS)-II [4] functions in “zero configuration”. In the mode of “zero configuration”, random walks can be used to estimate the network scale and further to approximately compute N , K in order to decide how to set the TTL segment (or maximum hop) of the package.

Viewed from the perspective of the recovery behavior, LT codes or Raptor codes based storage algorithm adopts the belief propagation (BP) process, which is recommended by Luby for decoding of fountain codes [6], as the recovery algorithm due to its low complexity. However, even though BP algorithm is simple and easy to implement, it does not explore all the encoding information in generator matrix G , so we do some amelioration on BP algorithm for a full exploitation of all the encoding information to enhance the retrieve performance. The detailed description is presented in Section 5. In addition, Growth code takes two situations into account—full recovery or optimal partial recovery.

TABLE 1: Comparison of flooding- and random walks-based data dissemination.

	Flooding	Random walks
Communication overhead	Big communication overhead	Small communication overhead
Global information requirements	Global Information is not required. Flooding disseminations can work in zero configuration combined with certain estimation of the global information	The global information of the total number of sensors N , sources K , and the maximum node degree of the graph are required. (However, the specific algorithm can be used for estimation of global information)
Implementation complexity	Simple and easy operations	Complex operations
Degree distribution guarantee	Distributed encoding process can be well guaranteed to satisfy Robust Soliton distribution	Using specific strategies to guarantee or approximate the Robust Soliton distribution

The goal is trying to completely recover all the storage data, but while it does not achieve full recovery, then pursuing the maximum of partial recovery to retrieve more storage data. This is a reasonable consideration in storage application. Inspired by these works, we proposed the information fusion based storage and retrieve algorithms for WSNs in disaster scenarios. Compared with the classic WSN paradigm, the major advantages using IFDS are the following. (i) IFDS algorithms adopt “flooding” to achieve data dissemination task, and each node never needs to keep a route table to sink node. Hence, each node never needs to rebuild new routing and update the route table due to failure of the nodes on the path to sink node. It is suitable for application in disaster environment. (ii) Over the various paths to sink node by flooding, each path transmits the “supplementary data” which includes not only the data information but also its relationship with other encoded blocks, so it has certain degree of redundancy, serving as a kind of “back up” for storage purpose. However, these advantages are at the cost of increasing a degree of communication overhead, computational overhead, and complexity.

With respect to the previous algorithms, several improvements are provided by IFDS algorithms. In the storage block assembling phase, the proposed IFDS algorithms do not need judge to accept or reject a data packet every time while the packet arrives a node, and each node in WSN calculates its own degree in preprocessing phase, which does once for all. In data dissemination phase, IFDS has a robust estimation method for hop segment $C_{\text{hop}}(s_{Si})$ in order to guarantee the desired dissemination task, so that all the sensed data can arrive at each node at least one time. Moreover, most of present algorithms never consider the node storage capacity which is actually an important factor affecting the data retrieve performance. The IFDS algorithms adopt a simple storage model for each node and give a detailed analysis on its behavior on data retrieve. Overall, the algorithms enhance the successful decoding probability, and we observe that a WSN with designable successful decoding probability can be deployed by selecting proper parameters—the ratio of source node number K and total number of nodes N , and the storage capacity M of each node. In what follows, the algorithms are discussed in detail.

3. Preliminaries and Modeling

In a real WSN, sensors are usually classified into several types based on different functions they assume. In Figure 1, we classify the set of sensors into the three kinds. Source sensors are located in the area where we expect to monitor for specific application. Source sensors are able to perform monitoring and generate packets of sensed data. Relay sensors collect received data into their buffer memory and are able to produce encoded blocks based on the distributed storage algorithm. Collector sensors or base station represents one or small number of WSNs nodes that are connected with the external network. The collector nodes is to collect data from their neighbors, recover the set of K source packets $\{B_{s1}, B_{s2}, \dots, B_{sk}\}$ that originated at the source sensor nodes during a single time period, and forward this data to a database in the external network. Three classes of sensors are equipped with memory for data storage. Suppose that the WSN consists of N nodes that are uniformly distributed at random in a three-dimensional region. Among these N nodes, there are K source nodes that have information to be disseminated throughout the network for storage. The K nodes are uniformly and independently chosen at random among the N nodes. If all the nodes are assigned with sensing tasks, then K is equal to N , so the model is a general model which covers all the application scenarios with different ratio of N and K . We assume that no node has knowledge about the locations of other nodes and no routing table is maintained. Moreover, except the information of neighbor nodes, we assume that each node has limited or no knowledge of global information, working at “limited global knowledge” or “zero-configuration” status. The limited global information refers to the total number of nodes N and the total number of source nodes K . Any further global information, for example, the maximal number of neighbors in the network, is not available. In order to illustrate the two algorithms clearly, we will use the definitions of Node Degree $d_n(u)$ and Code Degree $d_c(y)$ in [5]. In order to achieve a better recovery performance, we hope that source data can be stored with balance in each node as a backup. Therefore, it is required to make data coming from different source nodes fused and stored throughout the network. These data stored

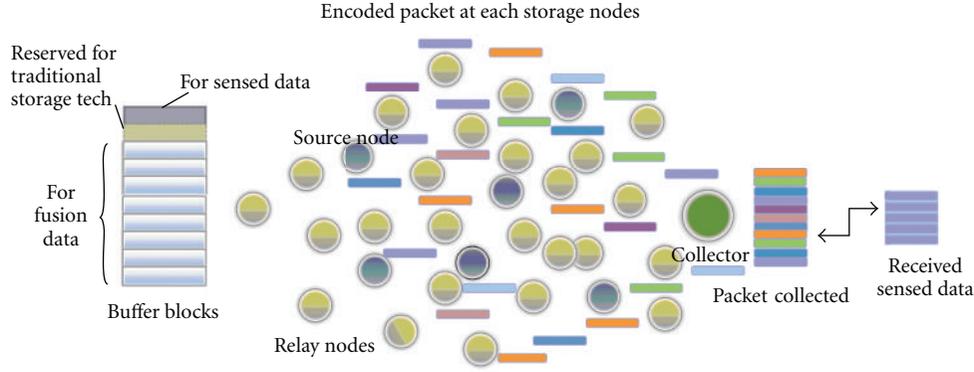


FIGURE 1: Illustration of information fusion-based distributed storage for WSNs.

in each node should be complementary and correlated. The process of data fusion is similar to the fountain code based on exclusive-or operations, but the process of encoding is a decentralized treatment in each node, so it works in a distributed way. Robust Soliton distribution [6] is adopted in our algorithms. In order to analyze the effect of the node capacity on the successful decoding probability as well as to make implementation of our algorithm easier, we model the node buffer as follows. The total buffer at each node is divided into several blocks according to the principle that each block has the size of sensed data package as a unit. For the source node, the first storage block is used for storing its own sensing data, the rest for distributed storage. For the relay nodes, all the blocks are used to store fused data. Each node reserves one storage block for traditional storage technology; in our implementation, we just simply store a selected packet copy with degree one. The buffer model is shown in Figure 1. Each block is labeled with one code degree to store fused data of certain several source packets. It is convenient to achieve the buffer model in practical applications using the existing hardware and software storage technology. In disaster areas, two cases of the node failure might happen. One is that a few nodes disappear due to out-charge, smash, or other reasons; the other is mass destruction or failure of nodes in a local area. Our storage strategy is designed for both scenarios.

4. Data Dissemination and Decentralized Storage

The concept of fountain code is a basis for the decentralized storage in WSNs. Although it is centralized in many coding literatures, we want to give a brief introduction of LT codes, especially concerning the Robust Soliton distribution for a better discussion on our storage and recovery algorithms in the following sections. Two storage algorithms are then presented to explain how to disseminate the sensed data and how to implement the encoding of LT codes at each storage node in decentralized way.

4.1. A Brief Introduction of LT Codes and Degree Distribution. Fountain is a class of erasure codes capable of reaching optimal erasure recovery on the binary erasure channels without fixing the rate. Fountain codes have remarkably simple encoding and decoding algorithms. In order to create an encoded symbol, an encoding host runs the encoding process as follows. Firstly, randomly choose the degree d_n of the packet from a degree distribution $r(d)$; the appropriate choice of r depends on the source file size k . Then, choose, uniformly at random, d_n distinct input packets, and then the sum of these input symbols over a suitable finite field (typically F_2) comprises the value of the encoded symbol. The concrete process of generating an encoding symbol using LT codes consists of three simple steps as follows [6]:

- (i) randomly choose the degree d of the encoding symbol from a degree distribution. The design and analysis of a good degree distribution is a primary focus of the remainder of this paper;
- (ii) choose uniformly at random d distinct input symbols as neighbors of the encoding symbol;
- (iii) the value of the encoding symbol is the exclusive-or of the d neighbors.

Each encoding symbol has a degree chosen independently from a degree distribution. Degree distribution $\rho(d)$ is the probability that an encoding symbol has degree d . Luby gives two degree distributions—the Ideal Soliton distribution and its melioration—the Robust Soliton distribution.

Definition 1 (Robust Soliton distribution [1]). For constants $c > 0$ and $\delta \in [0, 1]$, the Robust Soliton distribution $\mu(i)$ is given by

$$\mu(i) = \frac{\rho(i) + \tau(i)}{\beta}, \quad \text{for } 1 \leq i \leq k, \quad (1)$$

where $\beta = \sum_{i=1}^k (\rho(i) + \tau(i))$.

Here, $\rho(i)$ is Ideal Soliton distribution which is a probability distribution over $1 \leq i \leq k$; $\rho(i)$ and $\tau(i)$ are given by

$$\rho(i) = \begin{cases} \frac{1}{k}, & \text{for } i = 1, \\ \frac{1}{i(i-1)}, & \text{for } 2 \leq i \leq k, \end{cases}$$

$$\tau(i) = \begin{cases} \frac{S}{ik}, & \text{for } 1 \leq i \leq \frac{k}{S} - 1, \\ \frac{S \ln(S/\delta)}{k}, & \text{for } i = \frac{k}{S}, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The parameter S represents the average number of degree one code symbols and is defined as

$$S = c \cdot \sqrt{k} \cdot \ln\left(\frac{k}{\delta}\right). \quad (3)$$

As for decoding system, the most used is the BP algorithm and GE algorithm. BP algorithm first searches all degree-one symbol or say the column j that contains only one 1. And then all the 1's in its rows are canceled and their T_i are xored with T_j . The above process is iterated until the matrix G becomes all-0 matrix (decoding success) or until no more degree-one symbols can be found (decoding failure). BP is simple and fast, but while it encounters decoding failure, it may never use all the encoding columns of encoding G matrix as this causes a waste of part of encoding information. GE decoding algorithm adopts Gaussian elimination to solve problem $T = SG$ over typically F_2 , which consists of two steps: triangularization step and back-substitution step. In the triangularization step, the goal is to convert the given matrix using row operations to upper triangular matrix. If the triangularization step is successful, then the back-substitution step can proceed by converting the triangular matrix into the identity matrix after which the GE is successfully finished. GE could exert a higher successful decoding probability with smaller overhead but costs more time to complete decoding process due to its high complexity. BP algorithm is recommended by Luby as decoding methods for LT codes [5], which depends on the specific encoding process and degree distribution function of LT codes. Our study finds that Robust Soliton and Ideal Soliton distribution have a common characteristic. The probability of degree 2 is the greatest; the sum probability of smaller degrees is usually greater, or say that most probability exists at small degrees. Such a degree distribution provides a large probability for directly obtaining a degree-one column in each iteration of BP process. Ideal Soliton and Robust Soliton distribution with $k = 100$ are shown in Figures 2 and 3, respectively. We can see degree 2 is greatest with possibility around half maximum. Comparative larger possibility is distributed at smaller degrees such as degree 1, 2, 3, and 4. Based on these observations, BP algorithm can be basically adopted as the recovery algorithm for WSNs; however, in order to obtain a high successful retrieve probability and while never fully recovering all the data and then trying the best to

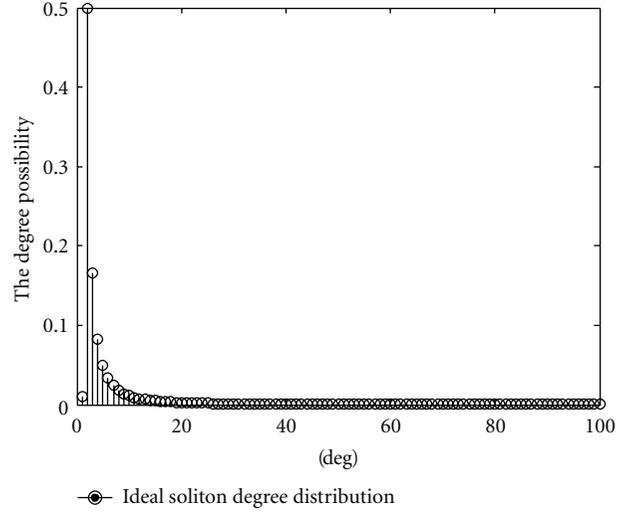


FIGURE 2: The Ideal Soliton degree distribution with $k = 100$.

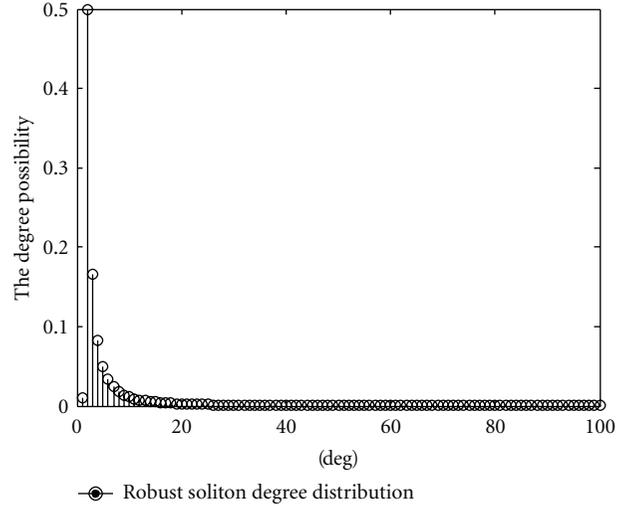


FIGURE 3: The Robust Soliton distribution $\mu(0.01, 0.01)$ with $k = 100$.

retrieve more data, we make some improvements over the BP algorithm. The part of works are focused in Section 5 concerning the recovery algorithm.

4.1.1. IFDS-I. In IFDS-I, each node in the network knows the limited global information N and K . Let $S = \{s_1, s_2, \dots, s_n\}$ be a set of sensing nodes or source nodes that are distributed randomly and uniformly in a field. Each of both source node and relay node acts as a storage node. Every node does not maintain routing or geographic tables, and the network topology G is not known. Every node can send a flooding message to the neighboring nodes and can detect the total number of neighbors by broadcasting a simple keep-alive message. Here we adopt flooding as the data dissemination method in order to satisfy the Robust Soliton distribution with reliability, based on which the proposed recovery algorithm can develop its advantages. We define the

```

                                IFDS-I Storage Algorithm
WHILE For each node in WSN  $S = \{s_1, s_2, \dots, s_n\}$ 
  DO {Receive packets from neighbors  $N(s_i)$ }
    {IF1 (hop segment  $\neq 0$ )}
      {IF2 (The node code of the received packet belongs to
        elements in  $d_{c(s_i)}$ .)}
        {IF3 (It is the first time to receive the packet  $B_{si}$ )}
          {Then accepting the packet into corresponding
            storage unit according to  $d_{(n)}$  &  $d_{c(s_i)}$ . If there is no
            other packets stored in storage block before, then store
            it directly.}
        ELSE
          { $M_j = \text{exclusive-or}(M_j, B_{si})$ ,  $j$  depends on  $d_{(n)}$  &  $d_{c(s_i)}$ .}
        ENDIF3}
      ELSE
        {Put the packet into forward queue, set
          hop segment as  $C_{\text{hop}}(B_{si}) = C_{\text{hop}}(B_{si}) - 1$ .}
      ENDIF2}
    ELSE
      {Discard the packet  $B_{si}$ , and no node to send to. }
    ENDIF1}
  ENDWHILE

```

ALGORITHM 1: The algorithm flow of IFDS-I.

average degree of the topology G as $d_{\text{mean}}(G)$. Each of the nodes in WSN except the collector nodes calculates its own code degree for its buffer blocks based on Robust Soliton distribution, which does once for all. The algorithm flow is illustrated as follows.

- (i) *Input*. A sensor network $S = \{s_1, s_2, \dots, s_n\}$ with N nodes; There are K source nodes which can produce source sensed packets $\{B_{s_1}, B_{s_2}, \dots, B_{s_k}\}$.
- (ii) *Output*. Storage buffer blocks $\{M_1, M_2, \dots, M_m\}$ for all sensors in S . Fused data are stored at each node according to node buffer model.
- (iii) *Preprocessing*. *Step i*. Choose randomly the code degree of the encoding packets from degree distribution function, structure a set $d_{(n)}$ with m elements for m buffer blocks at each node, $d_{(n)} = \{d_{M1(n)}, d_{M2(n)}, \dots, d_{Mm(n)}\}$. *Step ii*. Choose uniformly at random distinct source packets as degree distribution neighbors (ddn), $d_{c(s_i)}$ for each buffer blocks. $d_{c(s_i)} = \{(\text{ddn for } M_1), \dots, (\text{ddn for } M_m)\}$. *Step iii*. Keep alive link with neighbors and generate a set of neighbors $N(s_i)$ using flooding and then obtain node degree $d_{n(s_i)}$. For each source node $S = \{s_1, \dots, s_n\}$, generate source packet = node code, data style, hop, sensing data, flood to all of its neighbors $N(s_i)$, and set hop segment or TTL segment as $C_{\text{hop}}(B_{si}) = \lceil N/d_{\text{mean}}(G) \rceil - 1$. $d_{\text{mean}}(G)$ can be approximated by the node degree $d_{n(s_i)}$ of any arbitrary node s_i while the network is deployed with nodes of high density. $C_{\text{hop}}(B_{si})$ is a very important parameter for a desired data dissemination. The sensed data from

source nodes are desired to arrive at each node at least one time throughout the network; thus, the storage encoding process could be implemented rigidly according to the designed distribution degree function. Node code is a number that marks and distinguishes nodes in the WSN; Data style is used to tell the storage sensing data from other data; Hop is a data segment like TTL in TCP/IP protocol, which stands for the maximum hop of a packet. Compared with Aly's algorithms [5], the IFDS algorithms do not need to flip a coin to accept or reject a packet every time while a packet arrives a node, and each node in WSN calculates its own degree in preprocessing phase, which does once for all.

In order to show the data dissemination performance, we give an simple example for a WSN with parameters $N = 10$, $k = 5$. We define two matrix G_c and G_d . G_c is used to indicate the communication connectivity status between nodes in the WSN, it has N rows and N columns, and the matrix element $G_c(i, j)$ is a connectivity status with value 0 or 1; if $G_c(i, j) = 0$ ($i = 1, 2, \dots, N$; $j = 1, 2, \dots, N$), it means no direct connection exists between the nodes N_i and N_j ; on the contrary, if $G_c(i, j) = 1$, it means there exists a direct connection between the nodes N_i and N_j ; in the diagonal, we use 0 as default. G_d is used to record how the process of data dissemination goes, the matrix element $G_d(i, j)$ ($i = 1, 2, \dots, N$; $j = 1, 2, 3, \dots, k$) recodes the times of the sensed data from source node S_{S_j} arriving at node N_i .

In the example, we assume that N_1, N_2, \dots, N_5 are source nodes which are rewritten as the standard form $\{S_1, S_2, \dots, S_5\}$. At the beginning of flooding, all the source

```

                                BGRA Recovery Algorithm
WHILE ~ terminus (Set terminus = 0)
    IF1 (There exists one degree column in  $G$ )
        {Record sequence number of such columns into vector
          $rcdc$ .}
    ELSE
        IF2 (BP algorithm found one degree column earlier)
            {Construct  $G_g S_r = T_r$ ; }
        ELSE
            {Construct a linear equations  $GS = T$  and launch
             GE algorithm directly.}
        ENDIF2
    ENDIF1
    FOR1 (all the elements  $G(i, rcdc(j))$ )
        IF3 ( $G(i, rcdc(j)) = 1$ )
            {Record sequences number  $i$  of rows into  $rcdr$ .}
        ENDIF3
    ENDFOR1
    FOR2 (all the elements in  $rcdr$  and  $rcdc$ )
        { $S(rcdr(i)) = T(rcdc(i))$ ; }
    ENDFOR2
    FOR3 (all the elements in  $rcdc$ )
        {Set  $G(i, rcdc(j)) = 0$  ( $i = 1, \dots, n$ );
         FOR4 (all the elements in  $rcdr$ )
             IF4 ( $G(rcdr(i), j) = 1$ ;)
                 { $T(j) = x$  or ( $S_r(rcdr(1, i)), T(j)$ ); }
             ENDIF4
         ENDFOR4
         Set  $G(rcdr(i), j) = 0$ ; ( $i = 1, \dots, m$ ); }
    ENDFOR3
    IF5 (The  $G$  becomes all-0 matrix;)
        {Set terminus = 1;}
    ENDIF5
    IF6 (GE is used in the algorithm)
        {Combine the decoding results from BP and GE.}
    ENDIF6
ENDWHILE

```

ALGORITHM 2: The algorithm flow of BGRA.

nodes first calculate the hop segment value $C_{\text{hop}}(B_{S_j})$ ($j = 1, 2, \dots, 5$) for their flooding data blocks. In this case, all the source nodes obtain the same value, $C_{\text{hop}}(B_{S_j}) = (N/d_{n(s_i)}) - 1 = 1$. Then the source nodes fill the hop segment of data blocks with one and then flood the data to their neighbours. According to G_c , S_1 has neighbour nodes N_2, N_3, N_5, N_6 , and N_{10} ; S_1 floods the sensed data to these nodes, and G_d records the fact that the data block from S_1 arrives nodes N_2, N_3, N_5, N_6 , and N_{10} one time. The working flow for other source nodes is the same. After the first flooding, G_d becomes as Figure 4(b). Next, every nodes which received the sensed data during the first flooding will further flood the data to their neighbours. Based on G_c , the obtained G_d is further updated as Figure 4(c). And at the point, the hop value of data block is decreased to 0, so the flooding terminates. We can see that there is no 0 element in the final G_d , which indicates that each sensed data block is guaranteed to arrive at each node at least one time. Therefore, the data dissemination task is perfectly

completed, and the encoding process is also guaranteed for distributed storage.

In Figure 5, the red curve indicates the TTL of data packets or the number of transmissions required for a successful data dissemination, in order to assure that all the sensed data can arrive at each node at least one time. The blue curve is the actual TTL used by IFDS-I. It is obvious that the value on blue curve is equal or greater than the red one; this means that IFDS-I provides the data packets with a longer living life than enough, or say that the packets can arrive at more nodes. And we find that the counter of packet could actually be set as a number smaller than $N/d_{\text{mean}}(G)$, while WSN is a large network with high connectivity. The main operations of IFDS-I is shown in Algorithm 1.

4.1.2. IFDS-II. In IFDS-II, we assumed that N and K are known in advance for each node in the network. This might not be the case in practical disaster scenarios where the

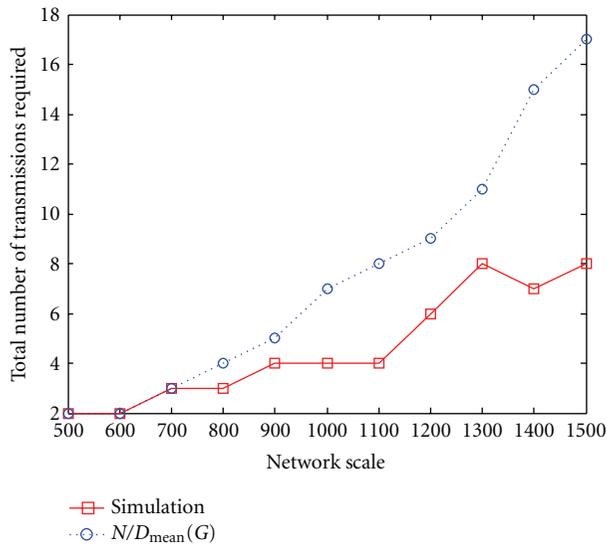
		G_c									
		N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}
S_1	N_1	0	1	1	0	1	1	0	0	0	1
S_2	N_2	1	0	1	0	1	0	0	0	1	1
S_3	N_3	1	1	0	0	1	0	0	0	1	1
S_4	N_4	0	0	0	0	1	0	1	1	1	1
S_5	N_5	1	1	1	1	0	0	0	0	0	1
	N_6	1	0	0	0	0	0	0	1	0	1
	N_7	0	0	0	1	0	0	0	1	1	1
	N_8	0	0	0	1	0	1	1	0	1	1
	N_9	0	1	1	1	0	0	1	1	0	1
	N_{10}	1	1	1	1	1	1	1	1	1	0

(a)

G_d after 1st flooding		G_d after 2nd flooding									
	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	
N_1	0	1	1	0	1	N_1	5	4	4	2	4
N_2	1	0	1	0	1	N_2	4	5	5	3	4
N_3	1	1	0	0	1	N_3	4	5	5	3	4
N_4	0	0	0	0	1	N_4	2	3	3	5	2
N_5	1	1	1	1	0	N_5	4	4	4	2	5
N_6	1	0	0	0	0	N_6	2	2	2	2	2
N_7	0	0	0	1	0	N_7	1	2	2	4	2
N_8	0	0	0	1	0	N_8	2	2	2	4	2
N_9	0	1	1	1	0	N_9	3	3	3	4	4
N_{10}	1	1	1	1	1	N_{10}	5	5	5	5	5

(b) (c)

FIGURE 4: Data dissemination implemented in IFDS-I.

FIGURE 5: The relationship between network scale and C_{hop} .

change of connectivity status may occur between nodes due to the event of sensor failure or new nodes joining in. Therefore, we extend IFDS-I to IFDS-II that is totally distributed without knowing global information with “zero configuration”. The idea is that each source node s_{Si} will estimate a value for its hop counter $C_{\text{hop}}(s_{Si})$ without knowing N and K . In IFDS-II, each source node s_{Si} will perform a hop-estimation phase that will calculate the value of the counter $C_{\text{hop}}(s_{Si})$. The hop-estimation process starts before data dissemination; however, the estimation not only works at the network startup; instead, it is dynamic process working throughout. The hop estimation is implemented while source nodes receive a keep-alive hop-estimation packet from their neighbors. A keep-alive hop-estimation packet consists of {node code, data style, estimated hop}.

Hop estimation: let s_{Si} be a source node in a distributed network. Each node s_{Si} , ($i = 1, 2, \dots, K$) will dynamically determine value of the counter $C_{\text{hop}}(s_{Si})$. The node s_{Si} knows its neighbors $N(s_{Si})$ by keep-alive message. Each source node will independently decide a value for its counter by following several steps. *Step i.* Flood a keep-alive hop-estimation packet = {node code, data style, estimated hop} to its neighbors $N(s_{Si})$. Node code segment fills its own node code; data style is labeled by hop-estimation packet, and estimated hop is recorded with 0. *Step ii.* Each node forwards the hop-estimation packet to its neighbors with estimated hop value plus one while receiving multiple the hop-estimation packets from the same source, the node compares them and choose the smallest to plus one. *Step iii.* When a source node received its corresponding hop-estimation packets again, it compares the values of estimated hop segments of all the packets received and choose the maximum as N_{dia} . *Step iv.* Set $C_{\text{hop}}(s_{Si}) = \lceil N_{\text{dia}}/2 \rceil$. Thus, one iteration of hop estimation is finished. The estimation algorithm flow is illustrated in Figure 6.

In the flow chart, each source node should maintain a hop value. The hop value is used for a real-time estimation of the network size based on the already received hop-estimation packets. It will be updated when a packet with a bigger estimated hop segment value is received. Every time when the packet passes through a node, the value of the estimated hop segment will increase one; so, a bigger hop segment means that the packet can reach to a node in a father location and it has come back to the source node again. Thus, the packets with the biggest estimated hop segment are those reaching the edge of the network. Continuous updating of the hop value until achieving convergence can be used to approximate the network radius or network scale. In this process, if structure of WSN alters, such as nodes failures, movements, or network expanding, all these changes will be reflected in the estimated hop segment of hop-estimation packets, and further the hop value will be also updated adaptively. For example, if the network is expanding, the hop value will increase and become stable after several updates. Moreover, while new sensing nodes are assigned in WSN, the estimation process will be triggered for these new source nodes which start estimating and knowing the network size. Once the hop counts $C_{\text{hop}}(s_{Si})$ is approximated at each source node, the encoding operations of IFDS-II are similar to

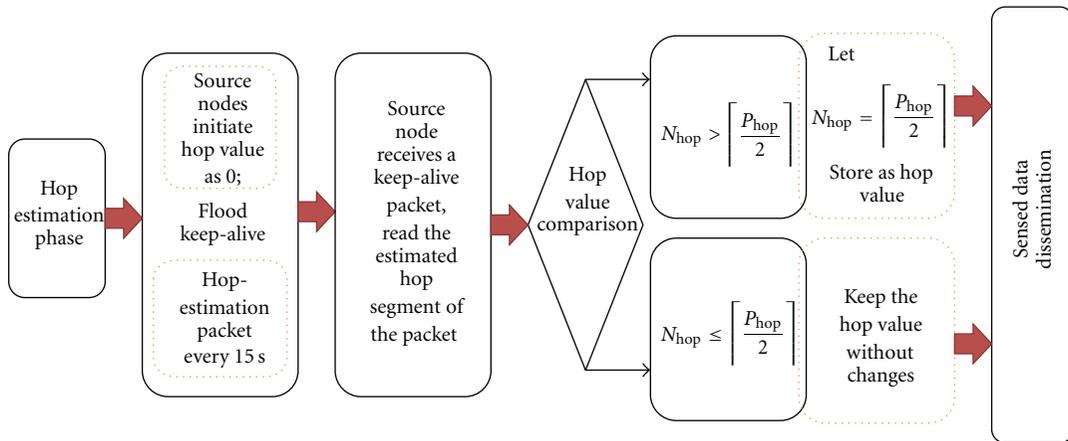


FIGURE 6: The hop-estimation process flow of IFDS-II.

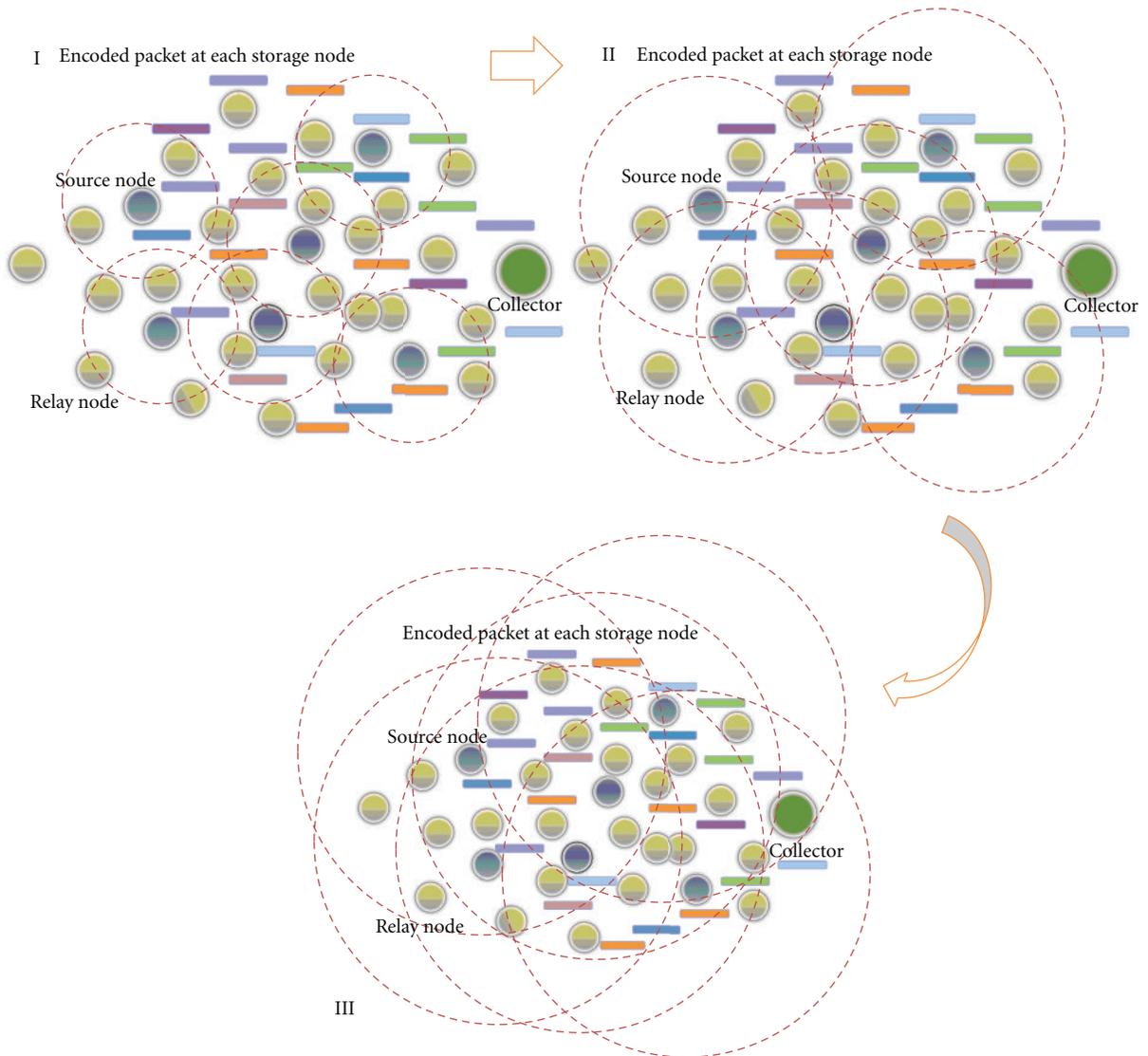


FIGURE 7: The data dissemination implemented by IFDS-II.

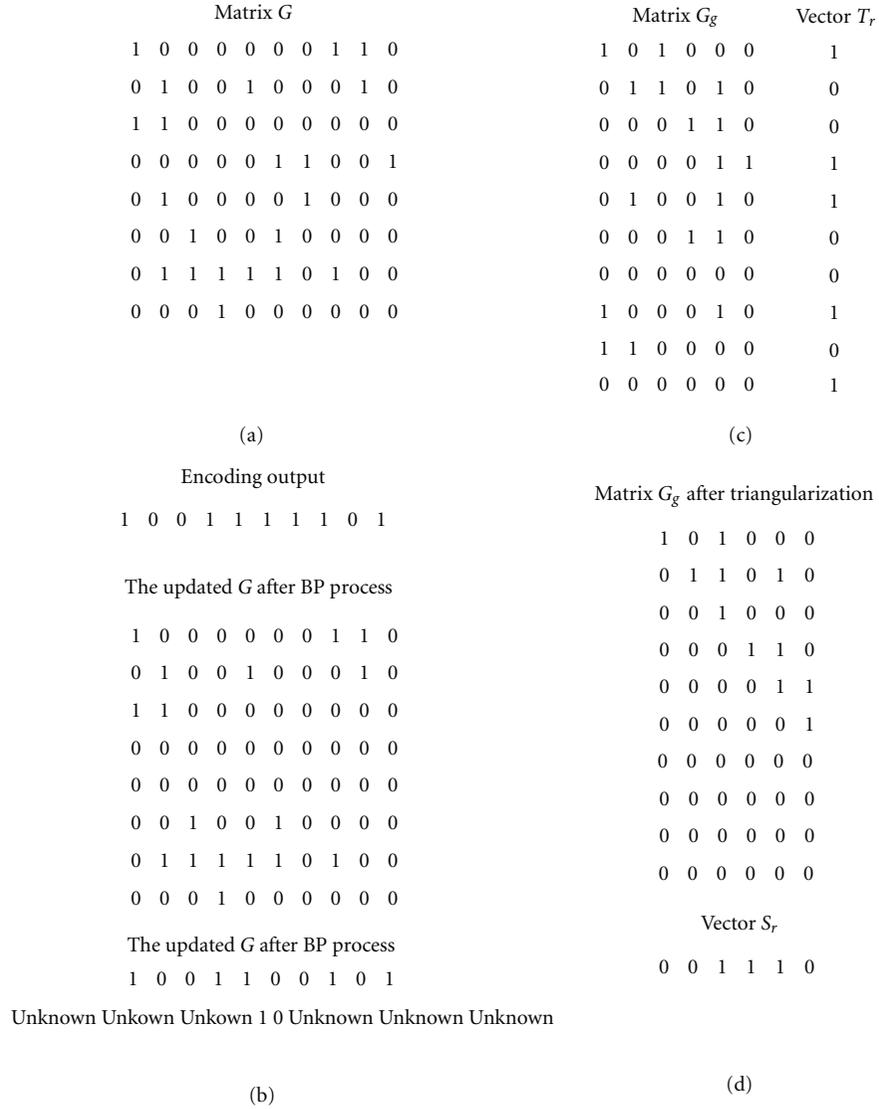


FIGURE 8: The original G matrix and the evolvement of G , T , and ST vector in the process of BGRA.

encoding operations of IFDS-I. In data dissemination phase, source nodes will assemble and flood the sensed data packets using the real-time hop value as the hop segment, and the completion evolution of dissemination task is shown as follows (Figure 7).

In the beginning stages (I, II), the maintained hop value is still never fully updated since the packets that spread to the network edge never come back; therefore, the assembled data packets cannot reach far from the source node due to a small TTL. Along with the dynamic estimation of the network size, the hop value is updated and continuously increasing; then the assembled data packet can reach farther in the process until the hop value reaches stable, and, at the point, the network radius or network scale can be correctly estimated; the packets flooded from source node can arrive at any nodes in the network. In our buffer model, one block is reserved for traditional storage method. Here we just randomly store one sensed packet with degree one using uniform distribution.

The storage performance of both IFDS-I and IFDS-II will be elaborated in Section 6 combining with the recovery algorithm proposed in Section 5.

4.2. Power Consumption of IFDS Algorithms. The overall energy in the WSN nodes is consumed in three distinct processes: data processing, data transmission, and sensing tasks. The proposed IFDS algorithms will majorly affect the first two processes; certain overhead will be brought during implementation of the algorithms. The communication overhead is the major overhead due to wide flooding for data dissemination. However, on the other hand, since the algorithms never need to maintain the routing tables, so they would never introduce the routing overhead. The distributed storage encoding will cause certain computation power consumption. In order to decrease the computation complexity, the proposed IFDS algorithms do not need to flip a coin to accept or reject a data packet every time

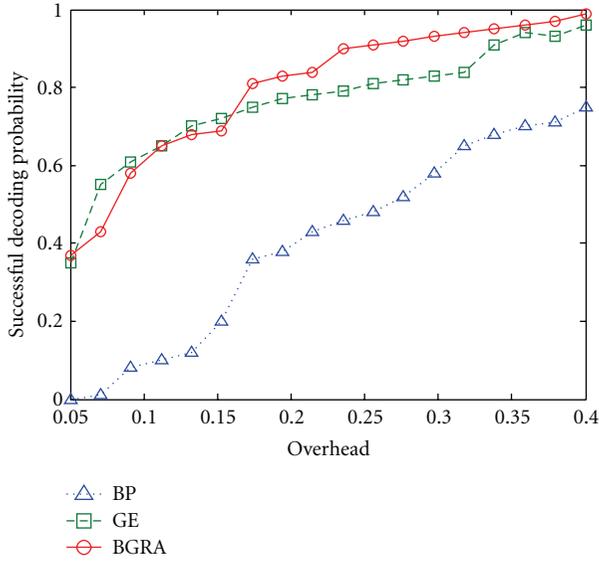


FIGURE 9: The successful decoding probability as a function of encoding overhead while $k = 100$ for LT code (0.01, 0.01).

while the packet arrives a node; instead, each node in WSN calculates its own degree in preprocessing phase, which does once for all. This is more energy efficient than the previous algorithms that will calculate the code degree and make selection from every arriving packet for encoding. The typical energy management techniques with the basic idea, to shut down sensors when not needed and wake them up when necessary, can be easily applied and combined with IFDS algorithms, because IFDS can estimate the network size adaptively. Moreover, Since the deployment of the WSNs in difficult-to-access areas makes it difficult to replace the batteries of sensor nodes. The use of solar cells, super capacitors, or rechargeable batteries is necessary for the long-term sensor node operation. A long-term operation could be achieved by adopting a combination of hardware and software techniques along with energy efficient WSN design.

5. Recovery Algorithm

In Section 3, we refer that BP algorithm is generally used as a basic recovery algorithm to retrieve storage data in WSNs. However, the BP algorithm has some limitations while used for storage applications. BP algorithm is fast with low complexity but it must find degree-one column in generator matrix G for each iteration to make decoding process go ahead, which prohibits the improvement of recovery efficiency, because it may not use all the encoding relationship recorded in G while it cannot find degree-one column in process, or say that the BP algorithm never takes use of all the encoding information while encountering stop set which will greatly reduce the successful probability of complete recovery. Gaussian elimination (GE) is another typical way for decoding of LT codes, and now many improved algorithms are proposed for decoding of fountain codes [9–13]. In order to solve the problems caused by the

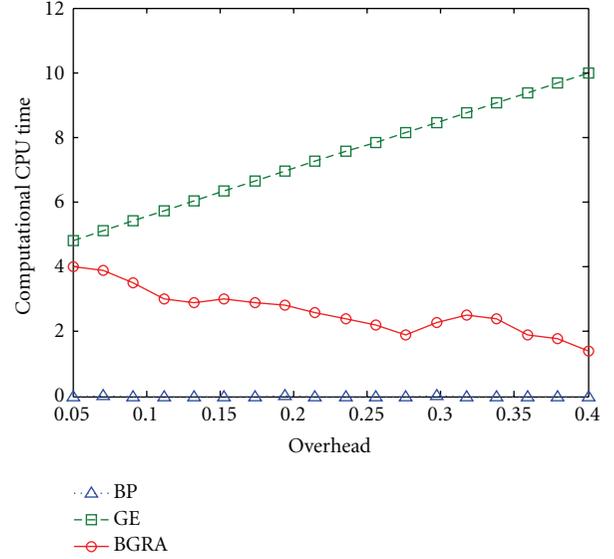


FIGURE 10: The computational CPU time as a function of encoding overhead while $k = 100$ for LT code (0.01, 0.01).

stop set of the BP algorithm, we expect to combine BP and GE algorithms and to provide a good tradeoff between the advantages of both algorithms for a better recovery performance. The core idea is to first use BP algorithm with very low complexity directly to find degree-one column in G and then to consider adoption of GE operation to find the potential degree-one column from the remaining columns. It can overcome the negative influence of algorithm termination of BP and improve the decoding efficiency for LT codes through fully digging out encoding information from matrix G . We name the algorithm, Belief propagation and Gaussian elimination based Recovery Algorithm (BGRA). The algorithm flow is shown in Algorithm 2.

We give an intuition of how the BGRA works by considering a simple example. In the example, we set $N = 10$, $K = 8$, that is, there are 10 storage nodes in the WSN amongst which there are 8 sensing nodes assigned with the monitor tasks. We use one bit with 0 or 1 to stand for one sensed source packet produced at sensing nodes. In practice, a package is to carry a certain bit of information, but here this does not affect our description of this algorithm. The sensed data from the 8 sensing nodes is $S_o = \{0, 0, 1, 1, 0, 1, 1, 0\}$. After the distributed encoding process using Robust Soliton distribution μ ($c = 0.01$, $\delta = 0.01$) is $\{1, 0, 0, 1, 1, 1, 1, 0, 1\}$ which is stored in the storage unit of the 10 storage nodes, respectively, the matrix G that records encoding information such as code degree and neighbors is shown in Figure 8(a). BP algorithm is launched to decode firstly. It searches all the columns with one degree in G . For matrix G in Figure 8(a), only the tenth column is degree-one column. BP algorithm starts decoding from the column and set $s_4 = t_{10} = 1$, then the algorithm refreshes all the corresponding columns in G and updates the decoding input vector T as well as decoding output vector S_T according to the recovery algorithm. The updated G and T are shown in Figure 8(b). Since the

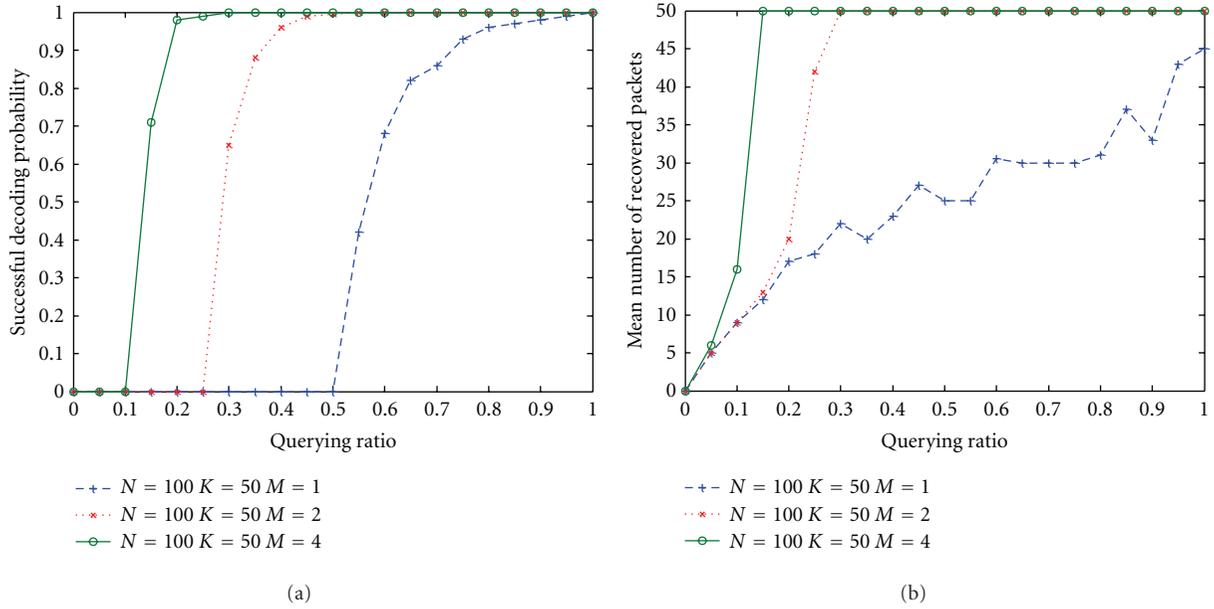


FIGURE 11: (a) The successful decoding performance; (b) the mean number of recovered packets as a function of querying ratio while $N = 100$, $K = 50$, $K/N = 50\%$, and M changes.

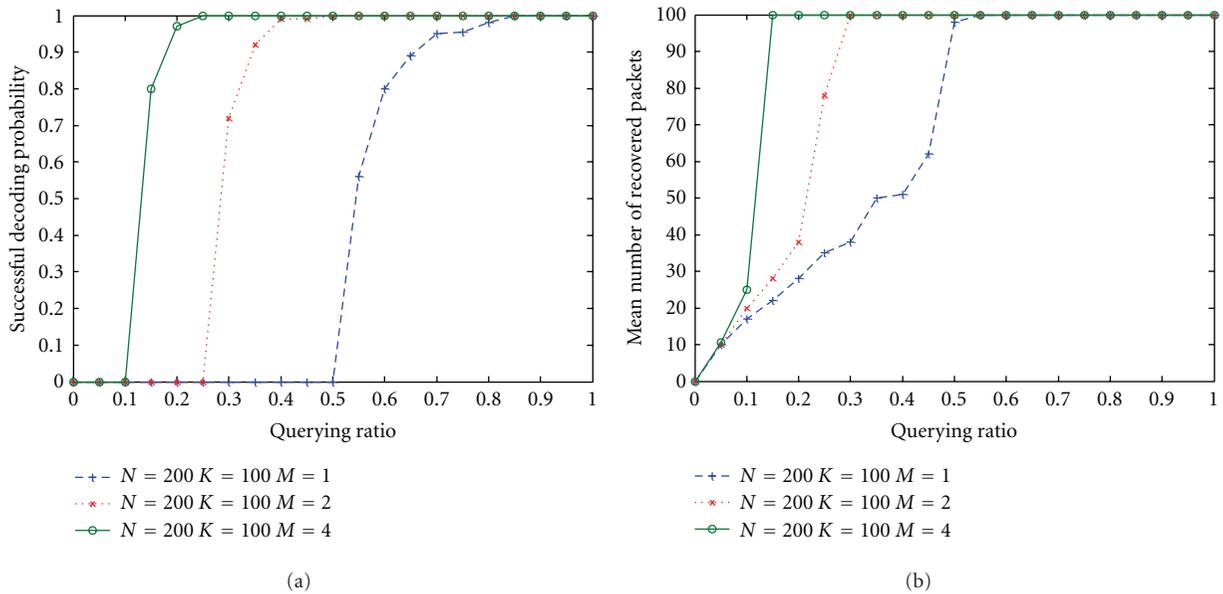


FIGURE 12: (a) The successful decoding performance; (b) the mean number of recovered packets as a function of querying ratio while $N = 200$, $K = 100$, $K/N = 50\%$, and M changes.

algorithm cannot find another degree-one column, BGRA launches GE algorithm to continue decoding based on the rest columns of G , which is the reason why BGRA can fully exploit encoding information from G . Before GE algorithm is started, linear equations $G_g S_r = T_r$ are constructed over the rest columns of G after BP process. G_g is a matrix that records code degree and neighbors of source symbols except those decoded during previous BP process. In order to construct G_g , the rows and columns with all-0 elements are deleted from G and its transpose is G_g . T_r is the updated T after

deleting elements corresponding with those all-0 columns. In our example, G_g is the transpose of G after deleting the fourth row and the tenth column. T_r is T without the tenth element. G_g and T_r are shown in Figure 8(c). The deleted row and column has only 0 elements after the first iteration of BP algorithm. The matrix G_g after triangularization is shown in Figure 8(d) as well as the solution S_r of linear equations $G_g S_r = T_r$. The final step of BGRA algorithm is to combine both parts of recovery results from BP and GE and to recover the source symbols. The solution is very

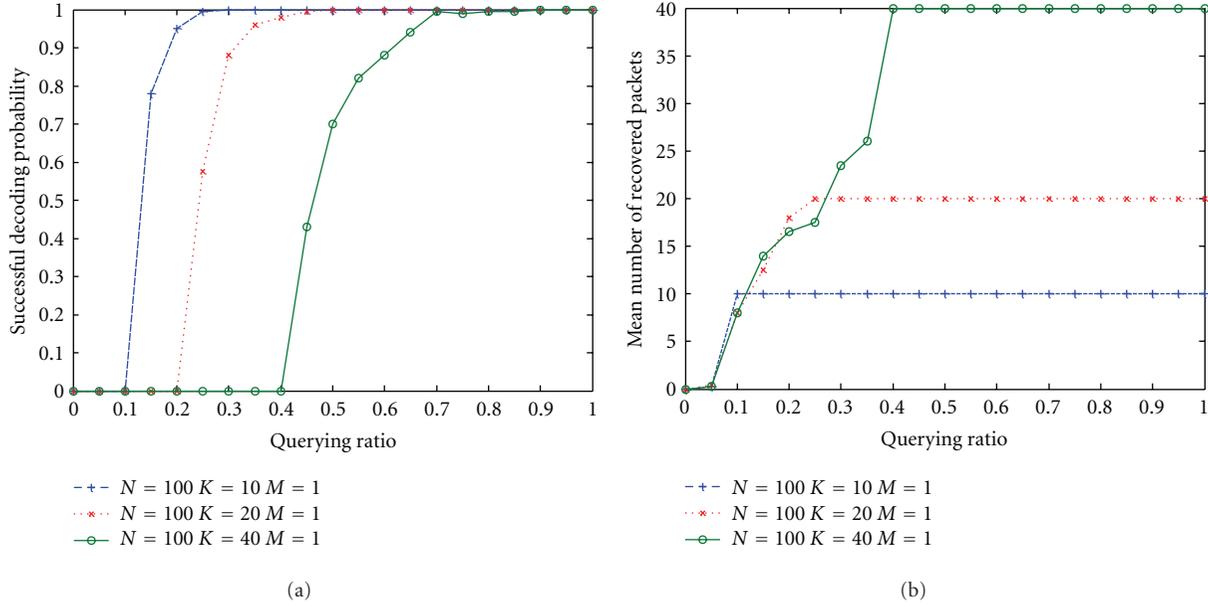


FIGURE 13: (a) The successful decoding performance; (b) the mean number of recovered packets as a function of querying ratio for networks with $N = 100, M = 1$ while the value of K/N changes.

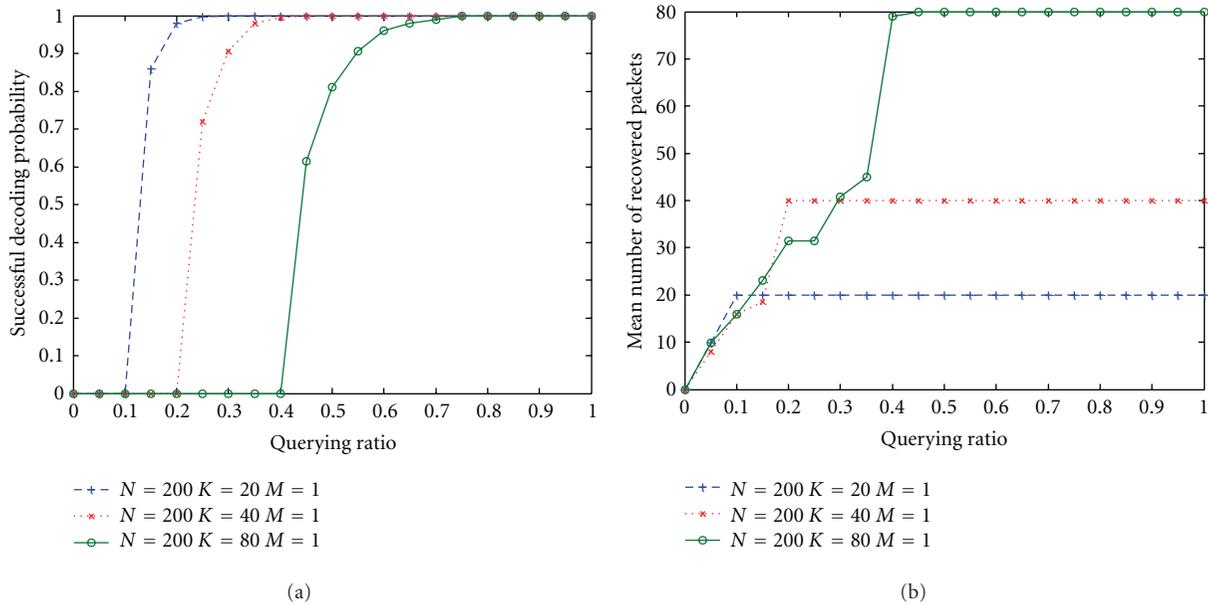


FIGURE 14: (a) The successful decoding performance; (b) the mean number of recovered packets as a function of querying ratio for networks with $N = 200, M = 1$ while the value of K/N changes.

simple; it just needs to orderly put all the elements of vector S_r into the places labeled “Unknown” in vector S_T . BGRA ends at this point, and the retrieve data is obtained as $S_T = \{0, 0, 1, 1, 0, 1, 1, 0\}$. In addition, the On the Fly GE algorithm proposed in [10] can be used for solution of $G_g S_r = T_r$ with a lower complexity.

BGRA algorithm, as integration of BP and GE algorithm, provides a good tradeoff between the advantages of both the BP algorithm and the GE algorithm. The BGRA algorithm has reasonable decoding complexity that is in between the

low complexity of the BP algorithm and the high decoding complexity of the GE algorithm. The BGRA algorithm has a successful decoding probability that is comparable to that of the GE algorithm and significantly better than that of the BP algorithm. Additionally, the decoding CPU computational time of BGRA algorithm does not rapidly increase with overhead, as is the case for the GE algorithm.

In order to compare the three algorithms, we experiment BP, GE, and BGRA algorithms with different values of overhead for the number of sensed source data $k = 100$,

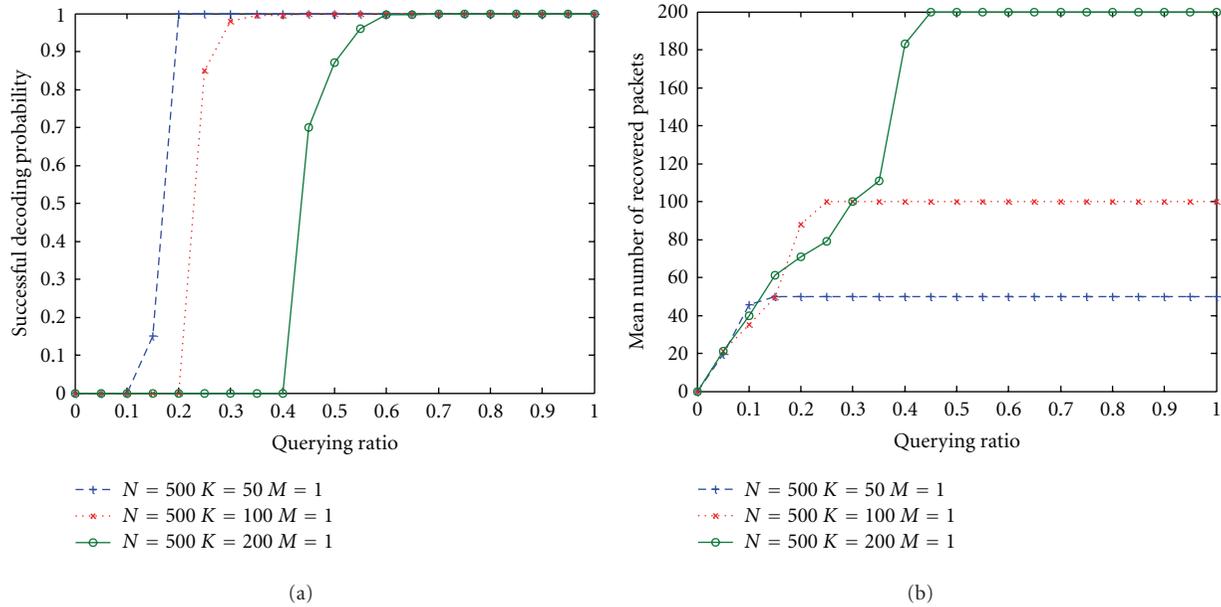


FIGURE 15: (a) The successful decoding performance; (b) the mean number of recovered packets as a function of querying ratio for networks with $N = 500$, $M = 1$ while the value of K/N changes.

using Robust soliton degree distribution ($c = 0.01$, $\delta = 0.01$) for storage encoding. Assuming that the number of retrieved encoded storage packets is n , the overhead is defined as $q = (n - k)/k$. A bigger overhead means that more storage nodes are queried for data recovery. Figure 9 shows that BP algorithm performs poor while overhead is small, but GE and BGRA exert a higher decoding performance; while overhead is more than about 0.2, both will produce a successful decoding probability over 90%. In Figure 10, we show the computational CPU time Rt using BP, GE, and BGRA while changing overhead. Compared with Rt of BP which keeps stable along with overhead's increase, Rt of GE grows quickly while overhead increases. Rt of BGRA ranges between BP and GE with the trend that more overhead costs less time to decode. Based on the analysis above, BP algorithm actually functions poor on data recovery behavior. Taking the computational capacity of node and successful retrieve probability into account, BGRA can provide a good data recovery performance.

6. Results and Discussion

Figures 11 and 12 show the decoding performance with IFDS-I for different network scale with fixed ratio of $K/N = 50\%$ while M changes. Figures 13–15 illustrate the decoding performance with IFDS-II for different network scale while the value of K/N changes. Querying ratio is the percentage of the number of queried nodes q among the total number of nodes N . Successful decoding probability P is the probability that the K source packets are all recovered from the q queried nodes. The successful decoding probability for different scale networks with fixed $K/N = 50\%$ while M is 1, 2, 4 is shown in Figures 11(a) and 12(a). It is obvious that a bigger

node buffer size can provide a better decoding performance; especially for a WSN while each node just has a small storage memory, to increase the storage capacity is a good way to improve the successful retrieve probability. Figures 11(b) and 12(b) illustrate the number of recovered packets at different querying ratio. Although a full recovery at the low querying ratio is never achieved, it could well give partial recovery of which performance is better than only using BP algorithm. The number of recovered packets is generally more than the number of queried nodes with a rapid increase along with the increase of querying ratio. The results show that the retrieve performance has a similar relationship with the querying ratio for any different scale networks while K/N is fixed.

Figures 13–15 illustrate the decoding performance with IFDS-II for different scale networks while each node has fixed one unit buffer block and K/N changes; a higher percentage of source node requires more queried nodes to retrieve the sensed data, which is in line with common sense. The results also show that 10% increase in the percentage of source node needs to query 10–20% more nodes in order to keep the same successful decoding probability. From Figures 13–15(a), we can see if the number of any queried nodes is slightly more than k ; then it could achieve a full data retrieve. Based on these findings, we can easily deploy a WSN for monitoring in practical catastrophic environment.

Figure 16(a) compares the performance of IFDS-I with DSA-I [5]. DSA-I algorithm utilizes flooding and the node degree of each node to disseminate the sensed data from sensors throughout the network, and the encoded data are stored distributedly in each node for later data retrieve. Both IFDS and DSA algorithms consider the storage capacity of each node which is modeled with the number of buffer units. In [5], the authors analyze the data retrieve performance

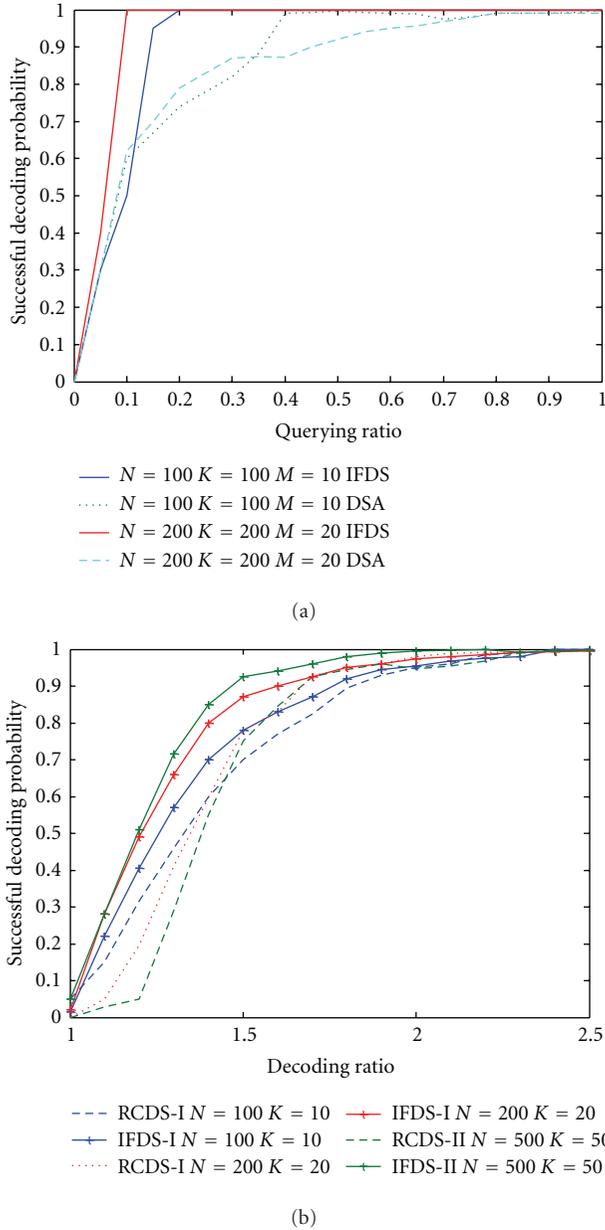


FIGURE 16: (a) Comparison of decoding performance using IFDS-I and DSA-I; (b) comparison of decoding performance using IFDS and RCDS.

of DSA-I when the node buffer size is 10% of the network storage size. Under the same condition, we give our results and we find that IFDS has a better decoding performance. Compared with 20–30% querying nodes using DSA-I, IFDS-I only requires to query 15% nodes in network with $N = K = 100$ and 10% nodes in network with $N = K = 200$ for nice recovery of all sensed packages from source nodes. IFDS has a similar data-disseminating method and the same degree distribution function with DSA; however, a better recovery algorithm is used by IFDS-I, so IFDS-I functions better than DSA-I. Figure 16(b) shows the performance

comparison between the IFDS and RCDS algorithms. RCDS-I are “limited global knowledge” based algorithms, RCDS-II works in “zero-configuration”. In order to compare IFDS and RCDS, we use the definition of the decoding ratio in [4]. The decoding ratio is defined as the ratio between the number of querying nodes and the number of sources. It can be seen that IFDS algorithms can exert a higher successful recovery probability while querying the same number of sensor nodes for different network scale. While the decoding ratio is between 1.2 to 2, the successful recovery probability of IFDS algorithm is about 10–15% greater than RCDS algorithm. In addition, we can also observe that the IFDS algorithms perform better for larger scale WSNs.

7. Conclusion

This paper proposes two IFDS algorithms in the “few global knowledge” and “zero-configuration” paradigm, respectively. An efficient retrieve algorithm is designed correspondingly, which is generally suitable for the storage algorithms using Robust Soliton distribution. The algorithms enhance the successful decoding probability. The detailed results of data retrieve performance and its relationship with three parameters—the total number of nodes N , the number of sensing nodes K , and the number of storage units equipped at each node M , is studied, which shows that we can control the successful decoding probability through setting up desired network parameters.

Acknowledgments

The authors thank the Research Project from Communication Branch of Yunnan Power Grid Corporation, Training Program of Yunnan Province for Middle-aged and Young Leaders of Disciplines in Science and Technology (Grant no. 2008PY031), and the National Natural Science Foundation of China (Grant no. 60861002) for financial support.

References

- [1] D. Butler, “2020 Computing: everything, everywhere,” *Nature*, vol. 440, no. 7083, pp. 402–405, 2006.
- [2] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, “Growth codes: maximizing sensor network data persistence,” in *Proceedings of the ACM SIGCOMM*, Pisa, Italy, September 2006.
- [3] Y. Lin, B. Liang, and B. Li, “Data persistence in large-scale sensor networks with decentralized fountain codes,” in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 1658–1666, Anchorage, Alaska, USA, May 2007.
- [4] S. A. Aly, Z. Kong, and E. Soljanin, “Raptor codes based distributed storage algorithms for wireless sensor networks,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '08)*, pp. 2051–2055, Toronto, Canada, July 2008.
- [5] S. A. Aly, M. Youssef, H. S. Darwish, and M. Zidan, “Distributed flooding-based storage algorithms for large-scale wireless sensor networks,” in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, Dresden, Germany, June 2009.

- [6] M. Luby, "LT codes," in *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science (FOCS '02)*, Vancouver, BC, Canada, November 2002.
- [7] D. J. C. MacKay, "Fountain codes," *IEE Proceedings Communications*, vol. 152, pp. 1062–1068, 2005.
- [8] S. A. Aly, Z. Kong, and E. Soljanin, "Fountain codes based distributed storage algorithms for large-scale wireless sensor networks," in *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN '08)*, pp. 171–182, St. Louis, Mo, USA, April 2008.
- [9] S. Kim, K. Ko, and S. Y. Chung, "Incremental Gaussian elimination decoding of raptor codes over BEC," *IEEE Communications Letters*, vol. 12, no. 4, pp. 307–309, 2008.
- [10] V. Bioglio, M. Grangetto, R. Gaeta, and M. Sereno, "On the fly Gaussian Elimination for LT codes," *IEEE Communications Letters*, vol. 13, no. 12, Article ID 5353274, pp. 953–955, 2009.
- [11] D. Burshtein and G. Miller, "An efficient maximum-likelihood decoding of LDPC codes over the binary erasure channel," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2837–2844, 2004.
- [12] W. Niu, Z. Xiao, M. Huang, J. Yu, and J. Hu, "An algorithm with high decoding success probability based on LT codes," in *Proceedings of the 9th International Symposium on Antennas Propagation and EM Theory (ISAPE '10)*, pp. 1047–1050, Guangzhou, China, November 2010.
- [13] H. Zhu, G. Li, and S. Feng, "BPL decoding algorithm of LT code," *Computer Science*, vol. 36, no. 10, pp. 77–81, 2009.

Research Article

A Mobile Computing Framework for Pervasive Adaptive Platforms

**Olivier Brousse,^{1,2,3} Jérémie Guillot,¹ Gilles Sassatelli,¹ Thierry Gil,¹
François Grize,² and Michel Robert¹**

¹LIRMM UMR 5506, Université Montpellier 2, CNRS, 161 Rue ADA, 34095 Montpellier Cedex 5, France

²Département des Systèmes d'Information, Faculté des Hautes Études Commerciales, Université de Lausanne, 1015 Lausanne, Switzerland

³LEAD-UMR 5022, Université de Bourgogne, CNRS, Pôle AAFE, Esplanade ERASME, BP 26513, 21065 Dijon Cedex, France

Correspondence should be addressed to Olivier Brousse, olivier.brousse@u-bourgogne.fr

Received 15 June 2011; Accepted 16 September 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Olivier Brousse et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ubiquitous computing is now the new computing trend, such systems that interact with their environment require self-adaptability. Bioinspiration is a natural candidate to provide the capability to handle complex and changing scenarios. This paper presents a programming framework dedicated to pervasive platforms programming. This bioinspired and agent-oriented framework has been developed within the frame of the PERPLEXUS European project that is intended to provide support for bioinspiration-driven system adaptability. This framework enables the platform to adapt itself to application requirements at high-level while using hardware acceleration at node level. The resulting programming solution has been used to program three collaborative robotic applications in which robots learn tasks and evolve for achieving a better adaptation to their environment.

1. Introduction

Pervasive computing has been gaining attention due to the emergence of a number of ubiquitous applications where context awareness is of importance. Examples of such applications range from ad-hoc networks of mobile terminals such as mobile phones to sensor networks systems aimed at monitoring geographical or seismic activity. All these systems involve (i) monitoring and processing collectively environmental and platform information, (ii) adapting to time-changing scenarios.

Considering the similarity between living organisms and the adaptability needs of such platforms, drawing inspiration from biology appears a natural solution. Although a number of techniques such as genetic algorithms or artificial neural networks exist, pervasive computing opens a new dimension of opportunities for further extending bioinspiration.

There exist several theories that relate to life, its origins, and all its associated characteristics. It is, however, usually considered that life relies on three essential mechanisms that

are phylogenesis, ontogenesis, and epigenesis (referred to as respectively P, O, and E throughout this paper):

- (i) Phylogenesis is the origin and evolution of a set of species. Evolution gears species toward a better adaptation of individuals to their environment; genetic algorithms are inspired from this very principle of life.
- (ii) Ontogenesis describes the origin and the development of an organism from the fertilized egg to its mature form. Biological processes like healing and fault tolerance are ontogenetic processes.
- (iii) Epigenesis refers to features that are not related to the underlying DNA sequence of an organism. Learning as of performed by artificial neural networks (ANN) is a process which scope remains limited to an individual lifetime and, therefore, is epigenetic.

The Perplexus European project [1] (that last from september 2006 to march 2010) aimed at developing a platform of ubiquitous computing elements that communicate

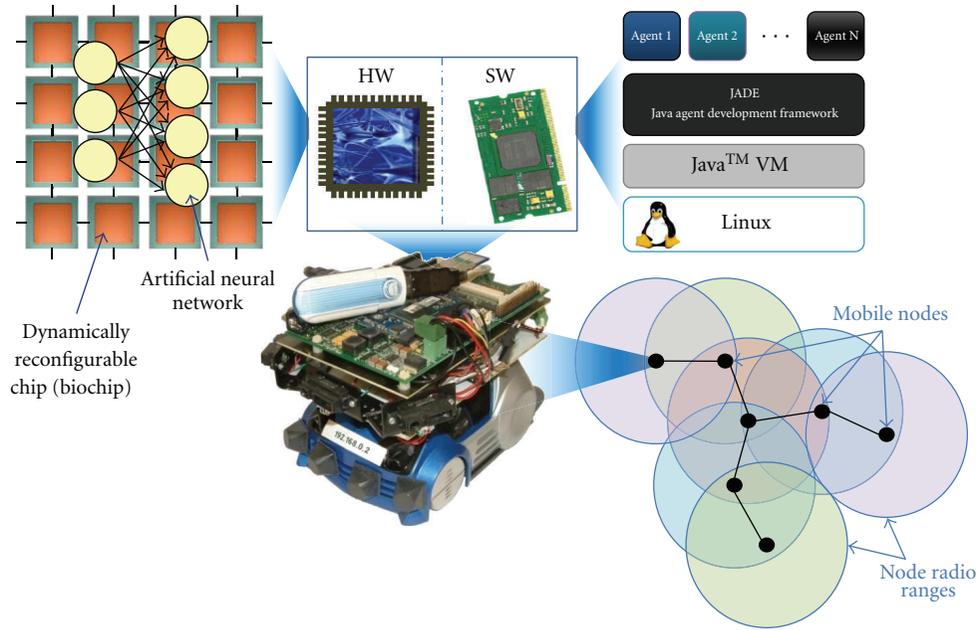


FIGURE 1: A pervasive sensor network example.

wirelessly and rely on these three principles of life. Intended objectives range from the simulation of complex phenomena such as culture dissemination [2] or biologically plausible neural networks [3] to the exploration of bioinspiration-driven system adaptation in ubiquitous platforms. As a consequence, this research dedicated platform has been developed keeping to explore the impact of bioinspired features in this context of omnipresent and present throughout computing. As a consequence, no particular efforts have been made to secure the platform as it is held in a research laboratory. In case a future evolution of the platform would aim to be disseminated on a larger scale, several solutions may be envisaged to secure it. As we use standardize technologies, Wifi may be ciphered, additional network and communications ciphering may be used such as VPN, SSL authentication, and JADE-S services. Another consequence of this restricted deployment stands at the power consumption level. Once again no specific effort has been made to limit the power consumption of the modules. As a matter of fact at the exception of the collaborative robotic applications presented in this paper, all platform modules are used plugged to power grid.

Each ubiquitous computing module (called Ubidule) is made of a XScale [4] microprocessor that runs an embedded Linux operating system and a bioinspired reconfigurable device that essentially serves the purpose of running artificial neural networks (ANNs). This device is referred to as Ubi-chip [5] (Ubidule Chip) throughout this paper. The Ubi-chip supports two main operating modes:

- (i) a native mode in which the chip behaves similarly to a FPGA [6], however, endowed with bioinspired features like automatic partial reconfiguration and self routing,
- (ii) a single instruction on multiple data (SIMD) processor mode [7] that allows parallel computation of algo-

rithms like neural network. This mode is presented in more details in the following.

Finally, ubidules are equipped with a wireless network adapter for internode communications, as well as sensors and actuators on an application-specific basis as illustrated in Figure 1.

This paper presents two contributions

- (i) First, a generic agent-based infrastructure that provides native support for bioinspiration dedicated to pervasive distributed platforms is described. This bioinspired programming framework based on agent oriented programming allows synchronizing population-level mechanisms (evolution through distributed genetic algorithms) and node-level mechanisms (learning processes using the reconfigurable device).
- (ii) Secondly, a means for transparently taking advantage of the Ubichip in SIMD mode is presented. This chip mode being dedicated to neural network simulations, it proves well suited to run learning mechanisms at the node level. The presented technique relies on a specific compiler that translates entire agent code sections into hardware executable binaries that speed up the execution.

The adaptability that results from these two contributions is demonstrated on three applications that use a fleet of autonomous vehicles; a lap race that uses Phylogenesis (evolution), an obstacle avoidance application that relies on collaborative learning as our first generation of applications and finally a robotic society evolution application that re-groups phylogenetic and epigenetic aspects of the previous applications.

This paper is organized as follows:

- (i) Section 2 describes the bioinspired agent programming framework used to the specification of pervasive and adaptive applications,
- (ii) Section 3 details the techniques used for enabling the use of the reconfigurable bioinspired device which is at the heart of the Ubidule,
- (iii) Section 4 presents the 3 applications in which Ubidules are embedded into small autonomous vehicles that learn and evolve,
- (iv) Section 5 concludes on this work and draws some perspectives for future work.

2. Adaptive Mobile Computing Environment

The modular structure of the Perplexus platform offers scalability thanks to the decentralized network structure which avoids central bottlenecks. Modules are then connected to each other using point-to-point and infrastructureless connections. In the case of the Perplexus project applications, the network reactivity and reliability are important criterions. We estimate that in most Perplexus applications this latency should not exceed 10 seconds for communication reliability and performance as well as for buffer memory reasons. Networking in ad-hoc platforms constitutes a challenge because of the topology of the network that does not rely on a fixed structure with routers, DHCP, or DNS servers. This challenge becomes critical when nodes are mobile. Indeed, it induces the need of distributed adaptive features at the platform/network level.

2.1. Network Support. The emergence of smart mobile devices able to manage network-based applications and the associated ad-hoc network support shares the same challenge with the Perplexus platform. In the literature such a paradigm is known as MANET for Mobile Ad-Hoc NETWORK [8]. This internet engineering task force (IETF) working group is in charge of proposing software solutions and standardizing IP routing protocols in the scope of wireless ad-hoc routing with either static or dynamic topologies.

MANET routing algorithms can be classified into two families.

- (i) *Reactive MANET Protocols (RMPs)* that search for a route between nodes A and B when a communication is requested. AODV (for ad-hoc on-demand distance vector) [9] and DSR (for dynamic source routing) [10] are reactive protocols. Once a route has been found, communications are directly established until the topology of the network changes which results in the computation of a new route.
- (ii) *Proactive MANET Protocols (PMPs)* in which nodes regularly exchange messages in order to maintain routes up to date and elect relay nodes. Optimized link state routing (OLSR) [11] is a proactive protocol complying with this principle. These protocols exhibit better performance and also prove more

power consuming because of the constant route updating process.

Critical points for the routing scheme in the Perplexus applications are communication reliability and latency. Proactive protocols that offer a better latency/power consumption tradeoff when nodes are moving are well suited for our platforms.

The OLSR routing protocol is among the most popular and effective MANET solutions [12, 13]. This proactive routing protocol regularly sends 3 different types of messages to create and maintain automatically network routes (i.e., in a proactive way). This mechanism is well suited for most mobile applications as it provides reduced communication latency due to mostly up-to-date routes. This is all the more true in comparison to reactive protocols that are slower to establish a route before actually communicating meaningful data.

The chosen OLSR implementation offers the possibility to set the number of desired relay nodes or to use plugins to provide additional services such as name/address translation or link quality routing [14]. Additionally three references present studies about the OLSR power consumption and/or propose energy-efficient versions of this algorithm [15–17]. Using one of these versions in conjunction with the nameservice plugin of OLSRd, the communication power consumption may be reduced significantly. This has not been done but it will be investigated for the next generation of the Perplexus platform.

2.1.1. OLSR Validation on Perplexus Platform. For validating this solution, we conducted several experiments which confirmed that proactive protocols such as OLSR perform better with respect to latency. Figure 2 shows the experimental protocol we used for OLSR. The map of the premises shows four nodes, three being static and the last one (Ubidule 3) in motion along the path (illustrated by the plain arrow).

As suggested in Figures 2 and 3, different network topologies are observed as node3 moves along the path. The changes from one to another occur whenever a node drops out or comes in the radio range of another. Results presented in Figure 4 correspond to a representative experiment. In this experiment, we set the number of relays to 1 and disabled the link-quality routing to observe OLSR with the lightest solution in real conditions without any optimization.

Figure 4 shows the evolution of the communications bit-rates received by the mobile node from the three other units; it can be clearly seen that a change in the network topology results in a break in one or more communication flows that lasts up to 5 seconds (dark arrows). In the case where packets are transiting through a relay node, they are not lost but temporarily stored in the relay and sent when communication is restored (light arrows). These results show that the OLSR protocol is compliant with the Perplexus platform. Results presented in Figure 4 also fulfill the latency constraint we estimated for Perplexus applications.

We consider these results satisfactory for the targeted applications; furthermore, the flexibility of the chosen OLSR implementation allows using a nameservice (DNS-like)

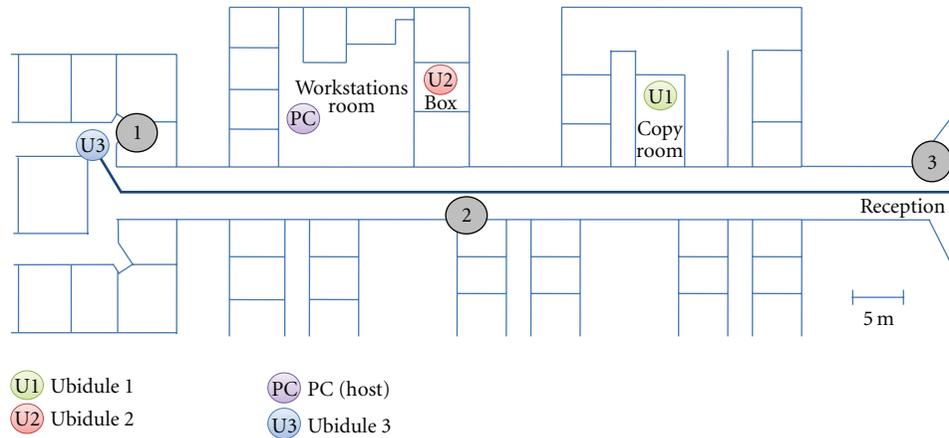


FIGURE 2: Mobile test protocol.

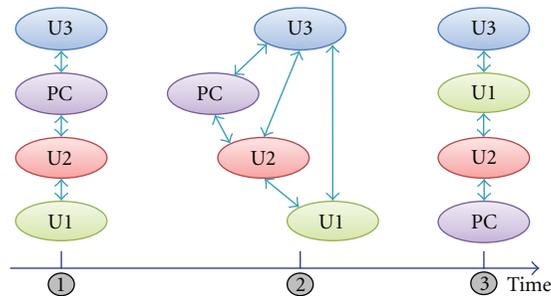


FIGURE 3: Time changing topology.

plug-in that proves mandatory in the following. The name-service plug-in acts in two successive steps:

- (1) the plug-in uses OLSR to broadcast messages containing IP and hostname information,
- (2) It collects other nameservice message and stores received data in the hostsIP file (i.e., /etc./hosts for Linux OS).

Consequently, OLSR with nameservice allows each module to get a local routing table working with IP addresses and hostnames and make the ad-hoc network act as a standard structured network. OLSR and a slightly modified name-service plug-in prove to be sufficient in term of network reactivity and efficient solution, in the case of our MANET application, as this solution perfectly fits our latency and performance needs and does not incur significant processing workload.

2.2. The FIPA Multiagent System. Programming distributed/pervasive applications are often regarded as a challenging task that requires a proper programming model capable of adequately capturing the specifications. Agent-oriented programming (AOP) derives from the initial theory of agent orientation which was first proposed by Shoham [18]. Agent-orientation was initially defined for promoting a social view of computing and finds natural applications in areas such as artificial intelligence or modeling of social

behaviors. AOP consists in making agents interact with each other through typed messages of different natures: agents may be informing, requesting, offering, accepting, and rejecting requests, services, or any other type of information. AOP furthermore sets constraints on the parameters defining the state of the agent (beliefs, commitments, and choices).

These constraints essentially define the agent oriented computational system which is then viewed as a set of communicating software modules that exhibit a certain degree of independence making the whole system more adaptive than an object oriented (OOP) computational system. These characteristics naturally geared the Perplexus modeling framework toward AOP as a solution for adaptability in our pervasive architecture.

2.2.1. FIPA-Based Agents. The IEEE group named Foundation for Intelligent and Physical Agents (IEEE-FIPA) defines standards allowing for interoperability among various multi-agent platforms. Figure 5 shows the FIPA standard structure of an agent platform (AP). Three main services ensure FIPA platforms reliability and functionality.

The agent management system (AMS) is in charge of the life cycle of platform agents; it can create, suspend, resume, or kill agents. The AMS also provides a white page service listing all agents “living” on the platform.

The directory facilitator (DF) is in charge of providing a yellow pages service. This service associates an agent to its offered services and a service to agents that provide it.

The message transport system (MTS) provides all communication functionalities at low-level. Therefore, agents can communicate with each other regardless their location (same or different APs).

Figure 5 shows that FIPA mandatory agents (i.e., AMS and DF) reside at the agent level next to user agents; they therefore behave as such and provide the above-mentioned functionalities. On the contrary, the message transport system lies at a lower level dedicated to communication protocols that provide a framework for interagent message communications. The FIPA standard does not include an AP search service that allows to discover FIPA peer APs as of

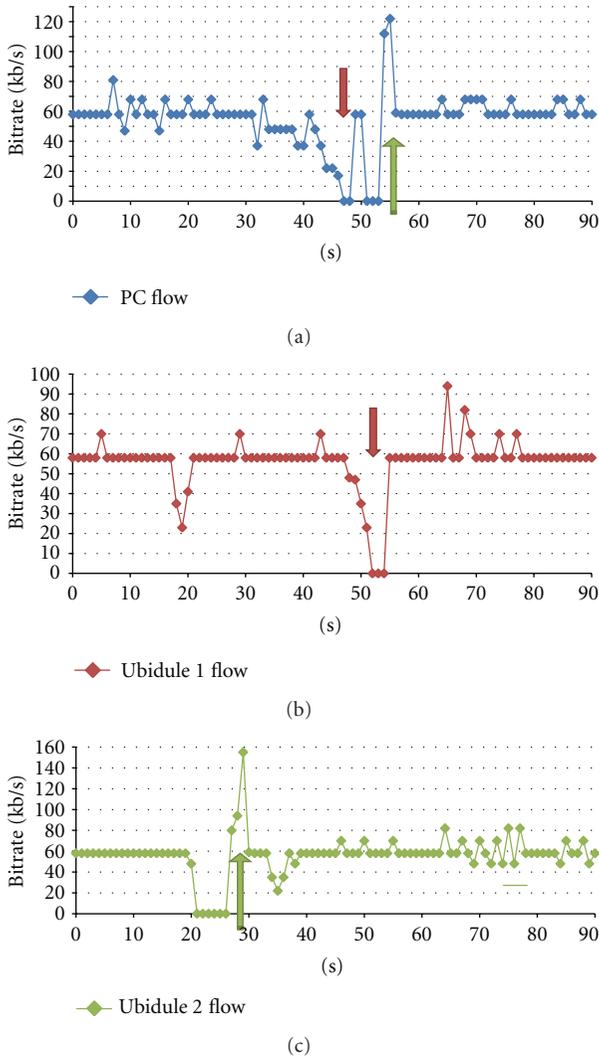


FIGURE 4: Mobile test: mobile node received message flows.

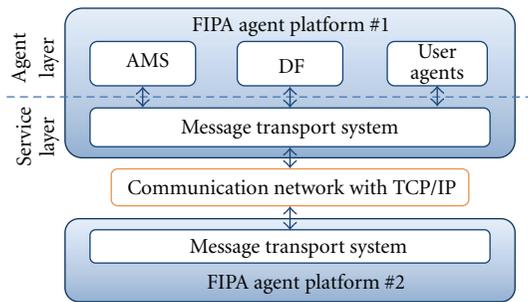


FIGURE 5: FIPA standard overview.

today. This feature exists in several protocols such as JXTA [19] or Kademia [20] peer-to-peer protocols. This drawback of the standard does not ease the use of multiple platforms which is targeted in this work. In this paper, we propose a solution to tackle this problem, detailed in Section 2.3 and finally allow an FIPA compliant AP to search for peer platforms.

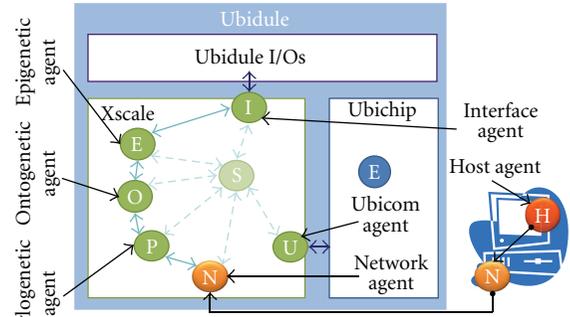


FIGURE 6: Ubidule programmed using BAF agents.

2.2.2. *JADE: Java Agent Development Framework.* There exists various multiagent platforms that allow developing agent-based applications, such as JADE [21], JXTA [19], FIPA OS [22], JAX [23], or MADKIT [24].

As the Ubicule XScale is a resource-limited embedded processor [25], many of the listed solution cannot be efficiently implemented. JADE was chosen for its portability (Java), FIPA compliance, and also because of the availability of a lightweight version called lightweight extended agent platform (LEAP) already used on embedded devices [26].

Agents in a JADE Framework “live” in containers. These containers exist either inside or outside of the original hardware hosting the JADE platform but are registered in the AMS and DF platform agents in an “original hosted main container.”

Communications take place using a message transport protocol (MTP) which in turn uses TCP/IP protocols. In our case, we decided to use HTTP MTP in order to ease AP communications and unify AP name and address with hardware hostname. This point is discussed with more details in the following section.

2.3. *Contribution 1: Bio-Mimetic Agent Framework.* Previously described solutions at network level (OLSR) and programming level (LEAP) are used as basis for a bio-inspired agent framework (BAF) suitable for distributed, decentralized, and mobile platforms where adaptability is mandatory. This section focuses on two fundamental aspects of the proposed BAF: on one hand the description of the BAF and overview of the provided functionality, on the other the description of the POE specific agents.

Bioinspiration and the three fundamentals of life being at the heart of the project, the proposed framework extends JADE default platform, that is, mandatory agents (AMS and DF) by defining agents whose purpose relate to both interfacing and bio-inspired mechanisms support as well as pervasive computing platform management agents. Figure 6 schematically depicts the ubidule programming which is regarded as a mixed hardware/software entity: the Ubichip for hardware support and the XScale microprocessor for software side.

The BAF specifies 7 agents belonging to 2 families:

- (i) application agents: phylogenetic, ontogenetic, and epigenetic agent(s),

- (ii) infrastructure agents: UbiCom, interface, network, and spy agent(s).

All these 7 agents have been developed using LEAP classes to support a dedicated function. Therefore, they add the BAF mandatory features to the legacy JADE. Figure 6 shows both the infrastructure and application agents and their interactions (for the sake of clarity AMS and DF agents are not represented).

- (i) P agent: the Phylogenetic agent is responsible of the execution of the distributed genetic algorithms: it calculates the local fitness of the individual (the local ubidule) and synchronizes this information with all other ubidules. It is responsible for triggering the death (end of a generation) and birth of the embodied individual hosted on the Ubidule.
- (ii) O agent: the ontogenetic agent is tightly coupled to the P agent: it takes orders from him and has the capability of creating other software agents.
- (iii) E agent: the Epigenetic agent embodies the individual and its behavior: it is either a software or hardware neural network.

Next to the three POE agents, four additional agents have been defined for interfacing and networking purposes.

- (i) I agent: the interface agent provides a set of methods for issuing commands to the actuators or retrieving data from the sensors of the ubidule.
- (ii) U agent: the UbiCom agent provides software API-like access to the Ubichip and manages hardware communications with the chip.
- (iii) S agent: the spy agent provides information on the platform state (agent status/results, activity traces, bug notification).
- (iv) N agent: the network agent provides a collection of methods for network-related aspects: time-synchronizing of data among ubidules, setting/getting clusters of ubidules, obtaining the list of neighbors, and so forth. For it requires access to low-level network-topology information, it also implements the MANET functionalities.

Finally, a host agent (H agent) instantiated on a workstation allows controlling remotely the Perplexus platform (Start/Stop/Schedule actions) through a graphical user interface.

Figure 7 shows the modifications applied to the FIPA platform for integrating the platform agents listed above. For the sake of clarity, only the P, O, E, and N agents are represented; all other agents reside on the agent layer with AMS and DF agents.

The additional features of the BAF (shaded areas) comprise the network agent and a low level service layer that handles the ad-hoc networking features. This layer includes OLSR and the nameservice plug-in. The hostname/IP table (periodically updated by the nameservice) can easily be accessed by other software entities such as JADE agents. Any

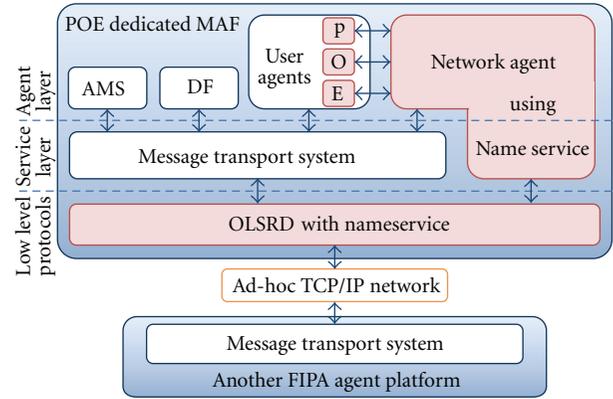


FIGURE 7: POE-dedicated BAF overview: white areas represent the classical JADE framework. BAF additional features to this framework appear in shaded areas.

agent can access these services through a specific Network agent.

The network agent is mandatory in a BAF platform. It allows FIPA platforms to communicate with each other and ensures the overall platform reliability. As this particular agent provides AP level services and low-level functionality (such as message broadcasting), it spans both highest level layers of the diagram. The use of the HTTP message transport protocol allows resolving the AP name and address in an ad-hoc network environment. Figure 8 describes this peer discovery mechanism.

Once the nameservice has edited the operating system Hostname/IP file (step 1), the network agent is able to create the peer platform list (step 2). Similarly, other agent lists can be created.

The host agent has been designed to provide a single interface for the platform management. This agent is able to remotely schedule applications from a host station thanks to the network agent services. A broadcast protocol is used for issuing global commands to the platform such as *global Service Search*, *Start Application*, *Stop Application*, or *Switch Mode* (switching from software mode to hardware accelerated mode, detailed later in Section 3.7). In this case, network agents sink command messages.

The main advantage of this method is that the host agent, and the user it represents, does not need to know addresses of all final receivers at design time allowing users to take advantage of the flexibility and scalability of the environment.

3. Hardware Acceleration

Bio-inspired features are heavy computational tasks that hardly fit with embedded devices such as the XScale processor used within the PERPLEXUS platform. The Ubichip has been designed to provide hardware support for such features. Figure 9 puts focus on the SIMD operating mode of the Ubichip used to accelerate parallel parts of PERPLEXUS applications.

In this specific mode, the ubicells are grouped by four to obtain an array of 16 bits processing elements (PEs)

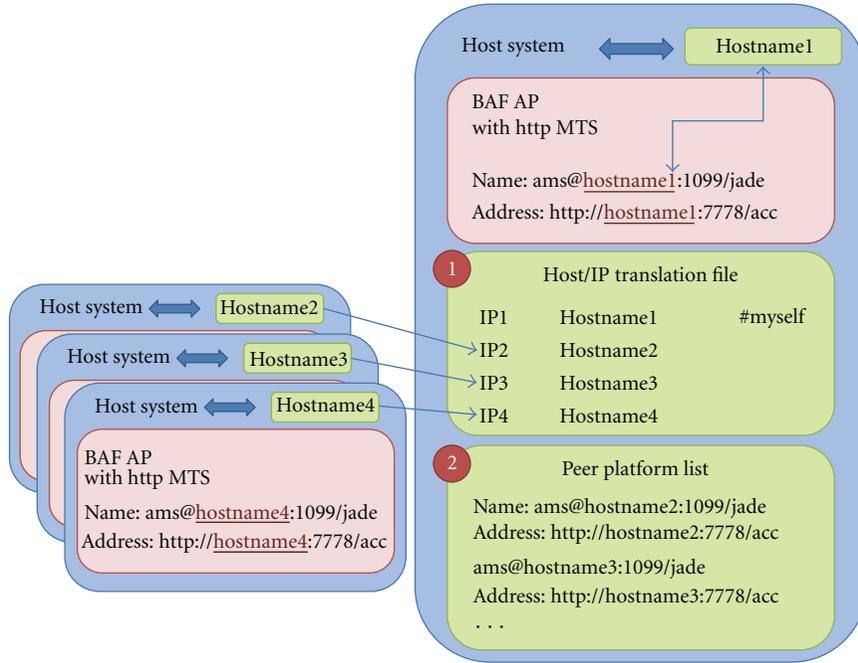


FIGURE 8: BAF AP address resolution.

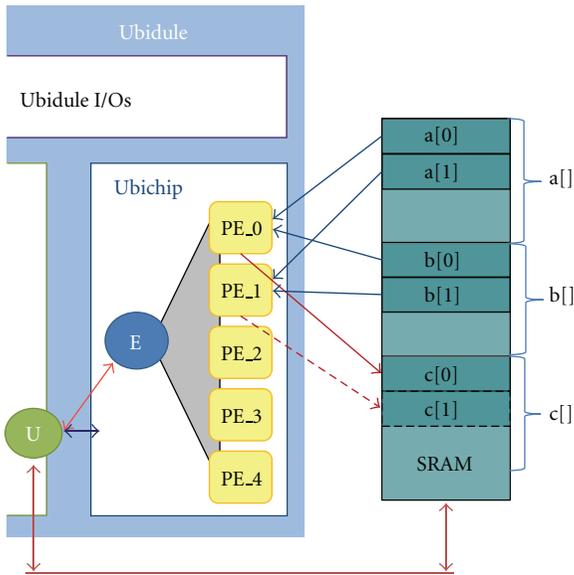


FIGURE 9: Ubichip SIMD mode architecture.

controlled by a sequencer. Program and data are stored in an external memory accessible by the sequencer and the XScale. The XScale is in charge of enabling or disabling the Ubichip allowing it to access the memory in a secured way. When the program ends, the Ubichip is able to interrupt the XScale allowing a proper result retrieval.

In this section, we present a solution to program this accelerator with the same programming language for both purely software and hardware accelerated agents.

3.1. Related Works. As agents in the BAF are captured in Java, we investigated the literature to find parallel oriented Java implementations. However, a classical Java virtual Machine (JVM) is by construction, executed on a single processor. Some Java hardware machines have been under study over the last decade, [27, 28] or [29], but none of them provide support for hardware parallelism as the original language was not intended to this.

Another approach is proposed by Manta [30] and relies on compiling Java threads to native x86 assembly code and run them on an x86 cluster through remote method Invocation (RMI). This solution removes the Java portability and does not target platforms such as the SIMD processor of the Ubichip.

Some software parallel classes like JPCL [31] add software parallelism to Java but JVMs are running on a single processor. Therefore, using this kind of libraries requires a framework that links several JVM running on several hardware targets and sharing the same global object space. The previously presented BAF ensures a similar software parallelism level but based on message passing scheme rather than shared memory.

The proposal is to provide a solution for easing the accelerator programming and consequently use a real- and fine-grain hardware acceleration in the PERPLEXUS framework.

3.2. Contribution 2: The Jubi Extensions. The fundamental concept behind the proposed approach relies on the use of directives for flagging parallel sections in a hardware-independent description based on Java: Java for ubiquitous or Jubi in short. Agent coded in Jubi can then be executed in

SW mode (in such case directives are ignored) or in hybrid SW/HW where flagged sections are compiled for parallel SIMD execution.

A flagged section of code presented as a component can be described with its inputs, outputs, and internal behavior. Adding this approach to Java requires setting firstly `in` and `out` keywords. An `NPE` keyword allows the user to specify the number of processing elements (PEs) that will be used for this application.

The following code where $c = f(a, b)$ gives an example of the applied transformations:

```
final int NPE = 4;
int a[NPE], b[NPE];
int c[NPE]
```

becomes

```
final int NPE = 4;
in int a[NPE], b[NPE];
out int c[NPE]
```

Then, to describe the behavior of the hardware block, we define the `#jubi` keyword that flags the code to be accelerated using the SIMD hardware. Finally, to enable the parallelization of software sequential loops in the hardware accelerated mode while keeping the sequential software execution possible, we introduce the `parallelfor` keyword. This keyword allows both software and hardware generating implementations from the same unified description.

The following code performs an addition on input vectors and illustrates the memory layout presented in Figure 9:

```
#jubi
parallelfor(int i=0; i<NPE; i++)
{
    c [i] = a[i] + b[i];
}.
```

3.3. Specific Tools. Figure 10 presents the compilation flow we propose in order to allow fast applications development in software or in hybrid HW/SW modes. In this figure, the software compilation flow appears on the left side whereas the hardware flow is on the right side.

Software applications execution only use software flow whereas hardware-accelerated applications require both sides to compile accelerated agents software part (named envelope) and hardware parts (named kernels). The agent envelope is a part of the Java file that triggers the hardware execution and feeds hardware kernels with appropriate data. This is done through the UbiCom agent that acts as a wrapper between sequential and parallel sections of the application code (i.e., software and hardware parts).

As presented in Figure 10, the processing of the Jubi file results in the creation of two distinct file types. One Java file that offers the possibility to start the application either in software mode or in hardware accelerated mode. For each `#jubi` block described in the Jubi file, one “Ubi”

file is created. Every Ubi file encapsulates the code to be accelerated. Figure 11 details the splitting process that produces both the Ubi files and Java files with the required hardware calls through the UbiCom agent.

The Java file is compiled thanks to the standard Java compiler (`javac`), whereas Ubi files are compiled into associated hardware kernels, “.hw” files by the JubiCompiler and UbiAssembler. The last step of the compilation flow is the loading of the HW-accelerated code from a “.hw” file into the program memory of the Ubichip with up-to-date data. This is done at runtime by the UbiCom agent behavior.

3.4. JubiSplitter. The entry point of the JubiTool compiling environment is the JubiSplitter that splits the Jubi description into a Java description for the software part and several Ubi descriptions for the hardware-accelerated parts. The JubiSplitter tool generates “softwareBehaviour” and “hardwareBehaviour” classes. These JADE Behavior classes represent respectively the entire agent functionality in SW mode and the envelope of hardware accelerated kernels, which contains non-parallelizable (non `#jubi` flagged) code sections in the HW mode. The execution mode is then chosen when the platform is configured. Figure 11 gives an example of this code splitting stage which is the first stage of the application.

3.5. JubiCompiler. Once a Jubi code has been split into Java and Ubi kernels the JubiCompiler tool compiles every Ubi code into Ubichip assembly. As a result, the following example, where uninitialized values are set to 0:

```
int a [NPE] = { 4,3,1,2 } ;
int b [NPE] = { 1,2,3,4 } ;
int c [NPE] ;
c = a + b
```

is then compiled into
.data

```
V01 = ‘ ‘00030004’ ’, ‘ ‘00020001’ ’
V02 = ‘ ‘00020001’ ’, ‘ ‘00040003’ ’
V03 = ‘ ‘00000000’ ’, ‘ ‘00000000’ ’
```

.code

```
load    r1,V01
load    r2,V02
mov     r1
add     r2
mov     r3
store   r3,V03.
```

Where the `r1`, `r2`, and `r3` represent the register of the Ubichip, `V01` corresponds to `a`, `V02` corresponds to `b` and `V03` corresponds to `c`.

The JubiCompiler is based on a flex and bison [32] description of the Jubi grammar. An array dimension set to `NPE` indicates that every processing element of the SIMD

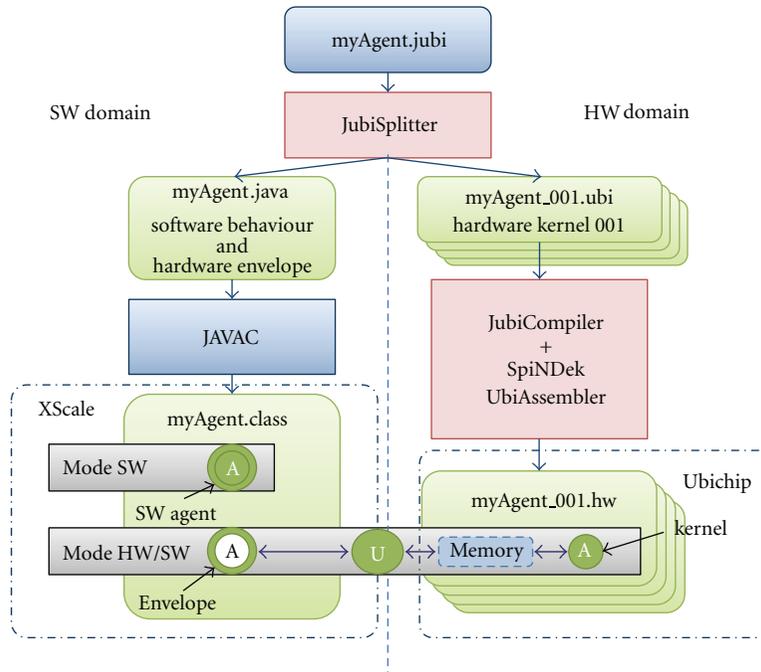


FIGURE 10: Unified compilation flow.

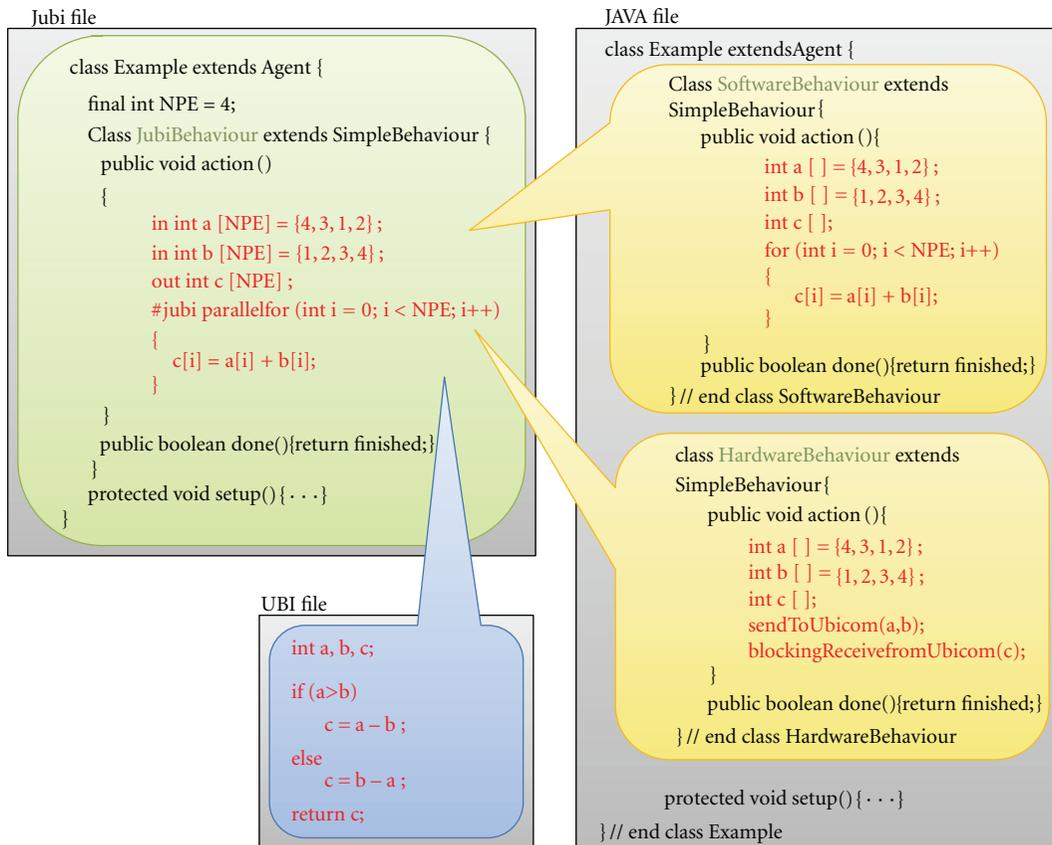


FIGURE 11: JubiSplitter: Code splitting and file creation.

processor will receive a different dataset. 1D arrays therefore become PE-local scalar variables. Hence, a 1D NPE-relative variable (in SW) is spanned over NPE PEs of the Ubichip architecture as NPE scalar variables (in HW).

3.6. UbiAssembler. The UbiAssembler is the final tool used in the flow. It is in charge of translating the assembly code generated by the JubiCompiler into Ubichip SIMD hardware binary executables. This tool is part of the SpiNDeK environment [33] and was developed to program the Ubichip using assembly. Two main features are provided in this tool in addition to the code translation: the memory layout setup that involves variable to physical address translation, and regular linking provided by label to physical address translation.

3.7. Toward an Adaptive Acceleration. The opportunity of running agents in software or hybrid hardware/software mode opens interesting perspectives in term of adaptability. Beyond enabling to assess the speedup resulting from hardware execution, this allows for online mapping of agents Ubi sections to the Ubichip. This may prove useful for adapting to changing performance requirements by migrating agents from hardware to software and the other way around.

The UbiCom agent has been designed for this purpose; it embeds Ubichip management functions and an interface that allows an agent to request a migration of a functionality (#jubi flagged sections) to the Ubichip. The U agent is able to start and stop the Ubichip and to load a binary file in the chip code memory. Then, after the loading phase the UbiCom waits until an interrupt is raised by the Ubichip to get data back and communicate them to the agent envelope.

The features presented in this section have been validated in VHDL simulations of the Ubichip model as the prototype was not available at writing time. Test programs that validate PERPLEXUS applications needed features have been compiled and fed into the Ubichip model as memory content files. The result assertion of these test programs proves the functionality of the proposed framework.

4. BAF Applications Validations

4.1. Introduction. Three case studies are presented in this section for, respectively, illustrating evolution and learning features of the proposed framework. These applications are not taking advantage of the hardware acceleration due to delays in the fabrication of the Ubichip accelerator.

Figure 12 schematically depicts the used robots, their sensors and actuators, as well as the framework agents presented previously.

The E agent is the main robot controller that reads data from the sensors and, depending on given or learned rules, issues commands to the wheel motors. The P agent is responsible of the robot controller evolution and therefore computes the robot fitness and runs the genetic algorithms together with the P agents of other robots. Ontogenetic agent instantiates the E agent based on the genome provided by the P agent. The N and I agents serve the purpose explained previously. The U agent is unused here as the chip is still

under fabrication; therefore, presented applications only make use of software mode.

The use of either all these agents or only a subset of them is a design decision.

4.2. Test Application 1. In order to prove the reliability of the platform, a simple proof-of-concept application based on a race has been developed. This application relies on all framework agents; the robot controller (E agent) being here a simple feed-forward artificial neural network (ANN) that reads binary information from three proximity sensors installed on the front, front-left, and front-right sides of the robots, and issues speed commands to the two motors. For this application, robots are moving into a closed arena containing obstacles and a start/finish line. The goal of robots is to run one lap.

Figure 13 shows the principle of this genetic race the lap time gives the fitness of a given robot controller and hence its genome which is the array of ANN weights. Therefore, there is no learning in this application, changes in the robot behavior being driven by evolution only.

These agents are crossed and/or mutated by the P agent to create the next generation replacing inadequate behaviors. Once a new individual genome is ready, the P agent forwards it to the O agent that instantiates the corresponding E agent. Generation after generation, robots exhibit better behaviors proving the reliability of the software and the possibility to handle POE problems via the platform.

A demonstration video that illustrates this application is available online at: http://www.lirmm.fr/ADAC/?page_id=9.

4.3. Test Application 2. Robots which participate use online learning (Epigenesis) for improving their performance. Figure 14 shows the robots that are enclosed in an arena scattered with obstacles (cylinders in Figure 14); collision avoidance is here the main objective. As this application only targets learning, the P and O agents are not used here.

The collaborative learning approach has been implemented in such a way that two networks are trained online in parallel (i.e., two individuals are running). Every-time a robot gets into an obstacle, its own network is trained to avoid this error in the future. The faulty robot also advises the other individual that the action it has just made, in the context it was, leads to an error. The other robot is then learning its proper network including this information. This scheme is repeated until robots are able to avoid obstacles exchanging their experience in live as represented in Figure 14.

Besides the previously described sensors, a bumper switch is added to inform the robot whenever a collision with an object occurs; it is located on the front side of the robot. The sensors here do not provide binary information but rather the distance with the nearest obstacle.

These robots move by issuing speed commands to each of the two motors. As depicted on Figure 14, an ANN is in charge of controlling the robot. Figure 15 depicts the principle of this application: the E agent is a multilayer perceptron ANN that uses a standard back-propagation learning algorithm.

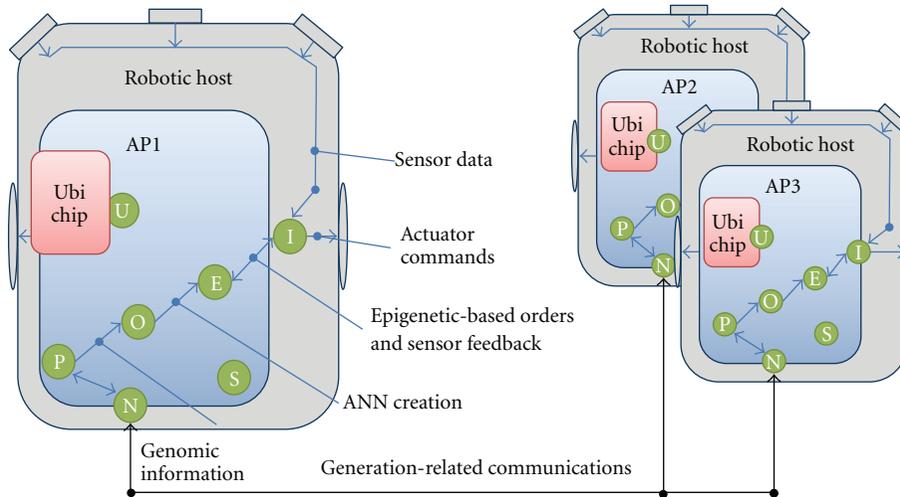


FIGURE 12: Application environment.

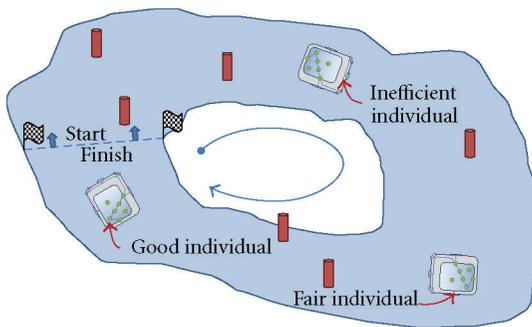


FIGURE 13: Evolution-driven application overview.

TABLE 1: Ultrasonic sensors areas.

Area	Distance range
0	0–200 mm
1	200–400 mm
2	400–600 mm
3	600–800 mm
4	>800 mm

Inputs of the ANN are the three values measured by the infrared and the ultrasonic sensors, we have defined five areas for each ultrasonic sensor as depicted in Table 1.

The outputs of the ANN are speed values sent to the motors, each value is set as an integer value from -7 (i.e., fast backward motion) to $+7$ (i.e., fast forward motion). The robot can turn by applying two different speeds on the motors.

During a given period, each robot performs the following tasks.

Robots are moving in an unknown environment. Each time they collide into an obstacle, a random modification of the relevant learning pattern is applied and an ANN learning phase is triggered online. The robot then notifies all its peers

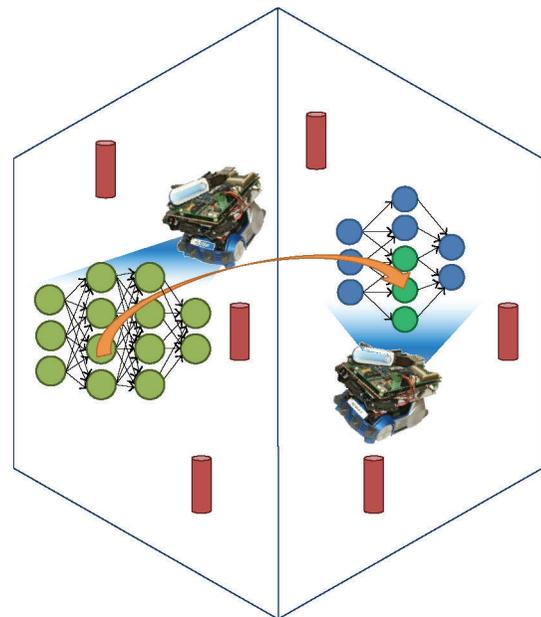


FIGURE 14: Collaborative learning application overview.

that this pattern shall be modified; and the modification is registered by all robots therefore collectively speeding up the convergence toward a satisfying solution.

In this application, a host system that runs the H agent (host agent) is used for launching the application and to collect information throughout the execution of the algorithm.

As depicted in Figure 16, the host workstation and all the ubidules are running the BAF environment.

Our experiments show that this technique exhibits a speedup (versus a single robot) that is almost linear with the number of used robots. Furthermore, it has been observed that a convergence threshold is reached after a number of iterations which is a function of the complexity of the environment. Once this threshold is reached, adding some

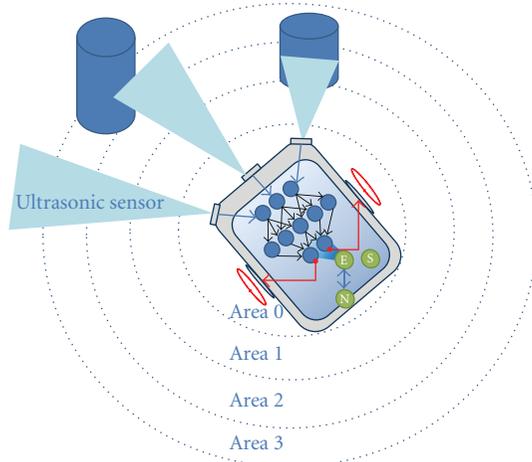


FIGURE 15: Collaborative learning application overview.

more obstacles in the arena re-triggers learning until a new threshold is reached, demonstrating the adaptability potential of the proposed solution.

A video showing the runs of the above-presented individuals is available online at: [ifundefinedselectfont http://www.lirmm.fr/ADAC/?page_id=9](http://www.lirmm.fr/ADAC/?page_id=9).

4.4. Evolving a Population of Learning Individuals. Based on the same idea to prove that bio-inspired features are useful for distributed and pervasive systems adaptability, the third application relies on the association of P and E features. We designed it as a mixed evolution and learning robotic demonstrator. In this application, the robot behavior is set using a dynamically changing quality function created following the individual genome. This function is based on couples state/action and is modified using reinforcement learning [34]. As a consequence, robots are following move rules that depend on the present sensors state and actions are rewarded if the action is a success or punished if a collision occurred. The application runtime is described in Figure 17 and can be explained with the following steps.

- (i) Robots move in the obstacle-scattered arena of Figure 14 learning from their errors during 2 minutes. When they collide with an obstacle, learning is triggered inducing a change in the quality function. To avoid wall sticking, we also make the robot to move backward on collision. Due to the limited time and various uncertainties induced by their genome, some individuals are not able to learn properly to avoid obstacle whereas good individuals are learning rapidly.
- (ii) The individual fitness is then calculated using the number of collisions balanced with the global-recorded speed of the robot during the 2 minutes run.
- (iii) The new generation is then created merging and mutating the genomes of individuals with the best fitness.

- (iv) The simulation ends when an individual achieves to spend an entire run without colliding with any obstacles.

In this application, robots inherit their characteristics such as right/left and front sensors zones from their respective parents. Robots are also improving themselves using a collision-triggered process that allows online learning. In Figure 17 blue part represents step of the application where individuals interact with each others using the BAF communications features.

One additional aspect we introduced in this application is the PE cooperation effect called the parental education. We define the Parental Education as a merging of the following two aspects: innate for newborn inherited behavior characteristics and acquired for the transmitted knowledge.

- (i) The first is the innate aspect that can be encountered in some species. This process similar to instinct allows newborn individual to walk within minutes, this is only possible because their parents and the whole species acquired this innate ability.
- (ii) The second is the parent influence on children representing the childhood learning in the nature, parents of evolved species like humans are teaching their children.

To mimic these PE aspects, we chose to transmit a given percentage (between 20 and 50%) of the parent quality function to the child genome.

One of the characteristics of the reinforcement algorithm is the reflex latency value. It corresponds to the delay between a given move and its associated reward or punishment. This characteristic can easily be used as a genome parameter and then evolve with the species.

The individual genome is represented in Figure 18 with the three main transmitted characteristics namely ultrasonic sensor zone definition, parental education patterns and reflex latency value.

Using parental knowledge rapidly brings robots to an average species-level behavior that online learning further improves. Subsequent offspring will, therefore, benefit from species capabilities evolution through inheritance.

In this application, the quality function is stored in an array where every value of quality is associated with the corresponding state/action couple. Following quality function examples, are extracted from application results with three possible move forward (MF), turn Left (TL), and turn Right (TR) actions per sensor state. Three sensing zones are used leading to $3^3 = 27$ possible sensors states, for the sake of simplicity only some of the short-length and wide-range zones quality functions are exposed in Table 2. Presented values are obtained after the first generation run.

The quality function used in this application is based on the action score. The action with the highest quality is used depending on the current sensor state. These actions are represented for the two first sensor states with shaded cells. If this action provokes an error (i.e., the robot get into an obstacle), the action score is lowered. On the contrary, if the action is successful its score is raised. Table 2 shows two final

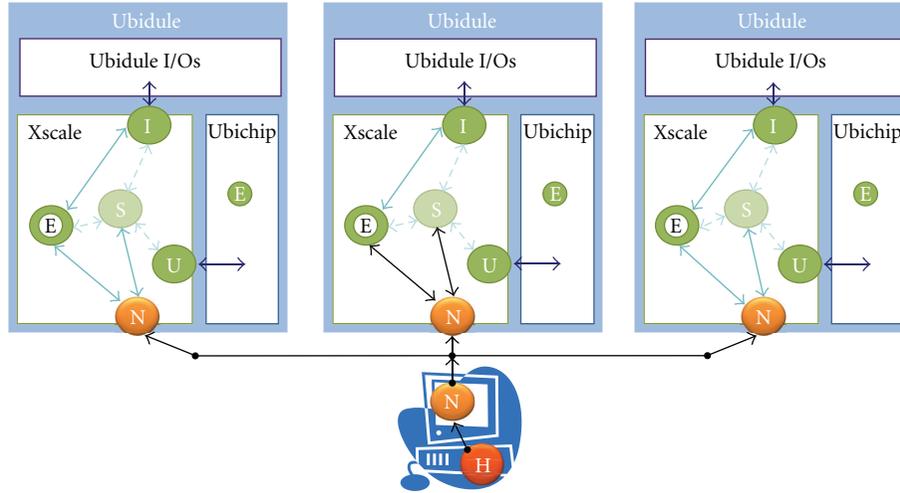


FIGURE 16: Application involved agents.

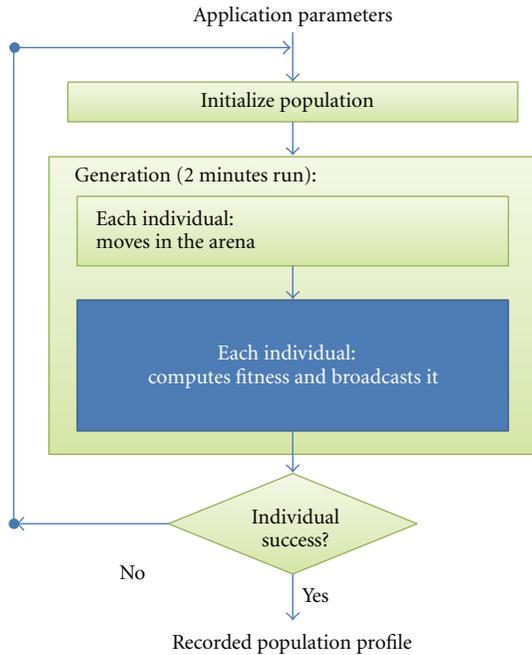


FIGURE 17: Second-generation application steps.

quality functions for different generation individuals. This demonstrates the generation after generation evolution that occurs even if 20% to 50% of the individual self-experience is transferred to children.

The fitness of an individual is defined in (1) where α is the reflex coefficient normalizer, S_{mean} the mean value of motion speed, and B_i the collision penalty of the i th move:

$$\text{Fitness} = \alpha \times \left(\sum_{i=1}^n S_{\text{mean}} \times (1 - B_i) \right). \quad (1)$$

With this fitness computation rule, we promote forward, moving individuals that avoid collision taking into account

TABLE 2: Quality function example.

L/F/R sensor zones	Action	gen 1 QF	gen 4 QF
0 0 0	MF	-4.0	-4.5
	TL	-5.625	-4.5
	TR	-5.0	-2.0
0 0 1	MF	-1.0	-2.0
	TL	-0.0	0.0
	TR	0.0	0.0
0 0 2	MF	-4.0625	-4.0625
	TL	-4.0625	-4.0625
	TR	0.25	1.9921875
⋮	⋮	⋮	⋮
2 2 1	MF	9	9
	TL	0.0	0.0
	TR	0.0	0.0
2 2 2	MF	1.9960938	2.0
	TL	0.0	0.0
	TR	0.0	0.0

the number of moves during the 2 minutes run as well as their speed.

Table 2 shows resulting quality functions of individuals whose genomes main difference is their respective reflex latency value. The combining of this reflex value with the closest sensor zone reveals to be critical in various cases and had a great influence on the genome fitness. In the above-cited examples the reflex latencies were respectively, 185 ms and 195 ms resulting in respective fitness of 744 pts and 674 pts. One generation later, the recorded fitness of two of their children are 776 pts and 744 pts using the same reflex latency differentiator.

This demonstration shows the efficiency of a PE-based application in the robotic field, and the faculty of the proposed framework to run advanced bioinspired applications. Our experiments show that even if every generation is

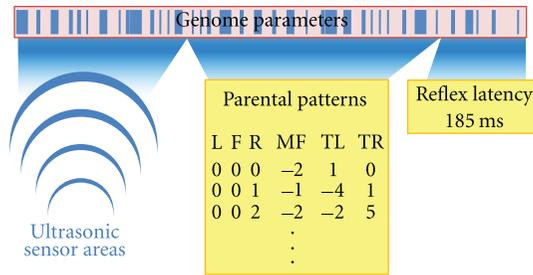


FIGURE 18: Robot genome example.

running only 2 minutes, the ability to provide information on the desired behavior to the next generation (PE effect) brings an interesting improvement compared with the classical P based application presented in Section 4.2.

A video showing the runs of the above presented individuals is available online at: http://www.lirmm.fr/ADAC/?page_id=9.

5. Conclusion

This paper presents a bioinspired agent-oriented framework dedicated to the prototyping of adaptive pervasive applications. Furthermore, this solution provides a means for taking advantage of hardware acceleration thanks to the use of language extensions associated with a specific compiler that generates code for the chip developed within the confines of the Perplexus European project.

The proposed proof-of-concept applications suggest that bioinspiration brings advantages for achieving adaptability in pervasive applications. To this end, dedicated robots with improved sensory capabilities are currently under fabrication. These robots will furthermore have the capability of hot swapping their depleted batteries autonomously thanks to a dedicated docking station, therefore, enabling to setup experiments lasting days or weeks.

Within the frame of the project, ongoing work focuses on demonstrating the combined advantages of the developed framework along with the bio-inspired device on a fleet composed of several tenths of robots running over long periods.

Although the adaptability features have been demonstrated on robotic applications, we believe that other application areas may benefit from the proposed solution. May it be for scheduling of communications for optimizing power in a sensor network, or devising techniques for transmitting data collected by distributed nodes to a gateway, the dependence to the environment makes such adaptive solutions attractive for coping with non deterministic scenarios.

Acknowledgments

This project is funded by the Future and Emerging Technologies Program IST-STREP of the European Community, under Grant IST-034632 (PERPLEXUS). The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not

responsible for any use that might be made of data appearing in this publication.

References

- [1] Perplexus, Pervasive computing platform for modeling complex virtually-unbounded systems, 2009, <http://www.perplexus.org/>.
- [2] J. Peña, O. Jorand, H. Volken, and A. Pérez-Urbe, "A connectionist, embodied and situated agent-based approach for studying the dissemination of culture," CESABM, UNIL.
- [3] O. Chibirova, J. Iglesias, V. Shaposhnyk, and A. E. P. Villa, "Dynamics of firing patterns in evolvable hierarchically organized neural networks," *Lecture Notes in Computer Science*, vol. 5216, pp. 296–307, 2008.
- [4] Intel corp., "Intel xscale microarchitecture," Tech. Rep., 2000.
- [5] Y. Thoma and A. Upegui, "Specification of bio-inspired features to be supported by the device," hEIG-VD, Yverdon, Switzerland, Internal Report, 2006.
- [6] A. Upegui, Y. Thoma, E. Sanchez, A. Perez-Urbe, J. M. Moreno, and J. Madrenas, "The perplexus bio-inspired chip," in *Proceedings of the 2nd NASA/ESA Conference on Adaptive Hardware and Systems(AHS '07)*, IEEE Computer Society, 2007.
- [7] J. M. Moreno, "Specification of the ubicell," Tech. Rep., Barcelona, Spain, UPC, Internal Report, 2006.
- [8] IETFMANET work group, "Mobile Ad-Hoc networks (MANET)," April 2009, <http://www.ietf.org/html.charters/manet-charter.html>.
- [9] C. E. Perkins and E. M. Royer, "Ad-Hoc on-demand distance vector," December 1998.
- [10] D. Johnson and D. Maltz, "The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4," February 2007.
- [11] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, *Optimized Link State Routing Protocol for Ad-Hoc Networks*, INRIA Roquencourt, HiPERCOM project, 2001.
- [12] T. Clausen, P. Jacquet, and L. Viennot, *Comparative study of CBR and TCP performance of MANET routing protocols*, Workshop MESAINRIA Roquencourt, HiPERCOM project.
- [13] A. Huhtonen, "Comparing AODV and OLSR routing protocols," Seminar on Internetworking.
- [14] A. Tønnesen, *Impementing and extending the optimized link state routing protocol*, Tech. Rep., M.S. thesis, UniK University Graduate Center University of Oslo, 2004.
- [15] F. de Rango, M. Fotino, and S. Marano, "Ee-olsr: energy efficient olsr routing protocol for mobile ad-hoc networks," in *Proceedings of the Military Communications Conference (MILCOM '08)*, San Diego, Calif, USA, November 2008.
- [16] F. D. Rango, J. C. Cano, M. Fotino, C. Calafate, P. Manzoni, and S. Marano, "OLSR vs DSR: a comparative analysis of proactive and reactive mechanisms from an energetic point of view in wireless ad hoc networks," *Computer Communications*, vol. 31, no. 16, pp. 3843–3854, 2008.
- [17] C. Taddia, A. Giovanardi, and G. Mazzini, "Energy efficiency in OLSR protocol," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON '06)*, pp. 792–796, Reston, Va, USA, September 2006.
- [18] Y. Shoham, "Agent-oriented programming," *Journal of Artificial Intelligence*, vol. 60, no. 1, pp. 123–129, 1996.
- [19] L. Gong, "Jxta: a network programming environment," *Internet Computing Online*, vol. 5, pp. 88–95, 2002.

- [20] The XLattice Project, “Kademlia: a design specification,” Tech. Rep., The XLattice Project, 2003, <http://xlattice.sourceforge.net/components/protocol/kademlia/specs.html>.
- [21] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*, Wiley Series in Agent Technology, Wiley, 2007.
- [22] FIPA-OS project, FIPA-OS Agent Toolkit, FIPA-OS project , 2007, <http://sourceforge.net/projects/fipa-os>.
- [23] F. Strauss, J. Schönwälder, and S. Mertens, Jax—a java agent x subagent toolkit, July 2000.
- [24] G. Nguyen, T. Dang, L. Hluchy, M. Laclavik, Z. Balogh, and I. Budinska, “Agent platform evaluation and comparison,” Slovak Academy of Sciences, Institute of Informatics, Pellucid 5FP IST-2001-34519, 2002.
- [25] Wikipedia, Xscale, <http://en.wikipedia.org/wiki/XScale#PXA-27x>.
- [26] J. Lawrence, LEAP into Ad-Hoc Networks, ACM Workshop on Agents in Ubiquitous and Wearable Computing, AAMAS.
- [27] M. Schoeberl, *Evaluation of a Java Processor*, Vienna University of Technology.
- [28] D. Hardin, “aj-100: a low-power java processor,” Embedded Processor Forum.
- [29] ARM, “Jazelle—arm architecture extention for java applications,” white paper, 2002.
- [30] J. Maassen, T. Kielmann, and H. E. Bal, “Parallel application experience with replicated method invocation,” *Concurrency Computation Practice and Experience*, vol. 13, no. 8-9, pp. 681–712, 2001.
- [31] T. Brecht, H. S., M. Shan, and J. Talbot, “Paraweb: towards world-wide supercomputing,” in *Proceedings of the European Symposium on Operating System Principles*, pp. 181–186, 1996.
- [32] GNU.org, “The gnu operating system,” June 2009, <http://www.gnu.org/software/bison/>.
- [33] M. Hauptvogel, J. Madrenas, and J. M. Moreno, “Spindek: an integrated design tool for the multiprocessor emulation of complex bioinspired spiking neural networks, submitted to congress on evolutionary computation,” IEEE CEC, 2009.
- [34] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: a survey,” *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

Research Article

Analysis of Mobility and Sharing of WSNs By IP Applications

Dennis J. A. Bijwaard,^{1,2} Paul J. M. Havinga,^{2,3} and Henk Eertink⁴

¹ *Inertia Technology, Offenbachlaan 2, 7522 JT Enschede, The Netherlands*

² *Ambient Systems, Colosseum 15d, 7521 PV Enschede, The Netherlands*

³ *Pervasive Systems Research Group, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands*

⁴ *Novay, Brouwerijstraat 1, 7523 XC Enschede, The Netherlands*

Correspondence should be addressed to Dennis J. A. Bijwaard, dennis@inertia-technology.com

Received 15 July 2011; Revised 6 October 2011; Accepted 13 October 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Dennis J. A. Bijwaard et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Movement of wireless sensor and actuator networks, and of nodes between WSNs are becoming more commonplace. However, enabling remote usage of sensory data in multiple applications, remote configuration, and actuation is still a big challenge. The purpose of this paper is to analyse and describe which mobility support can best be used in different scenarios, and how shared usage of mobile WSNs by multiple IP applications can best be scaled up. This paper describes logistic and person monitoring scenarios, where different types of movements take place. These mobility types and their implications are categorized and analysed. Different degrees of support for these mobility types are analysed in the context of the mobility scenarios. Additionally, different schemes are analysed for shared use of mobile WSNs by multiple applications. In conclusion, guidelines are provided for dealing with mobile and overlapping WSNs and the most promising scheme for shared use of mobile WSNs by IP applications.

1. Introduction

In this paper we analyse the mobility and sharing of internet-enabled wireless sensor and actuator networks (WSNs) by applications. Example applications are remote monitoring of goods that are transported between warehouses, monitoring of persons with health-related problems, and remotely controlling lights or motors (actuation). We focus on the following WSNs types (based on the taxonomy presented in [1]) where mobility and sharing of sensor data can be a concern.

(i) *Body Sensor Network (BSN)*. BSNs are sensor networks consisting of few wireless sensor nodes on or around a living being's body connected to a more powerful device such as a smart phone. Monitoring of vital signs, tracking, and data collection have been the main objectives of these sensor networks. Interaction with sensor-enabled objects [2], such as a dumbbell or ball, is an interesting upcoming usage area. BSNs are small-scale, use different types of sensors, and are usually limited to single-hop wireless communication. Due to the fact that various types of personal information can be collected by these networks, both security and privacy are

major concerns. Reliable data processing and timely feedback are of high importance. Applications using the sensor data can run on the mobile phone or on a server on the internet (e.g., via connectivity provided by general packet radio service (GPRS) or universal mobile telecommunications system (UMTS)).

(ii) *Structure Sensor Network (SSN)*. SSNs consist of medium to large numbers of wireless nodes usually attached to buildings (e.g., office), structures (e.g., bridges), and infrastructure (e.g., rails) or deployed in specific venues (industrial sites). SSNs may be deployed both indoors and outdoors. Wireless nodes can also be attached to objects moving inside the structure and between structures. SSNs usually extend their wireless coverage with multiple hops of wireless communication and often use a variety of sensors.

(iii) *Vehicle Sensor Network (VSN)*. The sensor data from within a moving vehicle (e.g., a car, boat, train, and plane) can also be transferred wirelessly (e.g., via GPRS) to a central server, be monitored remotely and/or merged with data from

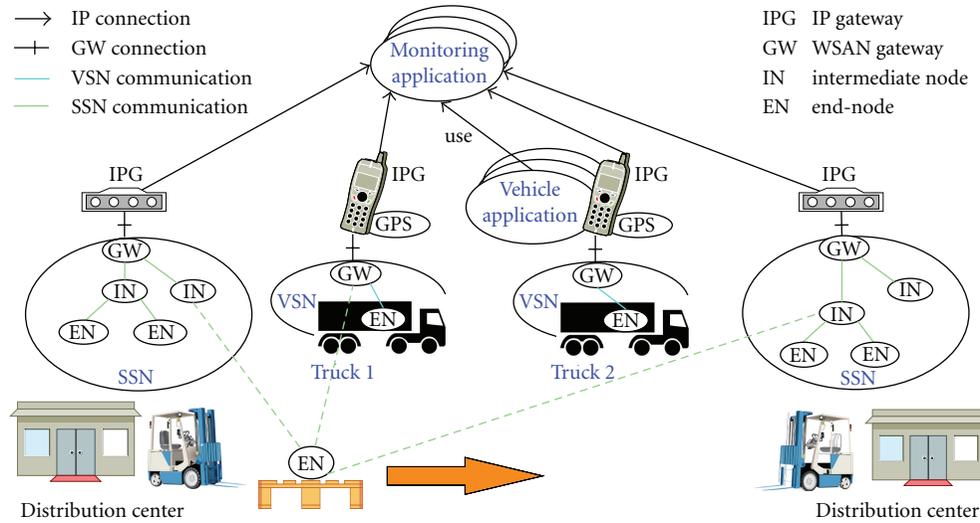


FIGURE 1: Monitoring moving goods in logistics.

other sensor networks. In warehouse logistics, VSNs are often used together with SSNs, for example, when monitored goods are transported in a truck from one warehouse to the other.

Other WSAN types are the environmental sensor network (ESN) that monitors conditions in the environment, the transport sensor network (TSN) that contains both VSN and wirelessly interconnected vehicles, and the participatory sensor network (PSN) consisting of smart phones with embedded sensors.

We analyse the different movements that can take place in and across WSANs. Furthermore, we analyse the movement of Internet-connected WSANs and applications that use them. These IP applications can use sensor information from the WSANs as well as configure and actuate the elements of individual nodes. The purpose of our analysis is to gain insight in the different types of mobility and to determine how they can best be supported in different usage scenarios. A lot of research has been done on mobility within WSANs (e.g., in [3–5]). However, in this paper, we focus on mobility issues of nodes that move between WSANs, WSANs that move in each other's range, and IP applications that use the sensor information. Additionally, this paper analyses ways to share multiple mobile WSANs in an IP application and how multiple IP applications can use the same WSANs. A number of issues related to shared WSAN usage were described by Shu et al. [6], and some solutions have been proposed for sharing WSANs [7, 8]. The purpose of the analysis of shared WSANs usage is to determine which sharing scheme can best be used with different numbers of attached IP applications, where both applications and WSANs can be mobile.

This paper is organized as follows. In Section 2, these WSAN types are used in mobility scenarios where IP application(s) use the WSANs. In Section 3, the types of mobility related to WSANs and IP applications are further detailed, and the consequences of these mobility types are analysed. Section 4 further analyses how to support these

mobility types in the scenarios. Section 5 describes and analyses different schemes for handling sharing of mobile WSANs. The article concludes with the most promising scheme to be used for shared mobile WSANs.

2. Mobility Scenarios

WSANs can bring clear benefits to large-scale enterprise systems by delegating part of the business functionality closer to the point of action [9]. Healthcare, wellbeing, and sport-related person monitoring with WSANs is another area that gains research attention [10]. We have defined four scenarios where different types of mobility take place when nodes, complete WSANs, or IP applications using the sensor data are moving. Two scenarios are described where a truck with monitored goods moves between distribution centres and two where a monitored person moves around. For both trucks and monitored persons, an IP application can run on the internet or be directly attached to the WSAN while using information from another IP application running on the internet. Both a smartphone and router can be the IP gateway (IPG) for WSANs and applications.

2.1. Moving Vehicle Sensor Network. In this scenario, goods are tagged [11] with a sensor node. This sensor node travels with it when it moves with a truck between distribution centres. The trucks have a VSN deployed, and the distribution centres have an SSN deployed, see Figure 1. All sensor data, including global positioning system (GPS) location, are provided to the monitoring application. The VSN in truck 1 may lose its connection to the monitoring application when travelling through low-coverage areas (e.g., tunnels), and the IPG will roam to other GPRS network providers when going abroad. The monitoring application would typically offer realtime insight in the conditions of the goods, both when in storage and during transit. Based on condition deterioration, the truck could be rerouted to a closer-by destination.

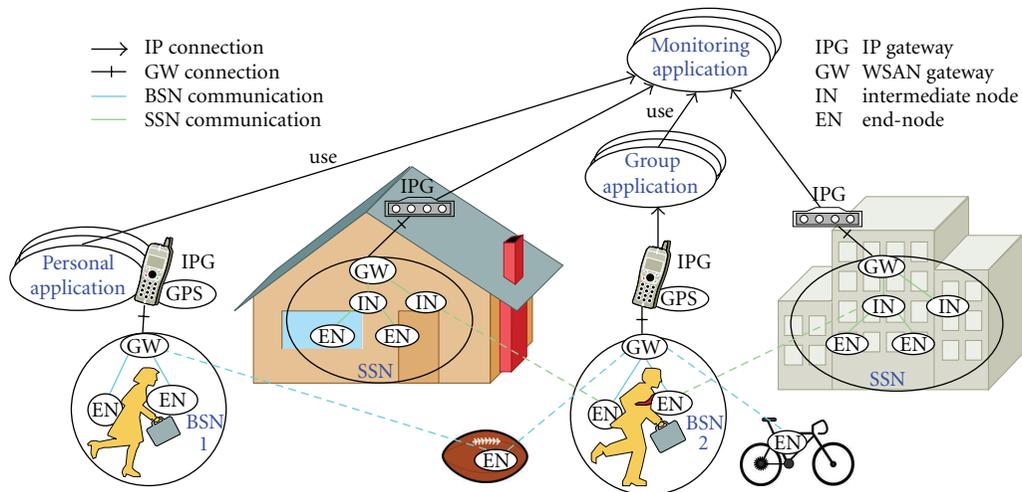


FIGURE 2: Moving BSN and personal applications.

2.2. Moving Vehicle Application. In this scenario, truck 2 in Figure 1 will have a GPRS connection to the Internet, and the vehicle application may lose connection to the monitoring application when travelling through low-coverage areas, and the IPG will roam to other network providers when going abroad. An example vehicle application could be monitoring the condition of goods in the truck and comparing the measurements with the inventory list to see if nothing is lost, misplaced, or spoiled. Via the monitoring application, the vehicle application could check historic conditions of the goods and location of missing goods or replacements.

2.3. Moving Body Sensor Network. In this scenario, a man with BSN 2 and smartphone moves between two houses with WiFi coverage and deployed SSN. The man uses objects that have sensor nodes attached that are compatible with the BSN. The BSN is used by a group application running remotely on the Internet (e.g., monitoring health status and location and may use other monitoring applications), see Figure 2. The smartphone will use the cheapest available Internet connection for communication to the Internet, such as WiFi.

2.4. Moving Personal Application. In this scenario, a woman with BSN 1 and smartphone moves between two houses with WiFi coverage and deployed SSN and uses sensor information from these SSN nodes. The BSN is used by a personal application running on the smartphone that she carries, see Figure 2. The smartphone will use the cheapest available Internet connection for obtaining measurements from a monitoring application. This monitoring application provides real-time sensor information from buildings based on GPS location.

3. Analysis of Mobility Types

Since WSAN nodes and its gateway can be attached to different moving objects, multiple types of mobility can occur within and across WSANs. Additionally, a device that

hosts an IP application using the sensor data can move. A wireless node can be an endnode that is usually equipped with sensors and/or actuators or an intermediate node that can extend the coverage area of the WSANs.

This paper makes a distinction between the following WSAN nodes: the *gateway* that makes it available to applications, *intermediate nodes* that extend the coverage of the WSAN gateway, and *endnodes* that can connect to the intermediate nodes or gateway. Although the paper assumes that the endnodes do not change to intermediate nodes (like in the Ambient WSANs [12]), most of the mobility types described also apply when they do (such as with the Collection Tree Protocol (CTP) [13]). In the CTP, an endnode can join the WSANs via another endnode, turning the latter into an intermediate node.

The *wireless resources* used by a WSAN are characterised by one or more radio channels and the type of radio transmission. Example radio transmission types are probabilistic such as in carrier sense multiple access (CSMA), using timeslots such as in-time division multiple access (TDMA), and frequency hopping such as used in Bluetooth. Different WSANs are defined to be *compatible* when nodes of the WSANs can communicate with both gateways: they use the same WSAN protocol; use the same wireless resources; share the same encryption key.

We distinguish the following types of mobility related to WSANs.

- (i) A moving IPG: network mobility takes place when the IPG starts using another wireless or wired network technology or starts using a different network provider on the same network technology. The implication of this change is that the Internet Protocol (IP) address of the IPG changes which will break connections when there is no transparent mobility support (like mobile IP (MIP) [14, 15]) in place. For short-lived connections like via HTTP, this connection break will result in a time-out. Movement can also make the IPG unreachable when there is no

TABLE 1: Mobility consequences for compatible WSANs and their nodes (data link layer).

		WSAN			Static interm. node		endnode	
		Associated	In	Out	In	Out	In	Out
Moving	WSAN	—	Reallocate	Ok	Option	Lost	Option	Lost
	Interm. node	Yes	Alternative	Lost	Alternative	Lost/alternative	Option	Lost/alternative
		No	Option	Lost	Option			
	endnode	Yes	Alternative	Lost	Alternative	Lost/alternative		
No		Option		Option				

network coverage or when it moves into a private or protected network. The moving IPG affects

- (a) an attached WSAN: the IPG provides the WSAN with Internet connectivity for applications that want to use information from, configure, or actuate nodes in the WSAN. Examples are moving BSNs and VSNs. The implication of movement can be (un)reachability and (dis)connection of IP applications,
 - (b) an attached IP application: an IP application can use sensor data from nearby or remote WSANs via TCP/IP. The IPG movement can break existing connections from the IP application to the WSAN and make others possible.
- (ii) A moving WSAN: a WSAN gateway that may have associated nodes. When the WSAN moves in range of another WSAN, matching wireless resources may require changing these resources in one of the WSANs to avoid bandwidth degradation and possible collisions. When compatible WSANs move in range, the nodes may associate to both of them. When a compatible WSAN moves in range of an intermediate or endnode, that node may join the WSAN. When the WSAN moves out of range of an associated intermediate or endnode, the association will be lost.
- (iii) A moving intermediate node (with or without connected nodes)
- (a) Within a WSAN. For instance, an intermediate node attached to a forklift can extend the radio coverage of the WSAN in the direction it moves and allow endnodes to communicate. When this intermediate node moves in range of a compatible WSAN gateway or an other intermediate node, it has the option to join that WSAN; when it moves out of range, it will lose the connection when it was associated. When the intermediate node moves in range of a compatible endnode, that endnode may join the WSAN. When the intermediate node moves out of range of an associated endnode, the endnode will lose its association,
 - (b) Across WSANs. For instance, an intermediate node attached to a forklift moving between

the coverage areas of different compatible WSANs and picking up goods with attached endnode(s). The intermediate node can join the other WSAN when it is out of range of the other one and can choose the WSAN when it is in range of both. When it comes in range of another node, that node can choose to join it when it goes out of range of a node, that node will lose its association unless there is an alternative intermediate node or gateway in range.

(iv) A moving endnode

- (a) Within a WSAN. The node may have to communicate via different intermediate nodes depending on their radio coverage. When an endnode moves in range of a compatible WSAN or connected intermediate node, it can join it. When it moves out of range of such a WSAN, it will be disassociated. When it moves out of range of a compatible intermediate node, it will be disassociated unless there is an alternative intermediate node or gateway in range,
- (b) Across WSANs. For instance, an endnode that is placed with goods transported between WSAN-enabled distribution centres (see Section 2). When an endnode moves in range of a compatible WSAN or intermediate node, it can join it. When it moves out of range of such a WSAN, it will be disassociated. When it moves out of range of an intermediate node, it will be disassociated when there is no alternative in range.

Table 1 summarizes the mobility consequences on the data link layer when a WSAN, intermediate or endnode moves in or out of range of another compatible WSAN, intermediate, or endnode. Before this movement, the moving entity can be associated or not, for WSANs this does not apply. When WSANs come in each other's range, they may need to reallocate their wireless resources when they use the same ones. When an intermediate or endnode comes in range of another WSAN, it has the option to associate with that WSAN (denoted as "option") or choose it as an alternative link. When an intermediate or endnode moves out of range of an intermediate node, it may still have an alternative to use, else the association with the WSAN is lost.

TABLE 2: Mobility consequences for WSANs used by IP applications (network layer).

		Static IPG with attached				
		Connected	In	WSAN	Application	Out
				Out	In	Out
Moving IPG With attached	WSAN	Yes			Alternative	Lost/reroute
		No			Option	
	Application	Yes	Alternative	Lost/reroute		
		No	Option			

Table 2 summarizes what happens when an IPG with attached WSAN or IP application moves in or out of range of another IPG with attached IP application or WSAN, respectively. It assumes a bidirectional connection between the WSAN and the IP application. Moving in range here means that an IP connection becomes possible; moving out of range here means that the IP connection breaks (e.g., when no mobility protocol like MIP is in place and the IPG changes IP address). The moving IPG can have an attached IP application or WSAN that is connected or disconnected. When a connection breaks after moving, it may be reestablished by setting up an alternative route or it may be lost when this is not possible. When a connection becomes possible after moving, this is denoted as “option.”

3.1. Remarks on WSAN Mobility Types

- (i) Clearly, there are a number of options for connected nodes when another compatible WSAN comes in reach; how they deal with this can vary per WSAN type. In Section 4, we analyse this further for the given scenarios.
- (ii) Table 1 merely describes the case where compatible WSANs and its nodes are considered. When incompatible WSAN protocols, wireless resources, or encryption are used, nodes cannot use these links. The gateway may still need to reallocate resources when the other WSAN operates on the same channel. Section 4 analyses different ways to support overlapping WSANs.
- (iii) Without mobility support, complete WSANs and IP applications will disconnect when the IPG changes IP address. For seamless mobility, a number of mobility schemes can be used (described in Section 5).
- (iv) WSAN nodes can potentially listen to messages in each of the WSAN they become part of, so they can also transfer information from one WSAN to another. Section 4 describes how data protection can be provided.

4. Analysing the Mobility Scenarios

In this section, the mobility scenarios from Section 2 are analysed in the light of the different mobility types described

in Section 3 and the level of mobility support that can be offered.

Important properties for mobility support in the scenarios are the following

(i) *Security/Privacy.* Security of WSANs is a complex issue. Cryptographic credentials can be used to authenticate a node in a network and to encrypt the traffic; examples of these credentials are keys and passwords. Keys can be symmetric, where one key is used for both encryption and decryption or asymmetric, where a pair of keys is used for encryption and decryption. [16] provides a set of guidelines to handle security in WSANs, however asymmetric encryption becomes possible in WSANs [17].

(ii) *Interference.* Networks that use the same wireless resources can potentially interfere with each other. This interference can take different forms. When the WSAN protocols use timeslots, misalignment may cause collisions in two slots for every message, while timeslot alignment limits this to maximally one collision per message. When the WSANs use probabilistic Media Access Control (MAC) protocols, the chance for collisions will increase since there are more nodes. When a combination of timeslots and probabilistic MAC protocols are used, all timeslots are likely to suffer packet loss. Adaptive MAC protocols (like [18–20]) could be used to reduce TDMA interference.

(iii) *Overlap Awareness.* When a WSAN is aware of the presence of another WSAN, it can adapt itself accordingly. The first step to become aware is detecting an increase of interference. Next, a scan can be done to detect periodic traffic and silence on the radio channel. The detected periodic patterns can be used to adapt the WSAN traffic to reduce interference. Scanning can also be used to detect familiar WSAN types. When received messages can be decoded and are of nonregistered intermediate nodes, there is a good chance that a compatible WSAN is nearby.

(iv) *Wireless Resource Adaptation.* When a WSAN is aware of an overlapping WSAN, it can adapt its wireless resources to reduce interference. Examples of WSAN adaptation are channel change, synchronisation and distribution of timeslots between WSANs, turning off the gateway, and changing mode of operation (e.g., change from gateway to intermediate node).

(v) *WSAN Mobility*. What do nodes need to do to switch to another WSAN? Clearly, this depends greatly on the WSAN type, for instance, in the following cases.

- (a) In the Ambient WSAN [12], all nodes have a unique 6-byte MAC address. The endnodes (called SmartPoints) can send messages (using CSMA) when they have compatible network keys. The intermediate nodes (called MicroRouters) need the (symmetric) network key to announce themselves to the gateway and to get (TDMA) timeslots assigned.
- (b) In a IPv6 over low-power wireless personal access networks (6LoWPAN) network [21], the MAC address (2 upto 8 bytes) is used for node identification, and communication can be beacon less (pure CSMA) or beacon enabled (a hybrid of CSMA and TDMA). Nodes need to register themselves using the 6LoWPAN-customized neighbour discovery protocol, which makes a unique node address available in the WSAN and makes the WSAN network prefix available to the node. MIP can be used to make a node uniquely addressable when it moves between different WSANs. 6LoWPAN networks can utilize the symmetric keys of the 802.15.4 MAC.
- (c) In the Inertia WSAN [22], the endnodes have a 2-byte address assigned; this address is used in the registration message to the gateway to obtain a TDMA timeslot. There are no intermediary nodes, since this network is primarily targeted at small body area networks. Objects with a node attached can be used by multiple WSANs in sequence. The messages are not (yet) encrypted.
- (d) BSNs can also be constructed using Bluetooth which uses frequency hopping for radio transmission. Bluetooth is single hop (research is done on multihop scatternets) and usually uses a powerful device like a smartphone or PC as master. For switching to another network, the master of the other BSN needs to pair with the device and connect to one of its services. When pairing is done beforehand, the master could be programmed to autoconnect to a specific service, which would enable mobility of devices between masters.

(vi) *IP Mobility*. No transparent mobility scheme like MIP is considered in the scenario analysis; different IP mobility schemes for IP-enabled WSANs will be compared in Section 5.

(vii) *Costs*. Different wireless communication technologies have different associated costs. Using WiFi is generally cheaper or even free, while mobile data roaming via GPRS can vary from a relatively cheap data bundle to very costly when exceeding the bundle and when crossing nation borders. Internal WSAN communication is considered free of charge in this paper.

(viii) *Protocol Robustness*. Protocols that are not robust against foreign messaging will suffer most from interference. Methods to detect broken packages vary from a cyclic redundancy check (CRC) check to encryption where decryption is likely to fail for broken packets. Techniques like forward error correction can be used to add redundancy to the messages to be able to reconstruct some of the broken messages when there is interference.

(ix) *WSAN Compatibility* (as defined in Section 3).

4.1. Moving Vehicle Sensor Network. In order to get a complete measurement trace from the moment the sensor node comes out of storage in the first distribution centre until it arrives with the truck in the other distribution centre, measurements need to be merged at IP level in the monitoring application. In order to correctly correlate the measurements, an indication is required that the VSN gateway is in range of the SSN gateway. One indication is the fact that sensor nodes that were first reporting via the VSN start reporting via the SSN. Another indication is correlation of the GPS coordinates of the truck and the distribution centres. A third indication could be the detection of the SSN by the VSN gateway.

The most prominent changes that can occur when a VSN moves are the following

- (i) The VSN moves in range of the SSN and potentially other VSNs (i.e., other trucks). When the WSANs use the same radio channel, there can be interference. When the WSANs are compatible, nodes may report via the other WSAN.
- (ii) The VSN moves out of range of the SSN and potentially other VSNs. In this case, the nodes that remain in coverage of the VSN need to associate with it in order to transmit.
- (iii) The VSN moves in range of intermediate and endnodes. When these nodes are compatible with the VSN, they may associate with it.
- (iv) The VSN moves out of range of associated intermediate and endnodes. These nodes will no longer be able to transmit via the VSN so need to associate with the SSN or another VSN.
- (v) The IPG in the truck may connect to different IP networks, for example, when it moves from one country to the other. Additionally, Internet connectivity can be temporarily unavailable.
- (vi) The GPS coordinates of the truck and a distribution centre will differ when the truck is on the road and be similar when the truck is close by.

From the changes above, an issue becomes apparent when compatible WSANs come in range, that is, nodes that can report to both WSANs. The SSN should be capable to handle a few more nodes from the truck (since the nodes may go to storage anyway), however the VSN has a limited Internet connection and could have a harder time

with additional nodes. Moreover, the monitoring application would have a harder time distinguishing the additional nodes reporting to the VSN from the ones that are really being transported by that truck. Therefore, the following solutions are proposed to restrict this freedom of the nodes.

- (i) When a compatible WSAN is detected, the VSN gateway could be switched off. However, this could give problems when multiple VSNs are close together, since they may all decide to switch off. Furthermore, the nodes in the truck may not be able to reach the SSN from within the truck.
- (ii) When a compatible WSAN is detected, the VSN gateway could be switched to intermediate node mode, so that it extends the coverage of that WSAN. However, this will put more load on the SSN and there may be a limit to the number of supported intermediate nodes (e.g., 64 in the Ambient network).
- (iii) Without detection, the WSANs can be separated by using different network keys, and only the nodes that need to be mobile between the WSANs can have multiple keys (i.e., the nodes that go from storage to transport to another storage). The nodes can decide themselves when they start using the other network key for transmission, for example, switch to the SSN when the VSN link degrades.

Of course, also interference will be a concern for WSANs that use the same wireless resources. When using timeslots, this can partly be resolved by synchronizing and/or distributing timeslots. Alternatively, the VSN could be changed to use noninterfering wireless resources or different network key before it reaches the distribution centre, for instance by detecting similarity in GPS coordinates of the truck and distribution centre and consulting via the monitoring application what resources are used by the SSN.

Additionally, since the IPG can change its IP address, it will need to a mechanism to still report the measurements to the monitoring application. Obviously, the IPG could buffer measurements and send them after reconnecting to the monitoring application.

4.2. Moving Vehicle Application. The most prominent changes that can occur when a truck with vehicle application moves are the following.

- (i) The IPG may connect to different GPRS or UMTS networks and optionally other wireless networks like WiFi.
- (ii) IP connectivity of the IPG can be temporarily unavailable when there is bad or no wireless network coverage.

The implication of network attachment changes is often that the IP address of the IPG changes or becomes unavailable, which will break existing connections from the vehicle application or VSN to other IP applications on the Internet. When there is no connection, it will be impossible to connect to the monitoring application to fetch SSN measurements; in other cases, the connection needs to be reestablished.

Moreover, IP applications on the Internet that are using data from the vehicle application may be confronted with a changed IP address or unreachable IPG and associated connection breaks. The IP address of the IPG can be unreachable when not connected, when in a private area network, and when a restrictive firewall blocks the Internet traffic.

4.3. Moving Body Sensor Network. The most prominent changes that can occur when a BSN attached to a smartphone moves are:

- (i) The BSN moves in range of the SSN and potentially other BSNs (i.e., other persons). When the WSANs use the same radio channel, there can be interference. When the WSANs are compatible, nodes may report via the other WSAN.
- (ii) The BSN moves out of range of the SSN and potentially other BSNs. In this case, the nodes that remain in coverage of the BSN need to associate with it in order to transmit.
- (iii) The BSN moves in range of objects with endnodes. When these nodes are compatible with the BSN, they may associate with it and transmit their measurements.
- (iv) The BSN moves out of range of associated objects with endnodes. These nodes will no longer be able to transmit via the BSN.
- (v) The smartphone may connect to different wireless networks and Internet connectivity can be temporarily unavailable.
- (vi) The GPS coordinates of the smartphone and an SSN will differ when the person is out of range and be similar when he/she is close by.

The following mobility support options can be considered in this scenario. (Note that data protection is an important privacy aspect in BSNs.)

(1) *WiFi Usage.* Based on costs, the smartphone will have preference for WiFi to send BSN messages to the group application. instead of the more costly GPRS. Of course a new connections needs to be established to the group application. When multihoming is supported, the GPRS connection could be kept open while using WiFi. When moving out of WiFi range, GPRS will be used again and the WiFi connection to the application will break.

(2) *Secured Object Use.* since objects can potentially listen, store, and forward information, communication of more sensitive BSN sensor data should be encrypted.

(3) *Separate Uplink.* Since the BSN and SSN need to connect to different applications, they use a separate IP connection. The BSN should use encryption for privacy-sensitive messages, and its uplink should use encryption towards the application. Inter-BSN traffic is impractical for normal usage, so BSNs should use different encryption keys for privacy.

(4) *BSN Messages via Compatible SSN.* When BSN and SSN are compatible, BSN endnodes may use any intermediate SSN node or gateway to send their information upstream. The information could be encrypted such that only a specific application can decrypt it, for instance, by using the public key of the application for encrypting the message payload. The connection details for the destined application should somehow be conveyed to the IPG of the SSN gateway. This makes this a more customized and therefore less attractive option.

(5) *Dual-Stack BSN Endnodes.* They are endnodes that can communicate both with the SSN and incompatible BSN. This can also be used to send messages with encrypted payload upstream. Here, the BSN message destination also needs to be conveyed to the SSN gateway.

WiFi usage and encryption are a must for lowering communication costs and enhancing privacy. A separate IP uplink for the BSN and SSN messages is considered more practical than sending BSN messages via a compatible SSN.

4.4. Moving Personal Application. The most prominent changes that can occur when a person with personal application on a smartphone and attached BSN moves are the following.

- (i) The smartphone may connect to different wireless networks, and Internet connectivity can be temporarily unavailable. In case of WiFi, local access to the IPG of the SSN may become possible.
- (ii) The BSN can come in range of a SSN.
- (iii) The BSN can get out of range of the SSN.

The following options can be considered for a moving application (on a smartphone) that uses its attached BSN and nearby SSN data.

(1) *Intranet Access to SSN Data.* Local access to SSN data may be possible in the associated Intranet when the smartphone is allowed to use this network. The SSN needs to advertise itself in some manner to enable discovery by the smartphone application.

(2) *Public SSN Server.* The SSN sends its sensor data to a publicly reachable server on the Internet from which applications can fetch it when they have the proper credentials. Retrieval could, for example, be based on the current GPS coordinates of the smartphone.

(3) *Direct Access to SSN Nodes.* Intercepting sensor information from the SSN in a BSN endnode is not really feasible, since SSN nodes direct their readings only towards the gateway and sleep most of the time to save energy and bandwidth (so requests could take very long). It would also require a compatible WSAN. The first two options are both viable. Direct access to SSN nodes is not really an option.

4.5. Conclusions for WSAN Mobility Scenarios. The following conclusions can be drawn for the WSAN mobility scenarios.

- (i) Support for moving endnodes between compatible VSNS and SSNs is feasible when all WSANs are controlled by one party (e.g., using [12, 21]). When multiple parties are involved, these WSANs are likely to use different encryption keys or protocols. For more flexibility, the endnodes could be equipped with multiple keys so that they can operate in all WSANs that they have keys for. The downside of this is that the network keys could potentially be obtained from each endnode, therefore the encryption should preferably work such that the encryption key only makes it possible to send something towards the gateway, not to decrypt all WSAN traffic. This can be accomplished by encrypting with the public key of the receiving gateway or the application. When using multiple applications, a group key could be used for the applications or the WSAN gateway (or its IPG) could do the encryption. In the latter case, traffic from the gateway to applications can then be encrypted separately.
- (ii) In order to reduce interference from overlapping WSANs, the moving one could adapt its wireless resources before the overlap, for example, when similar GPS coordinates are detected.
- (iii) In order to reduce interference from overlapping compatible WSANs, the moving one could turn off its gateway [12] or change to intermediate node mode.
- (iv) In order to avoid endnodes of compatible WSANs to move between one another, they can use different network encryption keys so that only nodes that have both keys can move to the other WSAN and choose when changing WSAN is most appropriate.
- (v) As discussed, merging SSN and BSN directly proves troublesome, especially for obtaining SSN measurements from nodes that often sleep. It is therefore more practical to merge BSN and SSN data at the application layer.
- (vi) Encryption needs to be in place when BSN nodes send privacy-related information, else foreign objects can store and forward it.
- (vii) WSAN protocols should be robust against foreign protocols, in order to coexist with other WSANs that use the same wireless resources.

5. Schemes for Shared Usage of Mobile WSANs

In this section, schemes are analysed where multiple applications use the same WSAN data [6–8], while the IPG of both the application and the WSAN can change IP address while moving. For handling mobility of a WSAN and connected applications, a number of options can be considered. The following properties are used for comparing them; a number of them originate from the scenarios analysis and others from deployment and complexity concerns.

- (i) *Multi-Move*. Is simultaneous moving of source and destination supported?
- (ii) *Smart Buffering*. Can intelligent buffering be done for applications when connections fail? Alarm messages and recent measurements can better be sent first since they usually have higher priority than older measurements.
- (iii) *Overhead*. Is there inherent overhead in the approach?
- (iv) *Duplication Node*. Where are messages destined for multiple recipients duplicated (or broadcasted)? Obviously, closer to the recipients is more efficient, especially when different recipients require different data rates [23]. Options are endnode, gateway, server, router, proxy, or relay.
- (v) *Maturity*. Is the scheme still in research or is it already available?
- (vi) *Deployment Needs*. What is necessary to deploy this on the current Internet?
- (vii) *Access Control*. Who checks whether a destination is allowed to get the content? Depending on the number of destinations, the source may need to be taken out of the loop.
- (viii) *Request Method*. Can application requests like configuration and actuation be transferred to the source using the methods of the mobility scheme or is an additional method required?

The combination of the properties overhead, duplication node, and access control give an indication of the scalability of a scheme. For instance, when a scheme has much overhead and duplication and access control are done at the source, it is not very scalable. The scalability increases when access control and duplication can be done closer to the destinations and when the overhead decreases.

5.1. IPv6 Mobility. 6LoWPAN turns the WSAAN into an Internet Protocol version 6 (IPv6) network and addresses mobility of nodes with MIP [24]. This maintains reachability of all nodes in the WSAAN when they move inside or across WSAANs. However, WSAAN nodes are often not reachable since they are sleeping to save energy. The 6LoWPAN gateway may then send additional information when a connection is broken because of sleeping duty cycle. Furthermore, 6LoWPAN assumes the application will handle resending to each individual node in case of failure. 6LoWPAN uses network mobility (NEMO) [25] for mobility of the complete WSAAN. This means that the whole WSAAN can change its point of attachment, since the network prefix of the WSAAN has MIP support.

There are a number of issues with 6LoWPAN for WSAANs.

- (i) Traditionally, WSAAN nodes just sent their readings towards the gateway, and an application can connect

with the gateway to receive the sensor readings and for configuration. In 6LoWPAN, the gateway is an IPv6 router, and an IP application that is interested in the readings, that needs to register its IP address with each individual node (unless multicast can be used as destination, and applications can join the multicast group). This makes the binding between the application and nodes very tight which hinders scalability.

- (ii) The burden of reaching sleeping nodes is placed on the IP application(s) that use them. Since the time window for sending messages to a WSAAN node can be very small, this may be infeasible from remote application locations because of unpredictable latency on the path towards the WSAAN. It is therefore advised to let the WSAAN gateway handle reachability of nodes.
- (iii) 6LoWPAN requires both IPv6 and a home agent (HA) with support for NEMO. Neither of those are currently widely deployed.
- (iv) Every WSAAN node will need to do access control for configuration and actuation from applications.
- (v) When security is required, every WSAAN node will need to do network or application layer encryption to secure the path towards the IP application, independent of data link layer security that may already be in place.
- (vi) When multiple applications require sensor information from the same node, that node needs to send the information twice (unless there is multicast support), which doubles bandwidth both within the WSAAN and its uplink.
- (vii) Transmission control protocol (TCP) connections are a bad match with dynamic WSAAN nodes that are often sleeping and since packets may also be dropped because of congestion or because the node battery drained or the node moved outside range. It is often better to send a new measurement than to retry an old measurement that got dropped because of collisions.
- (viii) There are still numerous challenges related to security in 6LoWPAN [26], not to mention combining security with nodes that move between WSAANs.

There are a number of things that the gateway could potentially handle transparently when it uses packet inspection to preprocess requests towards nodes and responses from nodes.

- (i) Access control on behalf of nodes.
- (ii) Buffer requests to sleeping nodes until they wake up.
- (iii) Handling interest of an application, for instance, by using IP multicast.
- (iv) Replication of sensor readings to multiple applications.
- (v) Converting TCP connection towards a node to (UDP) packets, and injecting UDP packets from

TABLE 3: 6LoWPAN Mobility.

	Multimove	Smart buffering	Overhead	Duplication node	Maturity	Deployment needs	Access control	Request method
6LoWPAN	Ok	–	Low	Endnode	–	IPV6, HA + NEMO	Endnode	Same

TABLE 4: Mobility using instant messaging.

	Multimove	Smart buffering	Overhead	Duplication node	Maturity	Deployment needs	Access control	Request method
SIMPLE, XMMP, IRC	Ok	–	Medium-high	Gateway	++	Client API, server	Server	Same
PSYC	Ok	–	Low	Server	+/-	Client API, server	Server	Same

the node to an existing TCP connection towards an application.

Most of these options turn the gateway from a simple IPv6 router to a stateful router that requires deep packet inspection and making real-time packet modifications. Moreover, transparent network layer security with nodes will make many of these options impossible without sharing key material between nodes and the multiple WSAW gateways they need to attach to.

Because of all these issues, complicated solutions and lack of IPv6 and IP multicast deployment, for the time being it makes more sense to look for a WSAW mobility scheme that does not require full IP access to individual WSAW nodes and allows efficient usage by multiple applications. The results for 6LoWPAN are summarized in Table 3.

5.2. Instant Messaging. When communication between an IP application and WSAW is seen as instant messaging over IP, it can make use of existing instant messaging solutions. Since these solutions have either a publicly reachable server or distributed ones, both the WSAW and IP application can move while sending messages. Most instant messaging approaches offer encryption of the connection to the messaging server or the messages themselves. Only a limited number of instant messaging protocols are suitable for integration in applications (i.e., are an open standard [27]) popular ones are Internet Relay Chat (IRC) [28], Protocol for SYNchronous Conferencing (PSYC) [29], SIP for Instant Messaging, and Presence Leveraging Extensions [30], (SIMPLE) and Extensible Messaging and Presence Protocol (XMPP) [31]. Most of these protocols are not designed for reliability, but reachability is good for all of them since they all provide one or more ways to traverse through firewalls. The messages in these protocols are quite large because they are text based, especially in SIMPLE and XMPP.

Unfortunately, only few instant messaging solutions (e.g., PSYC) offer efficient ways to send to multiple recipients. The results for instant messaging are summarized in Table 4.

5.3. Mobile Stream Endpoints. When communication between an IP application and a WSAW is seen as a bidirectional

message stream, a number of mobility schemes can be envisaged. The results for mobile stream endpoints are summarized in Table 5.

Transparent Mobility. MIP could be used to transparently support mobility for both sides of this bidirectional stream. A drawback of this approach is that the WSAW needs to duplicate its sensor messages to each application, and that there is no good support for intelligent buffering when there is a connection outage, since MIP transparently keeps connections open even when there is temporarily no Internet connection.

Nomadic Mobility. A bidirectional connection could be setup between the WSAW and each application. The overhead can be low when a compact asynchronous protocol is used or high when a synchronous protocol with verbose messages is used (such as Simple Object Access Protocol (SOAP) [32]). In cases of connection loss, the WSAW would queue the messages that could not be sent and resend them in another order when the connection is reestablished later (possibly from a new IP address). Big drawbacks of nomadic mobility are the following.

- (i) The WSAW and application may not be able to find one another when they move at the same time.
- (ii) Communication is duplicated when multiple applications use one WSAW.
- (iii) The WSAW will need to do access control for every application.
- (iv) Bidirectional messaging does not work very well with web services when only one communication endpoint is publicly reachable. This would involve some sort of polling to get the requests from the other direction.

Nomadic Mobility with Public Server. Nomadic mobility can be enhanced using a publicly reachable server towards which both WSAWs and applications set up a bidirectional stream. This enables both WSAWs and applications to be mobile and at the same time reduces messaging that would

TABLE 5: Mobility using stream endpoints.

	Multimove	Smart buffering	Overhead	Duplication node	Maturity	Deployment needs	Access control	Request method
MobileIP	Ok	–	Low	Gateway	+/-	HA	Gateway	Same
Nomadic	—	+/-	Low-high	Gateway	++	Self-contained	Gateway	Same
Nomadic with server	Ok	+/-	Low-high	Server	+/-	Self-contained	Server	Via server
session mobility	Ok	+/-	Low	Gateway	+/-	SIP server	Gateway	SIP message

TABLE 6: Mobility of content source.

	Multimove	Smart buffering	Overhead	Duplication node	Maturity	Deployment needs	Access control	Request method
IP multicast	—	–	Low	Router	++	Router(s)	Router	Separate
Content-based routing	—	++ with proxy	Low-medium	Server	++	Client API, server(s)	Server	WSAN subscribes
Cache and forward routing	Ok	++	Low	Proxy	–	Multiple IP tunnels	Proxy	Separate
Partial sessions with relays	Ok	+	Low	Relay	–	SIP server and relays	Appl. server	SIP message

otherwise be duplicated at the source, that is, the WSAN only has to send sensor information once and the server duplicates it to all connected applications. An example is the asynchronous Ambient middleware [12]. With web services, the bidirectional messaging drawback worsens, since all interested applications will have to poll for updated sensor data and the WSAN will have to poll for configuration and actuation requests.

Session Mobility. A session could be set up between the IP application and the WSAN, with for instance the Session Initiation Protocol (SIP) [33, 34]. This session will contain the bidirectional messaging connection between the WSAN and the IP application. A SIP re-INVITE can be used to move the endpoints on either side to another IP address. Just like in nomadic mobility, messages during connection outage can be queued and sent in a different order when the connection is reestablished. The WSAN gateway is expected to handle sending messages to sleeping nodes and will forward all messaging from the WSAN to the application. Since the WSAN gateway is the IP endpoint of communication with applications, it can also easily support network layer security mechanisms such as virtual private network (VPN) and Internet Protocol Security (IPsec) [35]. There are a number of issues with this approach.

- (i) Each WSAN will need to do access control for every application.
- (ii) The WSAN needs to replicate messages for every attached application, wasting uplink bandwidth.
- (iii) Connection setup must be supported at both network ends (possibly private or protected networks).

5.4. WSAN as Content Source. When communication between an IP application and WSAN is seen as a content stream from the WSAN to all interested applications, the

messaging could be optimized by bundling communication to groups of applications. The results for mobile content sources are summarized in Table 6.

IP Multicast. IP multicast by the sender enables sending information to multiple recipients that can join the stream. IP multicast has a number of issues.

- (i) Mobility of the content source has only recently become a research topic and would typically involve context transfer between routers.
- (ii) Configuration and actuation messages towards the source would have to use a different protocol.
- (iii) IP multicast is mainly deployed in content distribution networks for pre-defined sources, and a lot of routers in the Internet do not yet support or allow it.

Content-Based Routing. With content-based routing [36–38], routing is done based on elements of the content body instead of the destination. Interested applications can subscribe for different content. Content-based routing has the following issues.

- (i) Mobility of the source has only recently become a research topic.
- (ii) Routing is often implemented on the application layer, inheriting application protocol overhead.
- (iii) Configuration and actuation requires reverse traffic, but WSAN could subscribe for these events.
- (iv) Messages are not cached for unconnected clients, however a subscription proxy [39] could be used.

Cache-and-Forward Routing. In cache-and-forward routing [40], interested applications can subscribe to content via a local post office which will look up the source post office via

a naming service. The content is sent by the source via cache-and-forward routers towards the destination(s). It allows efficiently sending content by the (mobile) source to multiple recipients. Both the sender and the receivers can be mobile.

- (i) Cache-and-forward routing is a future Internet research topic and would require deployment of a number of network elements.
- (ii) Configuration and actuation messages towards source would have to use a different protocol.

Partial Session Mobility with Relays. When the session between the WSN and applications is split in sub-session between the WSN and sub-sessions between the relay and each application, mobility can be supported for both the WSN and the applications without duplication at the source (but at the relay instead). The duplication can be further reduced by adding additional relays in different network segments. With SIP, an SIP application server can be used to automate splitting the sessions [41, 42]. The INVITE from an application towards the WSN is therefore picked up by an SIP application server and split into sub-sessions.

- (i) One sub-session between the WSN and the relay. This sub-session is typically set up when the first application subscribes to the WSN messages using an SIP INVITE.
- (ii) Other sub-sessions between the relay and each application. These sub-sessions are set up for each application that subscribes.
- (iii) To further reduce duplicate message streams, a sub-session between a relay and another relay can be set up when multiple applications in the same network segment subscribe to the same WSN stream. An SIP re-INVITE can be used to split the sub-session between the initial relay and the application(s) to one: between the relays and others between the new relay and each application.

A further advantage of splitting the streams is that private and protected networks are less of an issue, since no connection needs to be set up directly between these sort of networks, because a relay will be used that is reachable by both endpoints. Configuration and actuation requests towards the WSN can likewise be intercepted and be transformed into a configuration or actuation message towards the WSN after access control is checked and when there is no conflict between multiple applications. For example, when one application requires a temperature update every 5 minutes instead of the default 15 minutes, the WSN nodes can be configured to send it every 5 minutes, and the relay would forward it in this pace to the requesting application and keep forwarding it every 15 minutes to the other applications.

5.5. Reflection. Currently, only few reasonably mature mobility schemes offer buffering, low overhead and do not need an additional protocol for requests, namely, a nomadic

with server when a compact asynchronous protocol is used, session mobility and content-based routing with a proxy. However, session mobility does not scale well since it does access control at the gateway for each application and duplicates messages for multiple applications at the gateway, wasting precious uplink bandwidth. Content-based routing with a proxy does scale well, but does not guarantee reliable communication and source mobility is still a research topic. So, the nomadic with public server with a compact asynchronous protocol provides the best current option. The partial sessions with relay scheme forms a good, but nonmature, alternative when more applications use the WSN data, since it can add relays in different network segments on the fly. This option is similar in efficiency to cache and forward routing although it uses a session approach instead of post offices and can use its own protocols for sending messages towards the source. The cache and forward routing could also provide a solution for some of the shortcomings of 6LoWPAN, since it makes it possible to cache and forward the sensor messages for interested applications instead of direct IP connections.

6. Conclusions

This paper analysed scenarios in which different WSN and application movements take place. Moving endnodes between different WSNs can be supported by compatible WSNs but does not allow controlling when the movement takes place. With different encryption in each WSN, the endnodes can associate with the other WSN at a convenient moment.

To reduce interference between WSNs, the moving gateway can turn off its gateway or switch to intermediate node mode to make endnodes communicate with the other WSNs. To prevent interference from overlapping WSNs, it is advisable to adapt the wireless resources before the overlap occurs, for instance, by detecting similarity in GPS coordinates of the WSNs.

With different WSN types, data of overlapping WSNs can best be merged at the application layer. In order to support coexistence of WSNs using the same wireless resources, WSN protocols should be robust against foreign protocol messaging.

When privacy is required, as is often the case in body sensor networks, messages can better be encrypted with the public key of the receiving gateway (or middleware), which can in turn send it encrypted to one or more applications.

For sharing WSNs among few applications, nomadic mobility with a server using compact asynchronous messaging has the best properties. When the number of applications increases, schemes that bundle traffic towards groups of applications become more attractive.

Acknowledgment

This paper is partially funded by AgentSchapNL as part of the Dutch Point One initiative (free project).

References

- [1] N. Meratnia, B. J. V. D. Zwaag, H. W. V. Dijk, D. J. A. Bijwaard, and P. J. M. Havinga, "Sensor networks in the low lands," *Sensors*, vol. 10, no. 9, pp. 8504–8525, 2010.
- [2] S. Bosch, R. S. Marin-Perianu, P. J. M. Havinga, M. Marin-Perianu, A. Horst, and A. Vasilescu, "Automatic recognition of object use based on wireless motion sensors," in *Proceedings of the International Symposium on Wearable Computers 2010*, pp. 143–150, IEEE Computer Society, Seoul, Republic of Korea, October 2010.
- [3] M. Ali, T. Suleman, and Z. A. Uzmi, "MMAC: a mobility-adaptive, collision-free MAC protocol for wireless sensor networks," in *Proceedings of the 24th IEEE International Performance, Computing, and Communications Conference (IPCCC '05)*, pp. 401–407, April 2005.
- [4] H. Pham and S. Jha, "An adaptive mobility-aware MAC protocol for sensor networks (MS-MAC)," in *Proceedings of the IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 558–560, October 2004.
- [5] D. Zhang, Q. Li, X. Zhang, and X. Wang, "DE-ASS: an adaptive MAC algorithm based on mobility evaluation for wireless sensor networks," in *Proceedings of the 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM '10)*, pp. 1–5, September 2010.
- [6] L. Shu, M. Hauswirth, L. Cheng, J. Ma, V. Reynolds, and L. Zhang, "Sharing worldwide sensor network," in *Proceedings of the International Symposium on Applications and the Internet (SAINT '08)*, pp. 189–192, August 2008.
- [7] M. Isomura, T. Riedel, C. Decker, M. Beigl, and H. Horiuchi, "Sharing sensor networks," in *Proceedings of the IEEE International Conference on Distributed Computing Systems Workshops, ICDCS Workshops 2006*, p. 61, July 2006.
- [8] A. Malatras, A. Asgari, and T. Bauge, "Web enabled wireless sensor networks for facilities management," *IEEE Systems Journal*, vol. 2, no. 4, pp. 500–512, 2008.
- [9] M. Marin-Perianu, N. Meratnia, P. J. M. Havinga et al., "Decentralized enterprise systems: a multiplatform," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 57–66, 2007.
- [10] A. Avci, S. Bosch, M. Marin-Perianu, R. S. Marin-Perianu, and P. J. M. Havinga, "Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: a survey," in *Proceedings of the 23th International Conference on Architecture of Computing Systems (ARCS '10)*, pp. 167–176, VDE Verlag, Hannover, Germany, February 2010.
- [11] L. Evers, M. J. J. Bijl, R. S. Marin-Perianu, R. S. Marin-Perianu, and P. J. M. Havinga, "Wireless sensor networks and beyond: a case study on transport and logistics," Technical Report TR-CTIT-05-26, Centre for Telematics and Information Technology University of Twente, Enschede, The Netherlands, 2005.
- [12] D. J. A. Bijwaard, W. A. P. van Kleunen, P. J. M. Havinga, L. Kleiboer, and M. J. J. Bijl, "Industry: using dynamic WSNs in smart logistics for fruits and pharmacy," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys '11)*, pp. 218–231, ACM, Seattle, Wash, USA, November 2011.
- [13] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, pp. 1–14, ACM, New York, NY, USA, November 2009.
- [14] D. Johnson, C. Perkins, and J. Arkko, "Mobility support in IPv6," RFC 3775, IETF, 2004.
- [15] C. Perkins, "IP mobility support for IPv4, revised," RFC 5944, IETF, 2010.
- [16] Y. W. Law and P. Havinga, "How to secure a wireless sensor network," in *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information*, pp. 89–95, December 2005.
- [17] J. Lopez, "Unleashing public-key cryptography in wireless sensor networks," *Journal of Computer Security*, vol. 14, pp. 469–482, 2006.
- [18] G. Haigang, L. Ming, W. Xiaomin, C. Lijun, and X. Li, "An interference free cluster-based TDMA protocol for wireless sensor networks," in *Wireless Algorithms, Systems, and Applications*, X. Cheng, W. Li, and T. Znati, Eds., vol. 4138 of *Lecture Notes in Computer Science*, pp. 217–227, Springer, Berlin, Germany, 2006.
- [19] M. Macedo, A. Grilo, and M. Nunes, "Distributed latency-energy minimization and interference avoidance in TDMA wireless sensor networks," *Computer Networks*, vol. 53, no. 5, pp. 569–582, 2009.
- [20] T. Wu and S. Biswas, "Reducing inter-cluster TDMA interference by adaptive MAC allocation in sensor networks," in *Proceedings of the First International IEEE WoWMoM Workshop on Autonomic Communications and Computing (ACC '05)*, vol. 2, pp. 507–511, IEEE Computer Society, Washington, DC, USA, 2005.
- [21] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals," RFC 4919, IETF, 2007.
- [22] Inertia Technology, <http://inertia-technology.com>.
- [23] R. Muller and G. Alonso, "Efficient sharing of sensor networks," in *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS '06)*, pp. 109–118, October 2006.
- [24] E. Perera, V. Sivaraman, and A. Seneviratne, "Survey on network mobility support," *Mobile Computing and Communications Review*, vol. 8, pp. 7–19, 2004.
- [25] V. Devarapalli, R. Wakikawa, R. Petrescu, and P. Thubert, "Network mobility (NEMO) basic support protocol," RFC 3963, IETF, 2005.
- [26] S. Park, K. Kim, W. Haddad, S. Chakrabarti, and J. Laganier, "IPv6 over low power WPAN security analysis," Internet Draft 05, IETF, 2001.
- [27] ITU-T, "Open standard," <http://www.itu.int/en/ITU-T/ipr/Pages/open.aspx>.
- [28] W. Kantrowitz, "Network questionnaires," RFC 459, IETF, 1973.
- [29] C. V. Loesch, "Whitepaper on PSYC," <http://www.psyc.eu/whitepaper>.
- [30] IETF, "The SIMPLE working group charter," <http://data-tracker.ietf.org/wg/simple/charter>.
- [31] P. Saint-Andre, "Extensible messaging and presence protocol (XMPP): core," RFC 3920, IETF, 2004.
- [32] N. Mitra and Y. Lafon, "SOAP specifications," <http://www.w3.org/TR/soap>.
- [33] A. Berger and D. Romascanu, "Power ethernet MIB," RFC 3621, IETF, 2003.
- [34] A. Roach, "Session initiation protocol (SIP)-specific event notification," RFC 3265, IETF, 2002.
- [35] S. Kent and K. Seo, "Security architecture for the internet protocol," RFC 4301, IETF, 2005.
- [36] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajao, R. E. Strom, and D. C. Sturman, "Efficient multicast protocol for content-based publish-subscribe systems," in *Proceedings of the 19th IEEE International Conference on Distributed Computing Systems (ICDCS'99)*, pp. 262–272, June 1999.

- [37] L. Fiege, F. Gartner, O. Kasten, and A. Zeidler, "Supporting mobility in content-based publish/subscribe middleware," in *Middleware 2003*, M. Endler and D. Schmidt, Eds., vol. 2672 of *Lecture Notes in Computer Science*, pp. 998–998, Springer, Berlin, Germany, 2003.
- [38] Elvin, <http://www.elvin.org>.
- [39] P. Sutton, R. Arkins, and B. Segall, "Supporting disconnectedness-transparent information delivery for mobile and invisible computing," in *Proceedings of the 1st International Symposium on Cluster Computing and the Grid (CCGRID '01)*, p. 277, IEEE Computer Society, Brisbane, Australia, May 2001.
- [40] S. Paul, R. Yates, D. Raychaudhuri, and J. Kurose, "The cache-and-forward network architecture for efficient mobile content delivery services in the future internet," in *Proceedings of the First ITU-T Kaleidoscope Academic Conference in Innovations in NGN: Future Network and Services (K-INGN '08)*, pp. 367–374, May 2008.
- [41] J. Aartse Tuijn and D. Bijwaard, "Spanning a multimedia session across multiple devices," *Bell Labs Technical Journal*, vol. 12, no. 4, pp. 179–193, 2006.
- [42] S. van der Gaast and D. Bijwaard, "Efficiency of personalized content distribution," *Bell Labs Technical Journal*, vol. 13, no. 2, pp. 135–145, 2008.

Research Article

Localization Algorithm Based on Maximum a Posteriori in Wireless Sensor Networks

Kezhong Lu, Xiaohua Xiang, Dian Zhang, Rui Mao, and Yuhong Feng

College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

Correspondence should be addressed to Kezhong Lu, kzlu@szu.edu.cn

Received 14 July 2011; Revised 19 September 2011; Accepted 19 September 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Kezhong Lu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many applications and protocols in wireless sensor networks need to know the locations of sensor nodes. A low-cost method to localize sensor nodes is to use received signal strength indication (RSSI) ranging technique together with the least-squares trilateration. However, the average localization error of this method is large due to the large ranging error of RSSI ranging technique. To reduce the average localization error, we propose a localization algorithm based on maximum a posteriori. This algorithm uses the Baye's formula to deduce the probability density of each sensor node's distribution in the target region from RSSI values. Then, each sensor node takes the point with the maximum probability density as its estimated location. Through simulation studies, we show that this algorithm outperforms the least-squares trilateration with respect to the average localization error.

1. Introduction

The process of determining the physical locations of sensor nodes is known as localization, which is a fundamental problem in wireless sensor networks [1, 2]. The locations of sensor nodes are essential in many applications and protocols. For example, the sensed information about an event without the location where it takes place is often meaningless. Similarly, geographic routing relies on the locations of nodes to forward packets [3].

The locations of sensor nodes can be directly obtained by preconfiguration or global positioning system (GPS). Pre-configuration requires each sensor node being placed at a known location, which is only suitable for the case that sensor nodes are easy to be placed and their number is small. On the other hand, a sensor node equipped with a GPS receiver is costly and does not work indoors. Therefore, both the above two methods are impractical for large-scale low-cost wireless sensor networks. It is desired that the locations of sensor nodes can be induced from their interactions, such as the detections of the distances between neighbors.

Many localization algorithms first use a ranging technique to estimate the Euclidean distances between nodes,

and then use the least-squares trilateration to determine the locations of sensor nodes by these estimated distances. Some conventional ranging techniques are received signal strength indication (RSSI), time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA), and so forth [4]. Among them, RSSI ranging technique has the least requirement for hardware, as the radio chip of current sensor node usually has a built-in function of reading RSSI value. But RSSI value is vulnerable to being disturbed by the surrounding environment and the ranging error of RSSI ranging technique may be at most $\pm 50\%$ [5]. Furthermore, the least-squares trilateration is sensitive to ranging errors [2]. If using RSSI ranging technique together with the least-squares trilateration to localize sensor nodes, the average localization error is very large.

To reduce the average localization error, we propose a localization algorithm based on maximum a posteriori. This algorithm uses the probability approach to estimate the location of each sensor node directly from RSSI values. Extensive simulation results have shown that the average localization error of this algorithm is less than that of the least-squares trilateration.

The remainder of this paper is organized as follows. Related work is discussed in Section 2, the network model is defined in Section 3, and our proposed localization algorithm is described in Section 4. Simulation results that illustrate the performance are included in Section 5, and Section 6 is the conclusion.

2. Related Work

Since localization is a fundamental problem in wireless sensor networks, there are many research works focusing on it recently. Localization algorithms can be divided into two categories: anchor-based localization algorithms and nonanchor-based localization algorithms. An anchor is a special node which has a priori knowledge of its location.

In anchor-based localization algorithms, the location of each sensor node is determined only by its distances from anchors. Priyantha et al. developed the cricket location support system which provides localization services for indoor mobile node [6]. Bulusu et al. proposed a GPS-less localization algorithm in which each mobile node localizes itself to the centroid of its adjacent connecting anchors [7]. Niculescu and Nath proposed a family of distributed localization algorithms “ad hoc positioning system” (APS) [8, 9]. In these algorithms, each node measures its distances from anchors by performing multihop propagation of distances to anchors throughout the network. Kumar et al. used RSSI-based weighted centroid to improve the localization algorithm proposed by Bulusu et al. [10]. Li and Liu proposed the rendered path (REP) protocol for locating nodes in anisotropic sensor networks with holes [11]. Lederer et al. also studied the problem of localizing a large sensor network having a complex shape, possibly with holes [12].

In nonanchor-based localization algorithms, the location of each sensor node is determined also by the distances between sensor nodes. Doherty et al. proposed a constraint-based localization scheme using semidefinite programming (SDP) to find a solution to the localization problem [13]. Shang et al. proposed an algorithm using classical multidimensional scaling (MDS) technique to calculate the locations of nodes given a set of distances [14]. Kwon et al. proposed a localization algorithm based on least square scaling (LSS) which is a variant of multidimensional scaling technique [15]. Khan et al. proposed a distributed iterative localization algorithm in m -dimensional Euclidean space with a minimal number of $m+1$ anchors [16]. Ding et al. viewed localization as a (nearest) Euclidean distance matrix (EDM) completion problem and thus gave an EDM approach [17]. Zhu et al. used stress normal property to localize sensor nodes with enough perturbation data [18].

Many of the above-mentioned algorithms [12, 14, 15, 17] have a similar optimization objective as the least-squares trilateration, that is, minimizing the (weighted) sum of all squared differences between each estimated distance and the corresponding distance calculated by the estimated locations of nodes. But when the distances are estimated by RSSI ranging technique, this optimization objective is likely subject to a large average localization error. In this paper,

we present another optimization objective with which sensor nodes can be localized more accurately.

3. Network Model

Before describing our proposed localization algorithm, we first assume the model of wireless sensor network as follows.

- (1) A wireless sensor network is deployed in a planar region B . Suppose that B is a rectangle, whose length is l and width is w . But later we will see that the proposed algorithm is not dependent on the shape of B . Without loss of generality, we assume that the lower left corner of B is the origin and the coordinate of the upper-right corner of B is (l, w) .
- (2) All sensor nodes are uniformly distributed in B . Anchors have a larger transmission range than sensor nodes. Each sensor node and anchor is denoted by a point in B . Let n denote the total number of anchors, a_i denote the i th anchor, and (x_i, y_i) denote the coordinate of a_i .
- (3) Initially each anchor broadcasts a beacon containing its location information. Then, each sensor node collects the RSSI values of all its neighbor anchors through these beacons. The RSSI value read by a sensor node obeys the wide log-normal shadowing radio signal propagation model [19]:

$$R(d) = P_T - PL(d_0) - 10\eta \log_{10} \frac{d}{d_0} + X_\sigma. \quad (1)$$

In (1), $R(d)$ denotes the RSSI value when the distance between the receiver and the transmitter is d ; P_T is the power of the transmitter; $PL(d_0)$ is a known reference power value at a reference distance d_0 from the transmitter; η is the path loss exponent that measures the rate at which the RSSI value decreases with distance; X_σ is a zero mean Gaussian distributed random variable with standard deviation and it accounts for the random effect of shadowing, that is, $X_\sigma = N(0, \sigma^2)$.

Given these known RSSI values, the locations of sensor nodes can be estimated by a localization algorithm. The major measurement of a localization algorithm is the average localization error, defined as the average distance between the actual location and the estimated location of each sensor node [5]. Because wireless sensor networks often work in unfriendly environments, some anchors may be faulty, which means they have incorrect information about their own locations. Therefore, fault tolerance is also important to a localization algorithm. In this paper, we define it as the ability to maintain a good localization result even if some anchors are faulty. Besides, execution time is also a common measurement of a localization algorithm.

4. Algorithm Description

In this section, we will describe our proposed localization algorithm based on maximum a posteriori. Consider a sensor node s . Let (x_s, y_s) denote the coordinate of s and r_k denote the random variable of the RSSI value of a_k read by s , where $0 \leq k < n$. Assume that the values of r_1, r_2, \dots, r_n in a test are R_1, R_2, \dots, R_n , respectively. The basic idea of our proposed localization algorithm is as follows. First, the target region B is divided into small grids of the same size. Next, the probability of s being in each grid is calculated from R_1, R_2, \dots, R_n . Then, the center of the grid with the largest probability is taken as the estimated value of (x_s, y_s) . Let g denote the side length of each grid. Without loss of generality, assume that the target region B has exact l/g grids in the horizontal direction and w/g grids in the vertical direction. Let G_{ij} denote the grid locating at the i th row and the j th column, where $0 \leq i < w/g, 0 \leq j < l/g$.

Let E_{ij} denote the event of s being in G_{ij} , whose probability is denoted by $P\{E_{ij}\}$. Let the conditional probability $P\{r_k = R_k, 0 \leq k < n \mid E_{ij}\}$ denote the probability that for all $0 \leq k < n$, $r_k = R_k$ under the condition of s being in G_{ij} . Now we have the condition that for all $0 \leq k < n$, $r_k = R_k$ and want to compute the conditional probability $P\{E_{ij} \mid r_k = R_k, 0 \leq k < n\}$. That is, to compute the a posteriori probability from priori probability. By the Bayes formula, we obtain $P\{E_{ij} \mid r_k = R_k, 0 \leq k < n\}$ as follows:

$$P\{E_{ij} \mid r_k = R_k, 0 \leq k < n\} = \frac{P\{r_k = R_k, 0 \leq k < n \mid E_{ij}\} P\{E_{ij}\}}{\sum_{0 \leq u < (w/g), 0 \leq v < (l/g)} P\{r_k = R_k, 0 \leq k < n \mid E_{uv}\} P\{E_{uv}\}}. \quad (2)$$

Because each sensor node is uniformly distributed in B , s has the same probability in each grid, that is, for all $0 \leq u < w/g$ and $0 \leq v < l/g$, $P\{E_{uv}\}$ is equal. So (2) can be simplified as follows:

$$P\{E_{ij} \mid r_k = R_k, 0 \leq k < n\} = \frac{P\{r_k = R_k, 0 \leq k < n \mid E_{ij}\}}{\sum_{u,v} P\{r_k = R_k, 0 \leq k < n \mid E_{uv}\}}. \quad (3)$$

Moreover, because s is also uniformly distributed in G_{ij} , $P\{r_k = R_k, 0 \leq k < n \mid E_{ij}\}$ is equal to the average probability that for all $0 \leq k < n$, $r_k = R_k$ under the condition of s being at each point in G_{ij} . Then, we obtain the following equation:

$$P\{r_k = R_k, 0 \leq k < n \mid E_{ij}\} = \frac{\int_{i_g}^{(i+1)g} \int_{j_g}^{(j+1)g} P\{r_k = R_k, 0 \leq k < n \mid x_b = x, y_b = y\} dx dy}{\int_{i_g}^{(i+1)g} \int_{j_g}^{(j+1)g} dx dy}. \quad (4)$$

When s is at a certain point in B , the RSSI value of each anchor read by s is not interfered with the RSSI values of other anchors. Therefore, under the condition that $x_b = x$ and $y_b = y$, the events $r_1 = R_1, r_2 = R_2, \dots, r_n = R_n$ are independent. Then, we obtain $P\{r_k = R_k, 0 \leq k < n \mid x_b = x, y_b = y\} = P\{r_1 = R_1 \mid x_b = x, y_b = y\} P\{r_2 = R_2 \mid x_b = x, y_b = y\} \cdots P\{r_n = R_n \mid x_b = x, y_b = y\}$. Combine it with (3) and (4), and we obtain the following equation:

$$P\{E_{ij} \mid r_k = R_k, 0 \leq k < n\} = \frac{\int_{i_g}^{(i+1)g} \int_{j_g}^{(j+1)g} \prod_{0 \leq k < n} P\{r_k = R_k \mid x_b = x, y_b = y\} dx dy}{\sum_{u,v} \int_{u_g}^{(u+1)g} \int_{v_g}^{(v+1)g} \prod_{0 \leq k < n} P\{r_k = R_k \mid x_b = x, y_b = y\} dx dy}. \quad (5)$$

According to (5), we need to calculate $P\{r_k = R_k \mid x_b = x, y_b = y\}, 0 \leq k < n$. However, it can be seen from (1) that r_k is a continuous random variable, so we cannot directly obtain $P\{E_{ij} \mid r_k = R_k, 0 \leq k < n\}$ through (5). But we can compute the probability of r_k being in the interval $[R_k, R_k + \varepsilon)$ under the condition of s being at a point (x, y) , which is denoted by $P\{R_k \leq r_k < R_k + \varepsilon \mid x_b = x, y_b = y\}$. So we can obtain the following equation:

$$P\{E_{ij} \mid r_k = R_k, 0 \leq k < n\} = \lim_{\varepsilon \rightarrow 0} \frac{\int_{i_g}^{(i+1)g} \int_{j_g}^{(j+1)g} \prod_{0 \leq k < n} P\{R_k \leq r_k < R_k + \varepsilon \mid x_b = x, y_b = y\} dx dy}{\sum_{u,v} \int_{u_g}^{(u+1)g} \int_{v_g}^{(v+1)g} \prod_{0 \leq k < n} P\{R_k \leq r_k < R_k + \varepsilon \mid x_b = x, y_b = y\} dx dy}. \quad (6)$$

Let d_k denotes the distance between the point (x, y) and a_k , that is, $d_k = \sqrt{(x - x_k)^2 + (y - y_k)^2}$. Let $\beta_k = R_k - P_T + PL(d_0) - 10\eta \log_{10}(d_0)$, $0 \leq k < n$. According to (1), if r_k is in the interval $[R_k, R_k + \varepsilon)$, then X_σ is in the interval $[\beta_k + 10\eta \log_{10} d_k, \beta_k + 10\eta \log_{10} d_k + \varepsilon)$, whose probability is as follows:

$$P\{\beta_k + 10\eta \log_{10} d_k \leq X_\sigma < \beta_k + 10\eta \log_{10} d_k + \varepsilon\} = \int_{\beta_k + 10\eta \log_{10} d_k}^{\beta_k + 10\eta \log_{10} d_k + \varepsilon} \frac{1}{\sqrt{2\pi}} e^{-(1/2)z^2} dz. \quad (7)$$

Combine (6) and (7), and we obtain the following equation:

$$P\{E_{ij} \mid r_k = R_k, 0 \leq k < n\} = \lim_{\varepsilon \rightarrow 0} \frac{\int_{i_g}^{(i+1)g} \int_{j_g}^{(j+1)g} \prod_{0 \leq k < n} \int_{\beta_k + 10\eta \log_{10} d_k}^{\beta_k + 10\eta \log_{10} d_k + \varepsilon} (1/\sqrt{2\pi}) e^{-(1/2)z^2} dz dx dy}{\sum_{u,v} \int_{u_g}^{(u+1)g} \int_{v_g}^{(v+1)g} \prod_{0 \leq k < n} \int_{\beta_k + 10\eta \log_{10} d_k}^{\beta_k + 10\eta \log_{10} d_k + \varepsilon} (1/\sqrt{2\pi}) e^{-(1/2)z^2} dz dx dy}. \quad (8)$$

We have $\lim_{\varepsilon \rightarrow 0} \int_{\beta_k + 10\eta \log_{10} d_k}^{\beta_k + 10\eta \log_{10} d_k + \varepsilon} (1/\sqrt{2\pi}) e^{-(1/2)z^2} dz = (\varepsilon/\sqrt{2\pi}) e^{-(1/2)(\beta_k + 10\eta \log_{10} d_k)^2}$ and substitute it into (8). Then, we obtain the following equation:

$$P\{E_{ij} \mid r_k = R_k, 0 \leq k < n\} = \frac{\int_{i_g}^{(i+1)g} \int_{j_g}^{(j+1)g} e^{-(1/2)\sum_{0 \leq k < n} (\beta_k + 10\eta \log_{10} d_k)^2} dx dy}{\sum_{u,v} \int_{u_g}^{(u+1)g} \int_{v_g}^{(v+1)g} e^{-(1/2)\sum_{0 \leq k < n} (\beta_k + 10\eta \log_{10} d_k)^2} dx dy}. \quad (9)$$

It can be seen from (9) that if G_{ij} has the largest value of $\int_{i_g}^{(i+1)g} \int_{j_g}^{(j+1)g} e^{-(1/2)\sum_{0 \leq k < n} (\beta_k + 10\eta \log_{10} d_k)^2} dx dy$ among all grids, then s has the largest probability in G_{ij} among all grids. If the size of grid is small enough, we have $\lim_{g \rightarrow 0} \int_{i_g}^{(i+1)g} \int_{j_g}^{(j+1)g} e^{-(1/2)\sum_{0 \leq k < n} (\beta_k + 10\eta \log_{10} d_k)^2} dx dy = g^2 e^{-(1/2)\sum_{0 \leq k < n} (\beta_k + 10\eta \log_{10} d_k)^2}$. Therefore, we can select the point with the smallest value of $\sum_{0 \leq k < n} (\beta_k + 10\eta \log_{10} d_k)^2$ in the target region B as the estimated location of s . We define the function $f(x, y)$ as follows:

$$f(x, y) = \sum_{0 \leq k < n} \left[R_k - P_T + PL(d_0) + 10\eta \log_{10} \frac{\sqrt{(x - x_k)^2 + (y - y_k)^2}}{d_0} \right]^2. \quad (10)$$

That is for the variable (x, y) whose domain is the target region B , to find a point (\hat{x}, \hat{y}) with the minimum value of $f(x, y)$. However, the function $f(x, y)$ is relatively complicated, so we cannot obtain the analytic expression of (\hat{x}, \hat{y}) by the partial differential method. Alternatively, we adopt an approximation method described as follows. First, the target region is divided into small grids and the value of $f(x, y)$ at the center of each grid is computed. Then, the center of the grid with the minimum value of $f(x, y)$ is taken as the approximation of (\hat{x}, \hat{y}) . If the size of grid is $g \times g$, then the total number of grids is lw/g^2 and the time complexity of localizing a node is $O(nlw/g^2)$. Obviously, when the target region is fixed, the execution time will become longer as the size of grid become smaller. For large wireless sensor networks, we can use the multi-grid method to reduce the execution time. First, the target region is divided into larger grids and the value of $f(x, y)$ at each grid point is computed. Next, those grids with relatively larger value of $f(x, y)$ are discarded. Then, the remaining grids are repeatedly divided into smaller grids until the size of grid reaches the required accuracy.

Furthermore, we analyze (10). Let \hat{d}_k denote the estimated distance between s and a_k only estimated from R_k . It

is easy to know that \hat{d}_k should be equal to d in (1) when X_σ is 0:

$$R_k = P_T - PL(d_0) - 10\eta \log_{10} \frac{\hat{d}_k}{d_0}. \quad (11)$$

Combine (10) and (11), and we obtain the following equation:

$$f(x, y) = \left(\frac{10\eta}{\ln 10} \right)^2 \sum_{0 \leq k < n} (\ln d_k - \ln \hat{d}_k)^2. \quad (12)$$

Equation (12) illuminates that the optimum estimated location of s should be the point with the minimum value of $\sum_{0 \leq k < n} (\ln d_k - \ln \hat{d}_k)^2$ in the target region B .

5. Performance Evaluation

5.1. Simulation Environment. To evaluate the performance of our proposed localization algorithm based on maximum a posteriori (MAP), we developed a simulation program realizing MAP algorithm. We compare MAP algorithm with the least-squares trilateration (LST) in which the point with the minimum value of $\sum_{0 \leq k < n} (d_k - \hat{d}_k)^2$ in the target region is taken as the estimated location of a node.

In the simulation, the target region B is a square region of 1000 m \times 1000 m. The transmission range of anchors is 1500 m. Sensor nodes and anchors are randomly and uniformly distributed in B . The RSSI value read by a sensor node is simulated according to (1). The value of each parameter is taken from a typical wireless sensor network [19]: P_T is set to 4 dBm, d_0 is set to 1 m, $PL(d_0)$ is set to 55 dB, η is set to 4, and the range of σ is set to [2, 14]. The Gaussian distributed random number X_σ is generated by the Box-Muller method. The platform has Intel Dual Core 2.80 GHz CPU and 1 GB memory. To make simulation results more accurate, for each simulation we perform 100 times and take the average result.

5.2. Grid Size. In MAP algorithm, the target area is divided into small grids to approximately obtain the point with the maximum probability density. The smaller the grid is, the closer the approximate solution is to the accurate solution, but the larger the amount of calculation is. Therefore, we first need to select the appropriate grid size. In the simulation, σ is set to 5, the number of anchors is set to 20, the number of sensor nodes is set to 100, and the side length of grid varies from 5 m to 100 m. Figure 1 shows the average localization errors of MAP algorithm. It can be seen that when the grid size is relatively large, the average localization error is significantly impacted by the grid size. But when the grid size of grid is small to a certain extent, this impact is almost

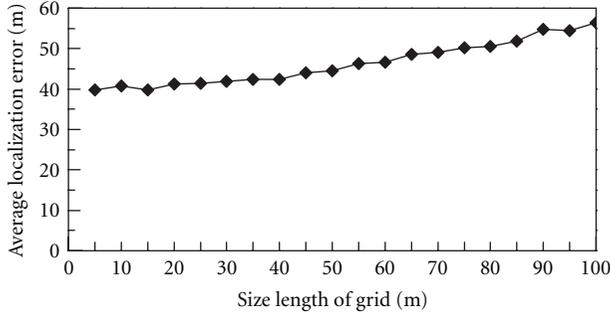


FIGURE 1: Average localization errors of MAP algorithm under different grid sizes.

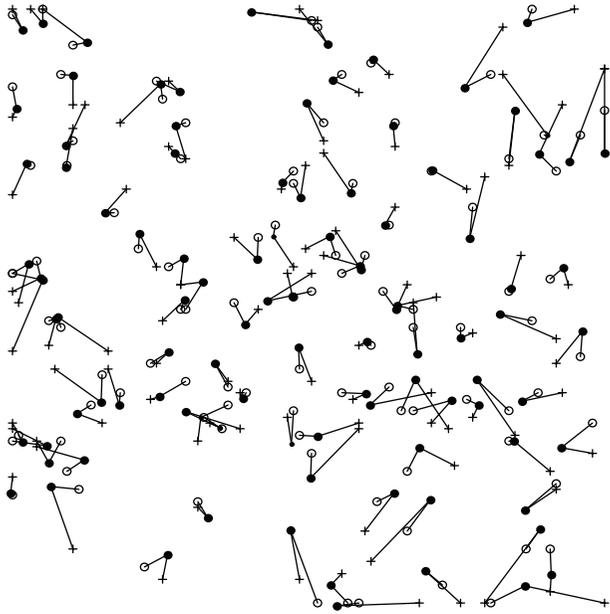
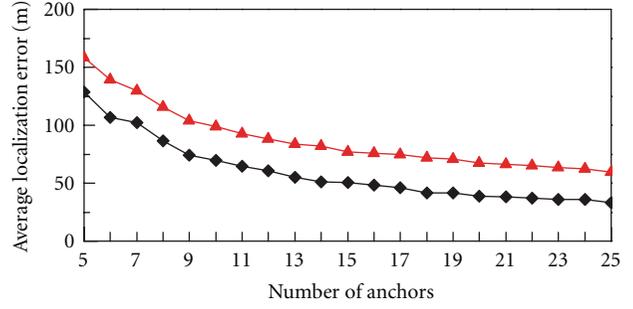


FIGURE 2: A localization result of MAP algorithm and LST algorithm. In the figure, the solid points represent the actual locations of sensor nodes, the hollow points represent the estimated locations of sensor nodes computed by MAP algorithm, and the cross points represent the estimated locations of sensor nodes computed by LST algorithm.

negligible. In the following, we will take the grid size as $10\text{ m} \times 10\text{ m}$.

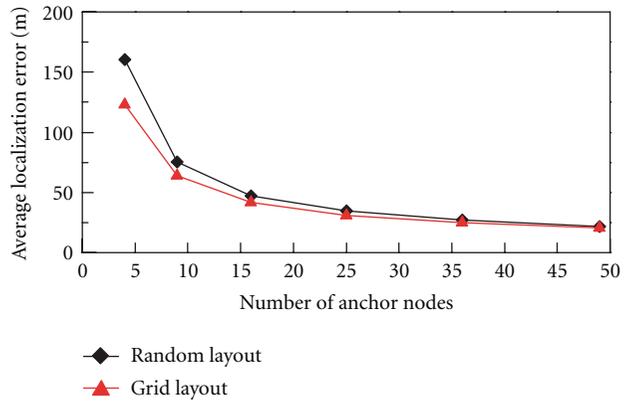
5.3. Localization Error. Figure 2 shows a localization result of MAP algorithm and LST algorithm. Both algorithms have the same inputs: σ is set to 5, the number of anchors is set to 20, and the number of sensor nodes is set to 100. It can be seen that the estimated location of most sensor nodes computed by MAP algorithm are closer to their actual locations. Accordingly, MAP algorithm has a smaller average localization error than LST algorithm.

Next, we test how the localization error is impacted by the number of anchors. In the simulation, σ is set to 5, the number of sensor nodes is set to 100, and the number of anchors varies from 5 to 25. It can be seen from Figure 3



◆ MAP algorithm
▲ LST algorithm

FIGURE 3: Average localization errors of MAP algorithm and LST algorithm under different numbers of anchors.



◆ Random layout
▲ Grid layout

FIGURE 4: Average localization errors of MAP algorithm under the random layout and the grid layout of anchor nodes.

that with the number of anchors increasing, the average localization errors of both algorithms are reduced. But MAP algorithm can achieve a smaller average localization error, which is reduced by nearly 34.8% compared with the least-squares trilateration.

Then, we test how the location error is impacted by the layout of anchor nodes. In the simulation, σ is set to 5, the number of sensor node is set to 100, the number of anchors varies among 4, 9, 16, 25, 36, and 49, and anchors are placed by the random layout and the grid layout, respectively. Figure 4 shows the average location errors of MAP algorithm under these two layouts. The average location error under the grid layout is nearly 87.8% of that under the random layout. Therefore, anchor nodes should be placed by the grid layout in practice.

The ranging error is a primary cause of the localization error, which depends on σ : the larger σ is, the larger the ranging error is. In the simulation, the number of anchors is set to 20, the number of sensor nodes is set to 100, and σ varies from 2 to 14. Figure 5 shows the average localization errors of MAP algorithm and LST algorithm. It can be seen that the average localization errors of both algorithms are approximately proportional to σ .

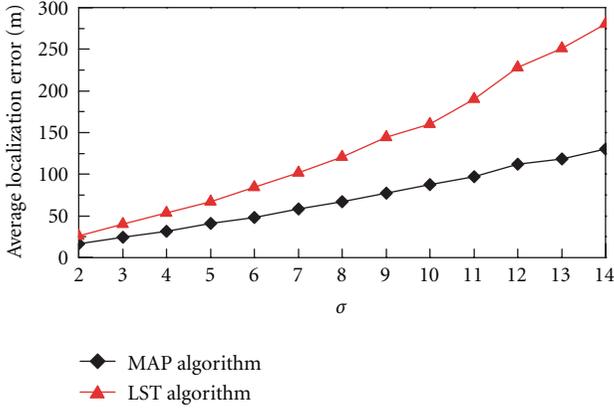


FIGURE 5: Average localization errors of MAP algorithm and LST algorithm under different values of σ .

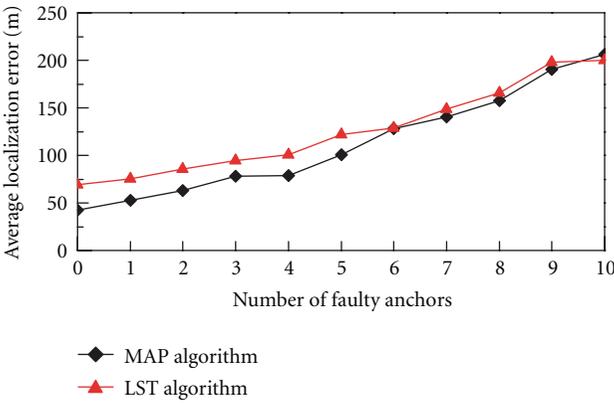


FIGURE 6: Average localization errors of MAP algorithm and LST algorithm under different numbers of faulty anchors.

5.4. Fault Tolerance. In MAP algorithm, the location of each sensor node is determined by all its neighbor anchors. If only a small number of anchors are faulty, the estimated location of each sensor node cannot have a big change. In the simulation, σ is set to 5, the number of anchors is set to 20, the number of sensor nodes is set to 100, and the number of faulty anchors varies from 0 to 10. Figure 6 shows the average location errors of MAP algorithm and LST algorithm. It can be seen that when less than 25% of anchors are faulty, the average localization error of MAP algorithm increase less than 85%.

5.5. Execution Time. Finally, we analyze the average execution time of MAP algorithm by simulation. In the simulation, the number of sensor nodes is set to 100. Figure 7 shows the average execution times of the two algorithms under different numbers of anchor nodes when the size of grid is 10 m \times 10 m. Figure 8 shows the average execution times of the two algorithms under different grid sizes when the number of anchors is 20. It can be seen that the execution times of both algorithms are approximately proportional to the number of anchor nodes and are approximately inversely proportional to the acreage of grid. This result is consistent with the analysis in Section 4. In a general case, MAP algorithm can

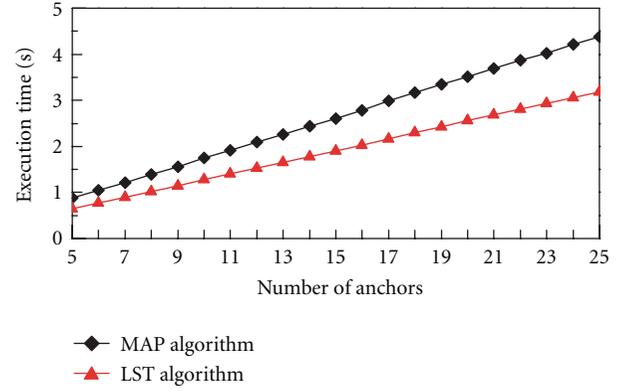


FIGURE 7: Execution times of MAP algorithm and LST algorithm under different number of anchors.

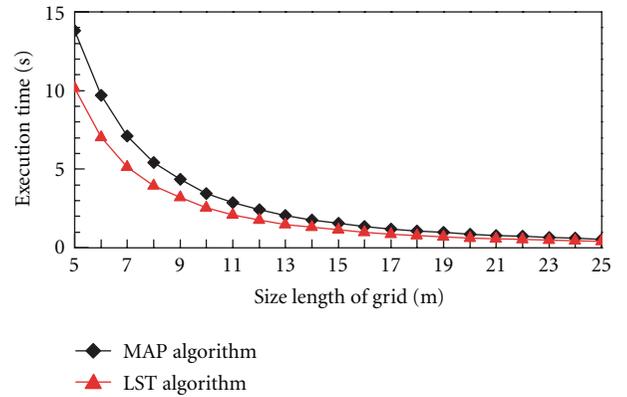


FIGURE 8: Execution times of MAP algorithm and LST algorithm under different grid sizes.

localize a sensor node in a short time. Since MAP algorithm has a more complicated calculation than LST algorithm, the execution time of MAP algorithm is longer.

6. Conclusion

RSSI ranging-based localization is regarded as a cost-effective solution for sensor node localization. But RSSI ranging technique has a large ranging error, which will bring a large average localization error to the general least-squares trilateration. In this paper, we propose a localization algorithm based on maximum a posteriori probability (MAP). In this algorithm, the point with the maximum probability density in the target region is taken as the estimated location of a sensor node. Extensive simulation results demonstrate the effectiveness of MAP algorithm. This algorithm reduces the average localization error by nearly 34.8% compared with the least-squares trilateration. Even if the number of anchors is small, this algorithm can also achieve a relatively small average localization error. In addition, the execution time of this algorithm is very short.

As a future work, we are currently studying when anchors are absent, how to determine the probability density of the distribution of a sensor node only by the RSSI values of

sensor nodes. Moreover, we plan to conduct some practical experiments to confirm the effectiveness of our proposed algorithm.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments which have significantly improved the quality of the paper. This paper was supported by the National Natural Science Foundation of China (Grant no. 61003272, no. 61033009, no. 61170076, and no. 61103001), the Guangdong Natural Science Foundation (Grant no. 10351806001000000), and the Shenzhen Science and Technology Foundation (Grant no. JC201005280408A and JC2009D3120046A).

References

- [1] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*, John Wiley & Sons, 2010.
- [2] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.
- [3] J. You, Q. Han, D. Lieckfeldt, J. Salzmann, and D. Timmermann, "Virtual position based geographic routing for wireless sensor networks," *Computer Communications*, vol. 33, no. 11, pp. 1255–1265, 2010.
- [4] J. Wang, R. K. Ghosh, and S. K. Das, "A survey on sensor localization," *Journal of Control Theory and Applications*, vol. 8, no. 1, pp. 2–11, 2010.
- [5] G. Zanca, F. Zorzi, A. Zanella, and M. Zorzi, "Experimental comparison of RSSI-based localization algorithms for indoor wireless sensor networks," in *Proceedings of the 3rd Workshop on Real-World Wireless Sensor Networks (REALWSN '08)*, pp. 1–5, April 2008.
- [6] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "Cricket location-support system," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 32–43, August 2000.
- [7] 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.
- [8] D. Niculescu and B. Nath, "DV based positioning in Ad Hoc networks," *Telecommunication Systems*, vol. 22, no. 1–4, pp. 267–280, 2003.
- [9] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (InfoCom '03)*, pp. 1734–1743, April 2003.
- [10] A. Kumar, V. Kumar, and V. Kapoor, "Range free localization schemes for wireless sensor networks," in *Proceedings of the 10th WSEAS International Conference on Software Engineering*, pp. 101–106, 2011.
- [11] M. Li and Y. Liu, "Rendered path: range-free localization in anisotropic sensor networks with holes," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 320–332, 2010.
- [12] S. Lederer, Y. Wang, and J. Gao, "Connectivity-based localization of large-scale sensor networks with complex shape," *ACM Transactions on Sensor Networks*, vol. 5, no. 4, article 31, 2009.
- [13] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom '01)*, pp. 1655–1663, April 2001.
- [14] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 961–974, 2004.
- [15] Y. Kwon, K. Mechitov, S. Sundresh, W. Kim, and G. Agha, "Resilient localization for sensor networks in outdoor environments," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS '05)*, pp. 643–652, June 2005.
- [16] U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 2000–2016, 2009.
- [17] Y. Ding, N. Krislock, J. Qian, and H. Wolkowicz, "Sensor network localization, Euclidean distance matrix completions, and graph realization," in *Proceedings of the 1st ACM International Workshop on Mobile Entity Localization and Tracking in GPS-Less Environments (MELT '08)*, pp. 129–134, 2008.
- [18] Y. Zhu, S. J. Gortler, and D. Thurston, "Sensor network localization using sensor perturbation," *ACM Transactions on Sensor Networks*, vol. 7, no. 4, p. 36, 2011.
- [19] K. Yedavalli, B. Krishnamachari, S. Ravulath, 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, April 2005.

Research Article

Adaptive WSN Scheduling for Lifetime Extension in Environmental Monitoring Applications

Jong Chern Lim and Chris Bleakley

*UCD Complex and Adaptive Systems Laboratory, UCD School of Computer Science and Informatics,
University College Dublin, Ireland*

Correspondence should be addressed to Jong Chern Lim, jongchern@gmail.com

Received 15 June 2011; Accepted 27 August 2011

Academic Editor: Yuhang Yang

Copyright © 2012 J. C. Lim and C. Bleakley. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless sensor networks (WSNs) are often used for environmental monitoring applications in which nodes periodically measure environmental conditions and immediately send the measurements back to the sink for processing. Since WSN nodes are typically battery powered, network lifetime is a major concern. A key research problem is how to determine the data gathering schedule that will maximize network lifetime while meeting the user's application-specific accuracy requirements. In this work, a novel algorithm for determining efficient sampling schedules for data gathering WSNs is proposed. The algorithm differs from previous work in that it dynamically adapts the sampling schedule based on the observed internode data correlation as well as the temporal correlation. The performance of the algorithm has been assessed using real-world datasets. For two-tier networks, the proposed algorithm outperforms a highly cited previously published algorithm by up to 512% in terms of lifetime and by up to 30% in terms of prediction accuracy. For multihop networks, the proposed algorithm improves on the previously published algorithm by up to 553% and 38% in terms of lifetime and accuracy, respectively.

1. Introduction

Wireless sensor networks (WSNs) consist of nodes which detect and track real-world quantities [1]. Nodes are autonomous and are able to self-organize into intelligent networks. Each node consists of a microcontroller, memory, a radio transceiver, and sensors. Most WSN nodes are battery powered. The limited supply of energy means power consumption is a major issue in WSNs. In most applications, the radio transceivers are the largest consumers of energy [2]. Consequently, much research has been conducted on reducing the amount of time that the radio is on [3–5].

An important application area for WSNs is environmental monitoring [1]. Environmental monitoring applications require that a physical quantity is periodically measured and the measurements are relayed across the network to the base station, or sink, for processing. In many cases, the base station must maintain an up-to-date (online) view of the physical quantity being measured. Thus measurements must be transferred to the sink as soon as they are available

[6–8]. WSN measurements of data, such as temperature, humidity, air pressure, wind speed, nitrogen dioxide, and light, often exhibit internode data correlation and strong temporal correlations between different sampling times at the same node [9–12]. Knowledge of these correlations can be exploited to reduce the number of measurements needed to meet the application-specific sensing accuracy requirements.

Figure 1 shows temperature readings taken from two nodes in an environmental monitoring deployment in a university campus. The figure shows that the data is correlated between the two nodes. Also from 17:00 onwards, a strong temporal correlation begins to emerge in the data. Figure 2 shows the results when the number of transmitted samples is reduced by 25%, with every skipped sample being temporally predicted from previous readings. The results show that the temporal predictor shows good accuracy from 17:00 onwards. In Figure 3, rather than having node 1 transmit data samples, only the readings from 25% of the nodes within the network are used at any one time to predict the

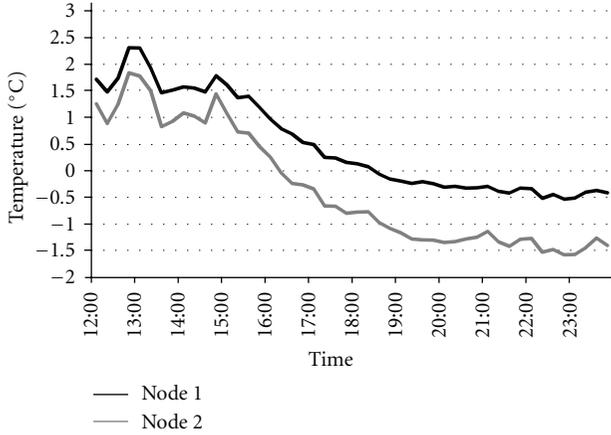


FIGURE 1: Temporal and internode data correlation of two nodes.

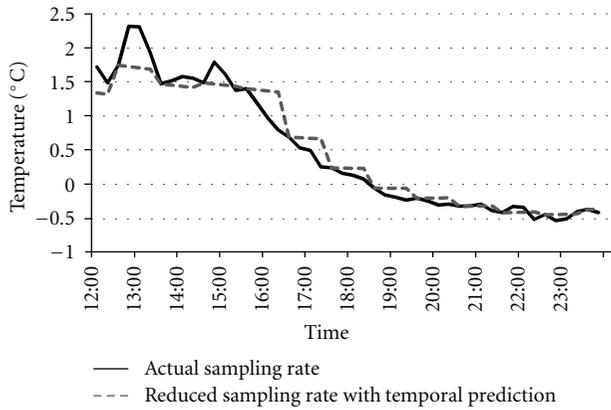


FIGURE 2: Performance of temporal predictor.

samples (internode data prediction). The results show that the internode predictor works well between 12:00 and 15:00. In this paper an algorithm which takes advantage of both temporal and internode correlation is proposed to reduce the number of transmitted samples at the cost of an application-specific acceptable error.

Clearly, there is a tradeoff between sensing accuracy and lifetime [13, 14]. In general, it can be said that improved accuracy requires collection and transmission of a greater number of sensor measurements which, in turn, means shorter network lifetime. The efficiency of a particular data collection schedule depends on the characteristics of the data being collected. These characteristics vary with time. Hence, the natural question arises: *for a given environmental monitoring application, how can the data gathering schedule be determined and dynamically adapted so as to maximize network lifetime while still meeting the application accuracy requirements?*

In this work, we propose a new adaptive scheduling algorithm for WSNs which can be used in environmental monitoring applications. The algorithm determines the sampling schedule based on user-specified accuracy goals, network connectivity, and a preliminary data collection phase. During preliminary data collection, data is collected

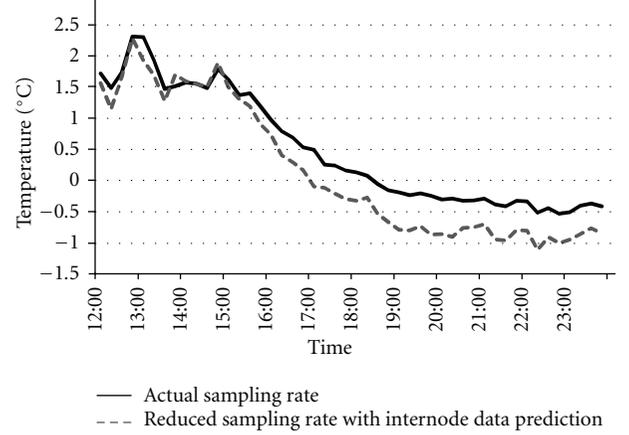


FIGURE 3: Performance of internode data predictor.

from all nodes at the full sampling rate. The preliminary data is divided into training and evaluation data sets. The training data is used to model the temporal and internode data correlation. The evaluation data is used to assess the performance of various candidate scheduling strategies. The models developed in the training phase are used to impute data which is not scheduled for collection according to the candidate strategy. The results of the imputation are compared with the measured data. The schedule which meets the user's accuracy requirements and maximizes network lifetime is deemed to be the most efficient and is applied to the network during the operational phase.

The algorithm supports schedule adaptation to allow for the time-varying nature of the data relationships. Firstly, the algorithm divides the day into a number of time periods or slots. A different subschedule is allowed in each slot. This allows the algorithm to adapt to the differing degrees of correlation present in the data at different times of the day, for example, midnight versus midday. Secondly, the accuracy of imputation is assessed during the operational phase. If the accuracy drops below the user-specific accuracy requirements, the slot is retrained and the subschedule updated. This allows the overall schedule to track long-term changes, such as the lengthening of daytime during spring.

The algorithm differs from previous work in that it supports dynamic adaptation of schedules. The algorithm supports subsampling and round-robin subsetting scheduling strategies. Variants of the algorithm are proposed for two-tier and multihop networks. The performance of the algorithm is assessed by simulation using real-world data sets. The algorithm is shown to significantly extend network lifetime when compared with a previously published scheduling algorithm. In terms of the round-robin subsetting algorithm proposed herein, it is different from coverage-based subsetting algorithms [15–17] in that it uses a data similarity metric rather than physical distance to measure correlation when forming subsets. The benefit of doing this is explained in Section 2.

The remainder of this paper consists of five sections. Section 2 describes related work. This is followed by an

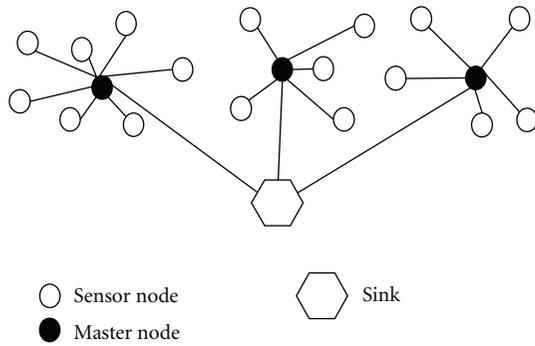


FIGURE 4: Two-tier network.

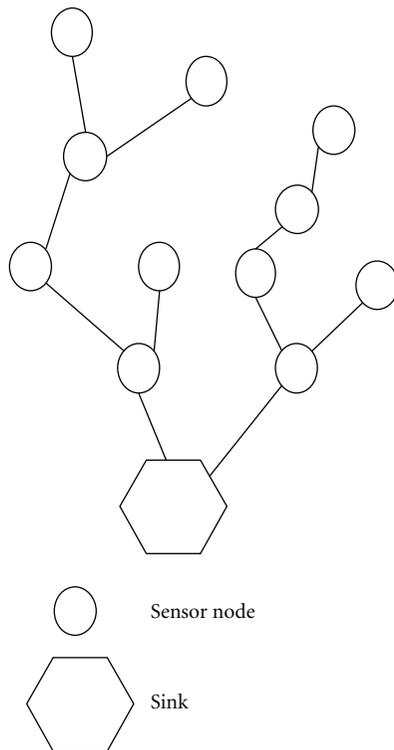


FIGURE 5: Multihop network.

explanation of the problem in Section 3. In Section 4, the proposed algorithm is described. In Section 5, the experimental method is described. In Section 6, the results and their implications are provided. Finally, the paper ends with conclusions.

2. Related Work

Two network topologies are commonly used for WSN applications: two-tier and multihop networks. Figures 4 and 5 show an example of a two-tier network and a multihop network. In the two-tier case, all battery-powered nodes have direct communication links with mains-powered nodes (master node) which can communicate data to the sink. In the multihop case, only the sink is mains powered and all communication must be routed to it via battery-powered

nodes. In the two-tier case, power consumption per node is proportional to the number of measurements per unit time. In the multihop case, power consumption per node is, in the conventional case, not proportional to the number of measurements per unit time, since the routing nodes must be on all of the time. However, in recent research, a number of authors have proposed cross-layer network protocols in which network availability is optimized so that it closely matches the application data transmission requirements [18, 19]. This approach, assumed herein, significantly reduces energy consumption and means that the power consumption per node is proportional to the number of measurements per unit time in the multihop case as well.

The scheduling algorithm proposed herein is targeted at environmental monitoring applications in which all of the data is immediately sent back to the sink. Since all of the data is sent to the sink for data gathering purposes, it makes sense to use this data for centralized scheduling as well. This obviates the need for energy-inefficient intranode schedule negotiation and allows for exploitation of multihop data correlations. In addition, much more computationally complex scheduling algorithms can be used at the sink than can be performed on the nodes, further improving performance.

Reducing the amount of data gathered in a WSN can be done by subsampling or subsetting. Subsampling is the process of making measurements less frequently; for example, a subsampling ratio of 2 would increase node sampling periods from 1 minute to 2 minutes. Round-robin subsetting is the process of using only a proportion of the nodes at any one time in a round-robin fashion; for example, a subsetting ratio of 2 would mean that half the nodes are sampled in even-numbered minutes (1, 3, 5, etc.) and the other half are sampled in odd-numbered minutes (0, 2, 4, etc.). In both examples, the energy consumption of the network is halved. The level of accuracy in imputing missing data varies depending on the degree of temporal or internode data correlation. The algorithm proposed in this work uses both subsampling and round-robin subsetting.

A number of publications have dealt with subsampling [20–22]. In all cases, measurements are suppressed, that is, not transmitted, if they can be accurately predicted based on previous measurements. Data suppression can either be *a priori*, before the measurement is taken, or *posteriori*, after the measurement is taken. As will be seen, depending on the data set, sometimes subsetting outperforms subsampling and sometimes vice versa. Hence the proposed approach supports both subsetting and subsampling.

Several publications have proposed algorithms for subsetting. These algorithms can be classified according to whether the subsetting decision is made based on the geographical coverage of the nodes or based on the data sensed by the nodes. Coverage-based schemes attempt to schedule nodes such that the entire area of interest is covered by the fewest sensor nodes [15–17]. The difficulty with this approach is that when obstacles are present within the area being monitored, sensor readings will not be well correlated with location [23]. In such cases the predominantly assumed

TABLE 1: Previous algorithms: main features.

Algorithm	Reference	Two-tier	Multihop	Centralized/distributed	Round-robin subsetting	Subsampling	Adaptive scheduling
CAG	[19]	×	✓	Distributed	×	✓	×
GUPTA	[18]	×	✓	Semidistributed	×	×	×
KEN	[24]	×	✓	Distributed	×	✓	×
SeReNe	[29]	×	✓	Centralized	×	×	×
RRC	[25]	✓	×	Centralized	✓	×	×
SS-MH/SS-2T	Proposed Method	✓	✓	Centralized	✓	✓	✓

disc-shaped sensing radius no longer holds true. For example, two sensors may be close together but be on different sides of a wall. In addition, node location information may not be readily available. Hence, in this work, we focus on data-similarity-based approaches. Another benefit of using a data similarity/correlation approach is that it can detect correlation changes in the environment over a long period of time. In this paper it is shown that as internode data correlations change, remodeling/retraining has to be done to maintain high accuracy in data imputation.

A number of methods have been proposed for subsetting based on data similarity. These methods can be grouped according to whether they use a centralized or distributed approach. In the centralized approach, the sink determines the sampling schedule whereas in the distributed approach, the nodes themselves decide on the subsets. The disadvantage of the distributed approach is that, if subsets are large, initializing and maintaining them requires a significant amount of internode communication, as in KEN [24]. As a consequence, contour maps and CAG [19] limit the range of subsets to one hop. The disadvantage of this is that long-distance correlations cannot be exploited. Furthermore these subsetting algorithms do not use a round-robin scheme thus achieving poor load balancing.

Herein we compare the proposed approach with the algorithm (which is named GUPTA in this paper) described in [18]. The GUPTA algorithm uses a data-driven approach, and two-tier and multihop versions are described. Unlike the algorithm proposed herein, the GUPTA method does not consider temporal correlations, adaptive scheduling, load balancing, or slotted scheduling. In the multihop version, the GUPTA algorithm is semidistributed because even though nodes make individual decisions whether to join a subset, it requires a centralized data gathering phase in order for all the nodes to gather training data from their neighbors.

In order to achieve load balancing for two tier networks, two systems have been previously proposed which incorporate round-robin subsetting [25, 26]. The system proposed in [25] converges slowly, forming multiple clusters before finding a satisfactory solution. This means that the system produces a significantly higher number of schedules thus making it difficult to maintain. The system described in [26] was developed by the authors of this paper as a prototype. The version described in this paper has a number of improvements. In addition to that we propose a novel

network optimized load-balanced subsetting for multihop networks.

Two systems have been previously described which use both subsetting and subsampling-KEN [24] and contour maps [27]. Unlike the proposal described herein, these algorithms do not perform any network level optimization, in the sense that nodes will still have to switch on their radios periodically to listen for packets as well as to relay packets even when they have no readings to send. Furthermore round-robin subsetting is not used.

Combining statistical WSN data models with probabilistic queries to improve the cost-effectiveness of WSN queries was investigated in the BBQ system [28]. However, BBQ focuses on multiple one-shot queries over the current state of the network, rather than continuous data gathering. In [29] SeReNe, a scheduling algorithm for answering queries is proposed. Similar to BBQ and the proposed method herein, it first gathers historical sensor readings. Through clustering SeReNe builds a subset of representative nodes to answer queries. The disadvantage of that is that for long-term queries SeReNe does not employ a round-robin scheme to achieve load balancing. In [30] the originators of SeReNe briefly discuss possible ways to adapt the model over a long period of time, but this was not evaluated. KEN uses data models as well to answer queries. KEN and SeReNe are similar in the sense that they are push-based methods whereas BBQ is a pull-based method. Herein, the user sets a probabilistic accuracy target *a priori* and possible schedules are assessed with respect to the target prior to their application.

A comparison between the various data-similarity-based scheduling algorithms that have been proposed is provided in Table 1. The algorithm proposed herein is the first to support schedule adaptation and round-robin subsetting.

3. Problem Statement

The goal of the scheduling algorithm is to determine the network sampling schedule which minimizes network communication for the worst-case node while ensuring that application-level accuracy requirements are met. The reason for minimizing communication by the worst-case node is to maintain load balancing thus enabling the network to continuously gather data from all nodes within the network continuously for a longer period of time. Even though sensor

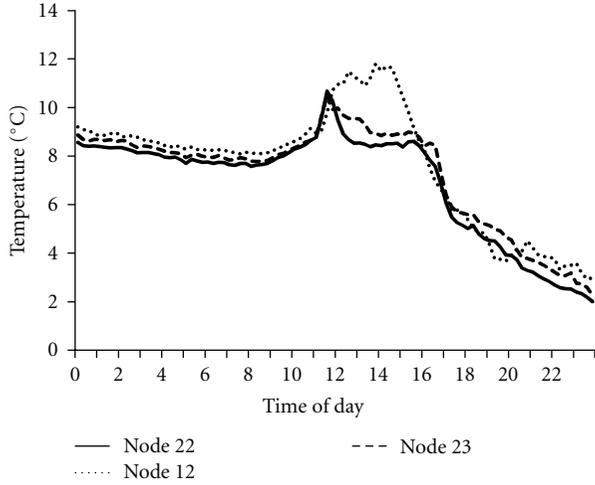


FIGURE 6: Data relationship of three nodes (temperature-LUCE deployment).

data of dead nodes can still be imputed, because the node is dead, validation and retraining of the predictors cannot be done when needed.

The user defines the accuracy requirement by setting a limit on the average probability (P_{lim}) of errors greater than a specified threshold (E_{lim}). For example, the user might require that 95% of reported measurements have an error of less than 0.5°C . In the case of measured values, the error e is equal to zero. In the case of imputed values, the error may be greater than zero. The goal of the algorithm is then to determine the schedule S_{ch} which minimizes the number of packets N_p transmitted by the worst-case node such that the probability $p(e)$ of errors less than E_{lim} is greater than P_{lim}

$$S_{ch} : \min(N_p) \quad \text{s.t. } p(e < E_{lim}) > P_{min}. \quad (1)$$

As stated previously, data correlations can be exploited in order to impute the missing values. In most previous work, these correlations are assumed to be static. Figure 6 shows the variation of temperature at three nodes over a day in a real-world dataset. Clearly the rate of change and internode data correlations are dependent on the time of day. Thus a scheduling algorithm should account for the fact that data correlations drift during the day and, for best performance, should use different subschedules at different times of the day. In addition, over long periods of time, temporal and internode data correlations can vary. Thus, imputation becomes less accurate. This deterioration in performance should be detected and the models retrained.

When subsetting, it is desirable that the subsets are disjoint and operate in a round-robin fashion so that the network is load balanced. Disjoint subsets are subsets such that for any two subsets C_i and C_j , $C_i \cap C_j = \phi$; that is, every node belongs to only one subset. In the two-tier case, determining disjoint subsets which provide accurate imputation of environmental conditions at all nodes is nontrivial. In the multihop case, the problem is more complex since every disjoint set must provide a representative

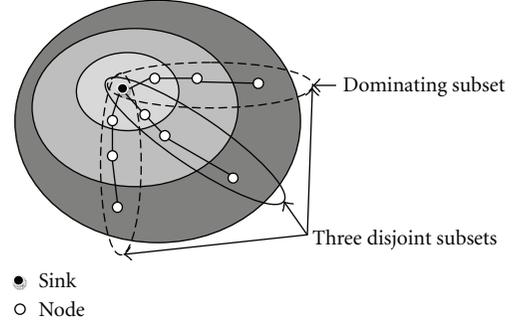


FIGURE 7: Disjoint subsetting example.

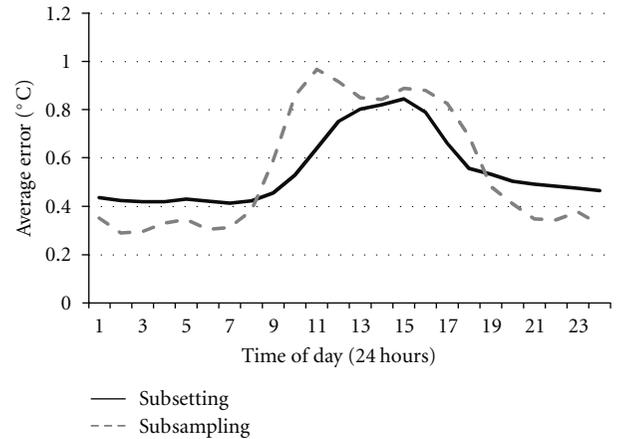


FIGURE 8: Performance of subsetting and subsampling averaged over 105 days.

node to represent each correlated region while also ensuring connectivity between all the nodes in the subset and the sink. For example, the three disjoint subsets in Figure 7 allow both load-balanced subsetting and continuous connectivity while having each correlated region represented by a node.

Figure 8 shows the performance of subsetting and subsampling with 75% of the data being predicted. Both methods are explained in detail in the following section. The figure shows that both algorithms perform well in the morning and at night. During the afternoon, both algorithms experience a significant loss in performance. Thus, on average, even if the accuracy of the method meets the user's requirements initially, it does not mean that the requirements are met throughout the day. To ensure user requirements are met, the amount of data being predicted during the afternoon has to be decreased. This can be done by reducing the subsampling/subsetting ratio.

4. Proposed Algorithm

In this section we explain the proposed slotted scheduling algorithm with variants for two-tier (SS-2T) and multihop (SS-MH) networks. The following sub-sections provide an overview of the algorithm; explain how schedules are

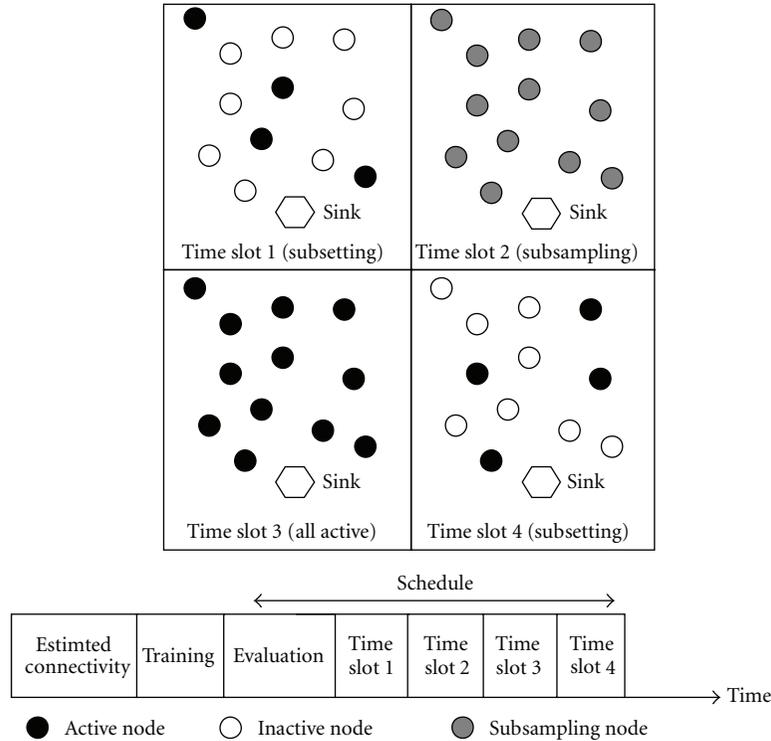


FIGURE 9: Slotted-scheduler timeline and network activity.

defined; describe how data imputation is performed; explain node-to-subset allocation for round-robin subsetting in both two-tier and multihop networks; explain the schedule selection process and detail the schedule update method.

4.1. Overview. Initially, the slotted scheduler gathers training and evaluation data and, in the multihop case, connectivity information from the network. During training and evaluation data collection, all nodes collect data at the user-specified maximum collection rate and transmit this data back to the sink. At the sink, the training data is used to build models for data imputation. The data from the evaluation phase is then used to assess the performance of various candidate scheduling strategies, that is, various ratios of subsetting and subsampling. The subschedule which meets the user's accuracy requirements and minimizes energy consumption is selected for application to the network in that slot during the operational phase. The selected data collection schedule is transmitted from the sink to the nodes. The network then enters the operational mode and data is collected according to the schedule. Data collected is monitored in order to detect changes in temporal/internode correlation. If changes are detected, the network reenters the training and evaluation phases in order to update the models and schedule.

Figure 9 illustrates how the slotted scheduling algorithm operates. The figure shows a 4-slot schedule with subsetting, subsampling, full rate collection and subsetting in the first, second, third, and fourth slots, respectively. The figure also shows the temporal sequencing of the establishment, training, evaluation, and operational phases. The operational phase is divided into a series of slots which repeats.

Figure 10 shows how subsetting or subsampling is chosen for each time slot. The temperature data plot, taken from the evaluation data, shows that between the times of 7:00 and 17:00 there is limited temporal correlation. The data does show internode data correlation during this period; thus, subsetting is selected for that time period. During this time, nodes 1 and 2 exhibit strong correlation and nodes 3 and 4 show strong correlation. To ensure each inactive node is represented within the active subset by a strongly correlated node, the subsetting algorithm groups nodes 1 and 3 as one subset and nodes 2 and 4 as the other subset. These two subsets are used in a round-robin fashion during the operational phase, as shown in the figure. From 17:00 onwards, the data begins to show strong temporal correlations. During this period, the temporal predictor performs more accurately. Thus subsampling is chosen for use by the scheduler. The figure shows that after 17:00 all nodes are activated for one time slot, and for the subsequent two time slots, nodes are inactive and data is temporally predicted.

4.2. Schedule Description. The schedule is based on the user-specified default data collection period. This is the maximum rate at which data can be collected, that is, with no subsampling or no-subsetting applied. The schedule is divided into a number of slots, or time periods, which span the day. A different subschedule can be specified for each slot. This allows the scheduler to adjust the data collection rate depending on time of day. For example, in a schedule with eight slots, each slot would last for four hours: slot 0 from midnight to 4 AM, slot 1 from 4 AM to 8 AM and so on.

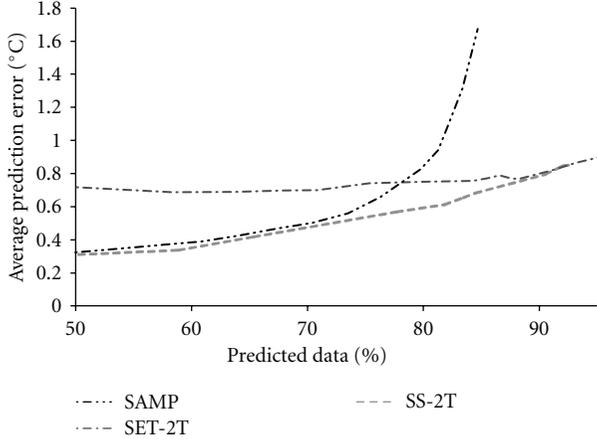


FIGURE 11: Variation in mean imputation error with percentage of noncollected data.

the sink. Piggybacking and compression schemes can be used to reduce this overhead. Data collection timing can be maintained using node wake-up synchronization [32].

Herein, we refer to data which is scheduled for collection as collected data and data which is not scheduled for collection as noncollected data. Noncollected data must be imputed based on collected data.

4.3. Data Imputation. In the case of subsampling, imputation is performed using linear prediction (LP). The linear predictor determines the coefficients of a forward linear predictor by minimizing the prediction error in the least squares sense based on the training data. During the operational phase, LP is used to estimate the noncollected data as a weighted sum of previous measurements obtained at the same node

$$x_i(i, t) = a(i, 1)x_o(i, t - r) - a(2)x_o(i, t - 2r) - \dots - a(p)x_o(i, t - pr), \quad (2)$$

where $x_i(i, t)$ is the current imputed sample at node i at time t , $x_o(i, t - r)$ is the observed (measured) data at node I at time $t - r$, $a(i)$ are the coefficients of the linear predictor, r is the subsampling ratio, and p is the length of the predictor.

In the case of subsetting, only one subset of the network is collected in each data collection round. Given that subset C_i is the operating subset consisting of the nodes s_1, s_2, \dots, s_L , then the predicted value of a node is

$$X_p = \sum_{l=1}^L \alpha_l s_l. \quad (3)$$

Given that the training data for a single node and the remaining nodes is o and O , respectively, then the weighted coefficients are

$$[\alpha_1, \alpha_2, \dots, \alpha_L]^T = (O^T O)^{-1} O^T o. \quad (4)$$

4.4. Round-Robin Subsetting. To achieve load balancing, every node in the network is allocated to a subset and the number of nodes per subset is constant. The key to accuracy is in allocating the nodes such that every subset contains a set of nodes which accurately represent environmental conditions over the whole network. Novel algorithms have been developed to solve the node allocation problem for subsetting in two-tier and multihop networks.

4.4.1. Two-Tier Networks. In the two-tier case, node-to-subset allocation is achieved by node clustering, followed by subset allocation, and allocation optimization.

Initially, nodes are clustered based on data similarity. Nodes are clustered using a normalized cut (N-cut) clustering algorithm [33] based on an entropy S metric. In this way, nodes with strong data relationships are put in the same cluster

$$S(i, j) = \ln\left(\sqrt{(2\pi e)^2 |\Sigma|}\right), \quad (5)$$

where Σ is the covariance matrix of data obtained from nodes i and j .

After clustering, node allocation is performed. The first node subset is formed by selecting one representative node from each cluster. In this way, the subset consists of nodes which represent the measurements in each cluster. The representative node is chosen as the node with the minimum total entropy S_{\min} within the cluster

$$S_{\min} = \min(S(i, j)), \quad (6)$$

$$\forall i \in \{1, \dots, N_c - 1\}, \quad \forall j \in \{1, \dots, N_c - 1\}, \quad i \neq j,$$

where N_c are the nodes within the cluster, i is the current, node id and j is the id of the other node.

The second subset is found by excluding the already allocated nodes from the set of available nodes and repeating the representative node selection step. This process is repeated until all of the nodes in the network are allocated to a subset.

The sequential subset allocation process can lead to poor results as the subsets allocated later in the process tend not to perform as well as those allocated earlier in the process. To address this, a genetic algorithm (GA) is applied to optimize the node allocation. First, two subsets are picked at random. Second, one node is chosen from each subset and they are swapped. Third, if the swap causes the sum of the entropy of the two subsets to increase, then the swap is made permanent; otherwise, the subsets revert back to their original states. The full subsetting algorithm is described in Algorithm 1.

Subset allocations and models are generated in this way for a range of subsetting ratios. The allocations are saved for later evaluation, see Section 4.5.

4.4.2. Multihop Networks. In the multihop case, allocation of nodes to subsets is performed in a different way. This is because, in multihop networks, all subsets must provide connectivity between all nodes in the subset and the sink. The algorithm works by growing the maximum number of

```

X = All sensor nodes
i = 1
while X! = ∅ do
  Cluster nodes
  Pick representative node from each cluster
  Ci = Chosen representative nodes
  X = X - Ci
  i ++
end
n = number of runs for Genetic Algorithm
while count! = n do
  Pick two random subsets Cp and Cq
  Stotal = SavgCp + SavgCq
  Cpold = Cp
  Cqold = Cq
  Swap a random node from Cp and Cq
  Stotalnew = SavgCp + SavgCq
  if Stotal > Stotalnew
    Cp = Cpold
    Cq = Cqold
  end
  count ++
end

```

ALGORITHM 1: Pseudocode for two-tier round-robin subset allocation.

subsets from the sink based on connectivity information and data similarity.

Using distance criteria, the algorithm determines which nodes are one hop away from the sink. Nodes which are one hop from the sink each form the root of a new subset. Thus the number of new subsets found is directly proportional to the distance criteria. Larger distance criteria will yield a larger number of subsets. The subsets are grown by selecting the nodes according to the following criteria:

- (i) being 1 hop away from a node currently in the subset,
- (ii) having the highest difference in average entropy between the nodes within the subset.

The subsets are grown in a round-robin fashion. If a subset cannot be grown, then the method continues growing the other subsets. Once the maximum number of subsets has been formed, the method then combines subsets in order to form larger subsets which are better spread over the network. The average difference in entropy between all subset pairs is found. Subsets with the greatest difference are combined. This step is repeated until all subsets have been combined. At each step, the subset allocation is saved for later evaluation, as described in the next sub-section. The multihop subsetting algorithm is fully described in Algorithm 2.

4.5. Selecting the Best Schedule. The performance of all possible subsampling and subsetting strategies is assessed for each slot. The subsampling or subsetting subschedule giving the best performance is selected for application to the network in that slot during the operational phase.

```

X = All sensor nodes
TL = Transmission Range Limit
C subsets are formed one for each node xn within the
TL of the sink
Nc = number of subsets (equivalent to number of 1 hop
nodes from the sink)
i = 1
X = X - C
while X! = ∅ do
  if i > Nc then
    i = 1
  end
  Pick node xn which is 1 hop from Ci and has
  highest average Entropy with Ci
  Ci = Ci + x
  X = X - xn
  i ++
end
Save C
while Nc > 2 do
  Combine each subset based on Entropy
  Nc = new number of subsets
  Save C
end

```

ALGORITHM 2: Pseudocode for multihop round-robin subset allocation.

The various sub-scheduling options are assessed using the evaluation data. In each case, the noncollected data is imputed and the result compared to the measured data to give the imputation error e

$$e(i, t) = \text{abs}(x_i(i, t) - x_o(i, t)). \quad (7)$$

The standard deviation of the error σ calculated over the whole network during the evaluation period is calculated. This is compared to the error target specified by the user. The target standard deviation of the error is calculated by projecting the target error limits (percentage of errors greater than threshold) onto a Gaussian probability distribution and finding the equivalent standard deviation σ_{lim} . Subschedules that lead to error standard deviations in excess of the target $\sigma > \sigma_{\text{lim}}$ are rejected. Since the schedules are load-balanced by construction, the energy consumption of routing is equal in all cases. Thus, the energy consumption is proportional to the number of collected measurements. Therefore, the remaining subschedule with the least number of measurements is selected for application to the network. The final schedule is determined by concatenation of the selected subschedules. If appropriate, the schedule can be compacted by merging consecutive subschedules that are the same, provided that the slot lengths remain equal.

4.6. Schedule Update. During the operational phase, the algorithm monitors the accuracy of the temporal and internode data imputation models. This allows the system to determine if the data characteristics have drifted since the models were last trained. This is done by comparing the

Method	Ratio																								
SET-2T (sub-setting)	1:7	[Grid with black cells at (0,1), (0,2), (0,3), (0,4), (0,5), (0,6), (0,7), (0,8), (0,9), (0,10), (0,11), (0,12), (0,13), (0,14), (0,15), (0,16), (0,17), (0,18), (0,19), (0,20), (0,21), (0,22), (0,23)]																							
	1:5	[Grid with black cells at (4,1), (4,2), (4,3), (4,4), (4,5), (4,6), (4,7), (4,8), (4,9), (4,10), (4,11), (4,12), (4,13), (4,14), (4,15), (4,16), (4,17), (4,18), (4,19), (4,20), (4,21), (4,22), (4,23)]																							
	1:4	[Grid with black cells at (18,1), (18,2), (18,3), (18,4), (18,5), (18,6), (18,7), (18,8), (18,9), (18,10), (18,11), (18,12), (18,13), (18,14), (18,15), (18,16), (18,17), (18,18), (18,19), (18,20), (18,21), (18,22), (18,23)]																							
	1:3	[Grid with black cells at (18,1), (18,2), (18,3), (18,4), (18,5), (18,6), (18,7), (18,8), (18,9), (18,10), (18,11), (18,12), (18,13), (18,14), (18,15), (18,16), (18,17), (18,18), (18,19), (18,20), (18,21), (18,22), (18,23)]																							
	1:2	[Grid with black cells at (18,1), (18,2), (18,3), (18,4), (18,5), (18,6), (18,7), (18,8), (18,9), (18,10), (18,11), (18,12), (18,13), (18,14), (18,15), (18,16), (18,17), (18,18), (18,19), (18,20), (18,21), (18,22), (18,23)]																							
SAMP (sub-sampling)	1:7	[Grid with black cells at (9,1), (9,2), (9,3), (9,4), (9,5), (9,6), (9,7), (9,8), (9,9), (9,10), (9,11), (9,12), (9,13), (9,14), (9,15), (9,16), (9,17), (9,18), (9,19), (9,20), (9,21), (9,22), (9,23)]																							
	1:5	[Grid with black cells at (9,1), (9,2), (9,3), (9,4), (9,5), (9,6), (9,7), (9,8), (9,9), (9,10), (9,11), (9,12), (9,13), (9,14), (9,15), (9,16), (9,17), (9,18), (9,19), (9,20), (9,21), (9,22), (9,23)]																							
	1:4	[Grid with black cells at (9,1), (9,2), (9,3), (9,4), (9,5), (9,6), (9,7), (9,8), (9,9), (9,10), (9,11), (9,12), (9,13), (9,14), (9,15), (9,16), (9,17), (9,18), (9,19), (9,20), (9,21), (9,22), (9,23)]																							
	1:3	[Grid with black cells at (9,1), (9,2), (9,3), (9,4), (9,5), (9,6), (9,7), (9,8), (9,9), (9,10), (9,11), (9,12), (9,13), (9,14), (9,15), (9,16), (9,17), (9,18), (9,19), (9,20), (9,21), (9,22), (9,23)]																							
	1:2	[Grid with black cells at (9,1), (9,2), (9,3), (9,4), (9,5), (9,6), (9,7), (9,8), (9,9), (9,10), (9,11), (9,12), (9,13), (9,14), (9,15), (9,16), (9,17), (9,18), (9,19), (9,20), (9,21), (9,22), (9,23)]																							
All nodes	1:1	[Grid with black cells at (9,1), (9,2), (9,3), (9,4), (9,5), (9,6), (9,7), (9,8), (9,9), (9,10), (9,11), (9,12), (9,13), (9,14), (9,15), (9,16), (9,17), (9,18), (9,19), (9,20), (9,21), (9,22), (9,23)]																							
	Time	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

FIGURE 12: Schedule for ST LUSE dataset, target error of 1.4°C in 80% of cases.

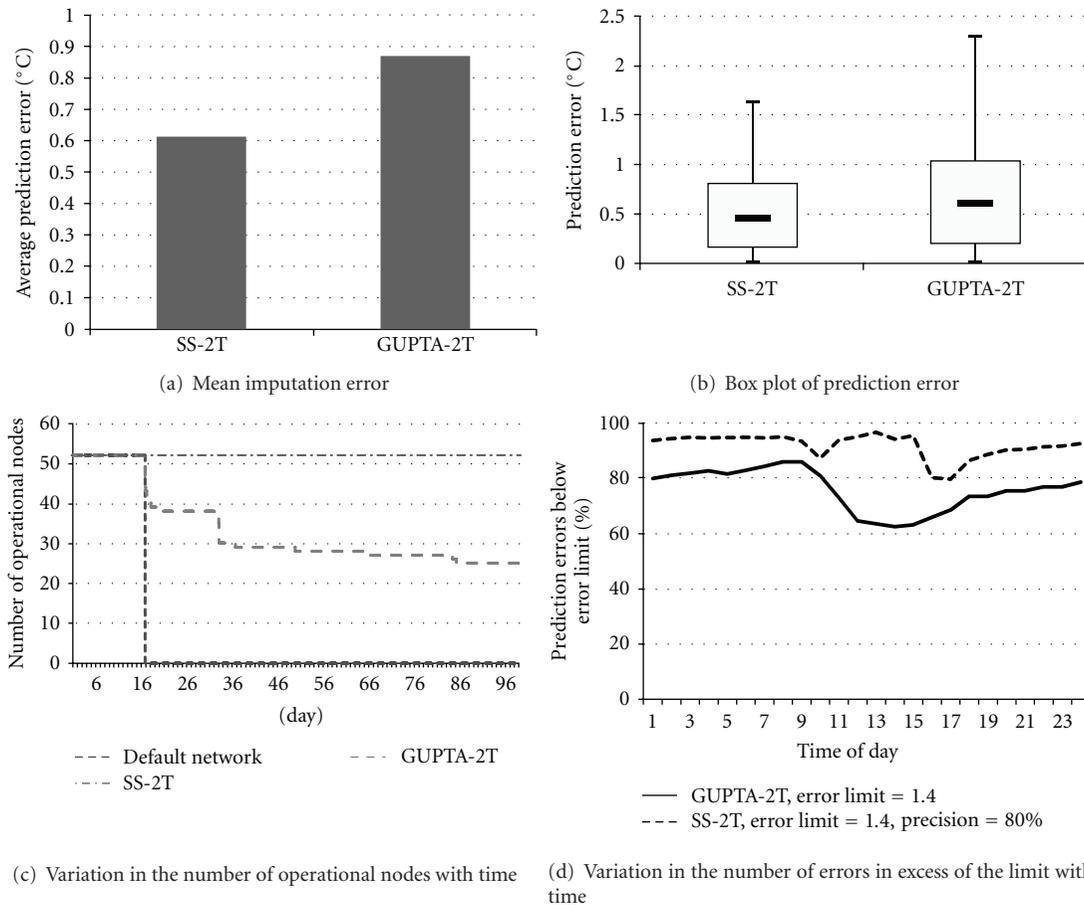


FIGURE 13: Performance of scheduling algorithms, ST LUCE dataset, two-tier network.

prediction accuracy seen when the training and evaluation data is used compared to the prediction accuracy seen with the current received sample.

In the case of the temporal model, the model is tested by predicting the current received sample and testing it with the last received sample (which is γ samples away). This prediction is done using (2). The error is found between

the current received sample and the predicted sample. Next using only evaluation data, the data from the same time slot is predicted with data which is γ samples away. A comparison is done between the error found using the current received sample and the error found using the evaluation data. A node is marked when the error difference is above a threshold

limit. When the percentage of marked nodes is above T_{lim} for a duration of D_{lim} days then retraining is triggered.

For the internode model, it is first tested using the current samples received from the nodes of the current operational subset C_i . Using (3) each received sample is imputed using the other received samples at that particular time slot. The error between the predicted value and the actual value for each sample is found. The error results are then compared with the results when the same test is repeated on the evaluation data using the same time slot and the same subset of nodes C_i . A node is marked when the difference between the prediction error using current received samples and the prediction error using evaluation data is above a certain threshold. Similar to the temporal model test, when the limits of T_{lim} and D_{lim} are broken, retraining is commenced.

5. Experimental Method

The algorithm described in the previous section was implemented in Matlab and tested on two datasets taken from the Lausanne Urban Canopy Experiment (LUCE) [34]. Table 2 provides a summary of the datasets.

Results were evaluated in terms of mean imputation error (see (7)), percentage of noncollected data, variation of the number of operational nodes with time, and network lifetime. Mean imputation error is the mean error of the imputed noncollected data. The percentage of noncollected data (*PND*) is related to the amount of data transmitted and thus to the lifetime of the network. The percentage of data collected and transmitted to the sink is $100\% - PND$. We compare the results for different systems in terms of two definitions of lifetime. The first definition of lifetime is $L_{100\%}$ which is the length of time for which all nodes are alive. The reason for choosing this metric is because when the first node dies, this node can no longer be used for retraining. Thus if the node data correlation with other nodes changes, this cannot be corrected thus rendering the imputed readings from the other nodes void. The second definition of lifetime is $L_{50\%}$ which is the length of time for which 50% or more of the nodes remain alive.

Scheduling algorithms such as [32, 35] reduce idle listening significantly through the proper use of schedules. Such algorithms make the power consumption of sensor nodes closely proportional to the number of transmitted packets. For each simulation done, each sensor node is initialized with a limited battery power. Every transmitted packet is set to consume 1 unit of battery power. We assume the network allows piggybacking thus ensuring that even in the multihop case only a single packet is transmitted by each node during each sampling cycle. Similar assumptions were made in [18].

The performance of the proposed algorithm is compared to that of the default network and to the GUPTA algorithm. In the default network, every node collects data every collection round; that is, all data is collected.

There are two variants of the GUPTA algorithm used herein. GUPTA-2T for two-tier networks and GUPTA-MH for multihop networks. For the GUPTA algorithm, initially when the correlation structure is unknown, all the network

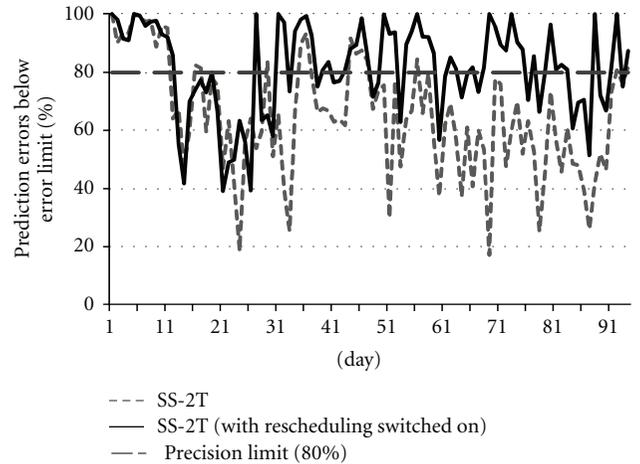


FIGURE 14: Performance of SS-2T with rescheduling on and off, ST LUCE dataset, two-tier network.

```

Input: A sensor network with a correlation graph
Output: A correlated dominating set M
BEGIN M = node  $s_i$  with largest IM
while  $B(\text{new } M, M) > 0$  do
    Pick  $S_i$  for which  $B(M \cup S_i, M)$  is maximum
end

```

ALGORITHM 3: GUPTA centralized algorithm.

nodes are periodically involved in transmitting data to the data-gathering node using a communication tree. Using this setup, each node then collects data from its d -hop neighbors using a piggyback scheme. In GUPTA-MH simulations, each node collects 3-hop neighborhood information.

GUPTA-2T algorithm proposed in [18] is used on a multihop network. In [18] during each iteration the number of nodes which can join the connected correlation-dominating set (CCDS) is bounded by the number of hops. As the GUPTA-2T algorithm used herein is used on a two-tier network, the algorithm is no longer bounded by hop count. The benefit of this is that there is a wider selection of nodes which can be added to the operating dominating set.

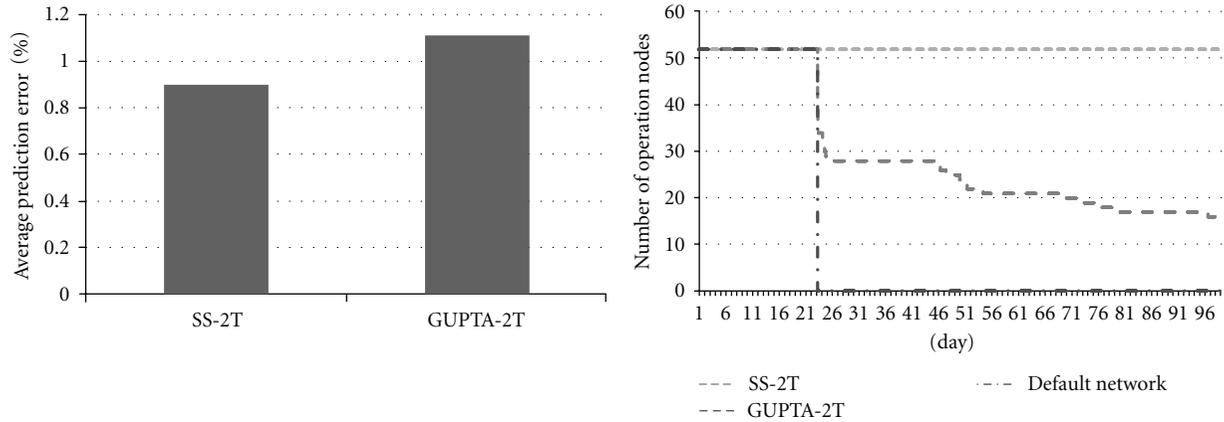
The GUPTA-2T algorithm works by adding nodes which will give the most benefit to the dominating set. This is continuously done till there is no more benefit in adding nodes. Given that IM is the group of nodes which can be inferred by M and new IM the nodes which can be inferred by $M \cup s_i$ (s_i is any node not belonging to M), then the benefit function is $B(M \cup s_i, M) = \text{new IM} - \text{IM}$. The purpose of the benefit function is to maximize the number of inferred nodes thus maximizing the number of sleeping nodes. Pseudocode for GUPTA-2T is shown in Algorithm 3.

For GUPTA-MH, a node s with priority $p(s)$ is marked deleted if the following conditions are satisfied:

- (i) the node s has not been mark selected;

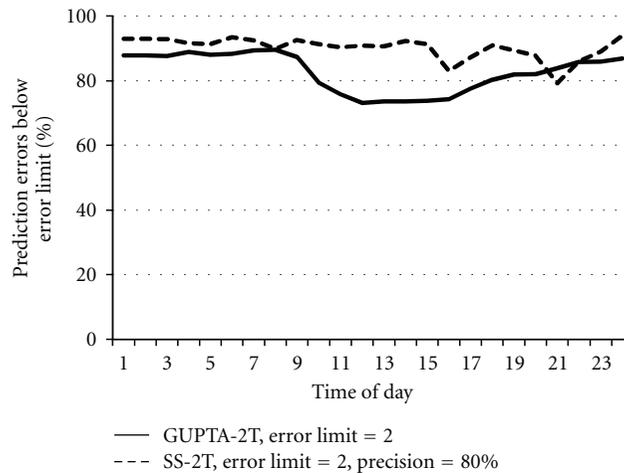
TABLE 2: Datasets.

Name	Date	Sampling period	Duration (Days)	Number of nodes	Percentage of missing data	Area
RH LUCE (relative humidity)	1/12/2006	15 minutes	112	52	9.79%	106600 m ²
ST LUCE (surface temperature)	1/12/2006	15 minutes	112	52	7.86%	106600 m ²



(a) Mean imputation error

(b) Variation in the number of operational nodes with time



(c) Variation in the number of errors in excess of the limit with time

FIGURE 15: Performance of scheduling algorithms, RH LUCE dataset and two-tier network.

- (ii) the connectivity of the communication subgraph is not affected by the deletion of the node s ;
- (iii) there is a correlation edge in the correlation graph such that every node in the set S is either marked selected or has a priority more than $p(s)$.

The score $p(s)$ is the sum of the number of nodes which are correlated with the node s . The more the nodes which can be predicted by s , the higher the $p(s)$.

The number of messages sent during training is not considered in the results as both algorithms require a training phase. For the GUPTA algorithm, the first 14 days are used to build the model. In the case of the proposed algorithm, the

first 7 days was used to build the internode/temporal model (training) while the subsequent 7 days was used to assess the performance of various subsetting and subsampling ratios (evaluation). It is assumed that the underlying network is able to handle packet loss. In the multihop case, two nodes are assumed to have connectivity if they are less than 135 meters apart.

In the case of adaptive scheduling, two days of data was used for rescheduling the nodes. The first day is used for a training phase while the second for the evaluation phase. Rescheduling was triggered if 60% T_{lim} of nodes are less than the user-specified error threshold for two days (D_{lim}). When testing, rescheduling nodes with more than 6% of

TABLE 3: Improvement in lifetime by slotted-scheduler (two-tier) and GUPTA (centralized) with respect to default network (RH LUCE).

Data Set	Algorithm	Error limit	Packet limit	Average prediction error	$L_{100\%}$	$L_{50\%}$
ST LUCE	GUPTA-2T	0.25	2150	0.61	0%	226%
ST LUCE	SS-2)	1.2	2150	0.42	226%	226%
RH-LUCE	GUPTA-2T	0.75	2150	1.1	0%	120%
RH-LUCE	SS-2T	2	2150	0.9	226%	226%

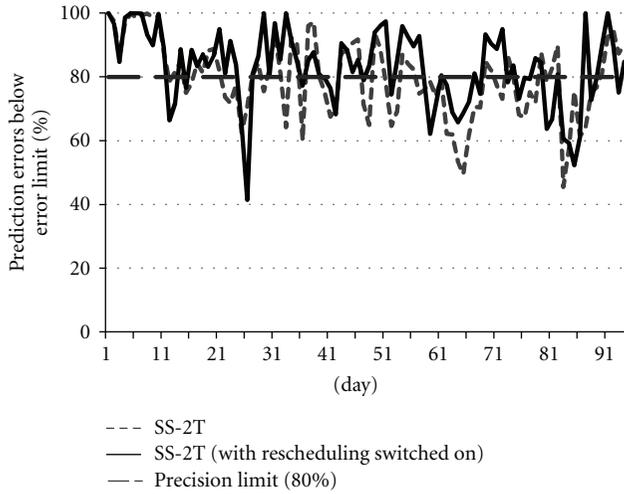


FIGURE 16: Performance of SS-2T with rescheduling on and off, RH LUCE dataset and two-tier network.

missing data were deleted from the dataset as this impeded rescheduling.

6. Results

This section is divided into two subsections covering the two-tier and multihop cases, respectively. In each section the proposed algorithm is compared with GUPTA, a previously published algorithm. Secondly, the advantages of using retraining with the proposed algorithm are shown.

6.1. Two-Tier Network. Firstly the performance of the various subsampling, subsampling, and imputation methods was assessed using the ST LUCE data set and a two-tier network. Figure 11 shows the variation in mean imputation error with the percentage of imputed data for various methods. Three methods were compared—subsampling (SAMP), two-tier subsampling (SET-2T), and the full slotted scheduler algorithm including both subsampling and subsampling (SS-2T). Rescheduling was switched off. For low imputation percentages, subsampling performs better than subsampling. For high imputation percentages, subsampling performs better than subsampling. The proposed slotted scheduling algorithm combines the advantages of subsampling and subsampling and performs best in all cases.

Figure 12 shows the schedule created by the slotted scheduler for an error limit of 1.4°C in 80% of cases. The figure shows that the choice between subsampling versus

subsampling as well as the ratio varies during the day depending on the data statistics. Between the times of 00:00 and 09:00 subsampling is scheduled for use. During that period, only one seventh of the nodes were scheduled to sample and transmit at each sampling period for the majority of the duration. From 09:00 till 17:00 (during the day), subsampling is used with a sampling ratio of 1:2. From 20:00 onwards, the scheduler reverts back to the use of subsampling.

The GUPTA and slotted scheduling algorithms were compared using an error limit of 0.25°C and of 1.2°C in 80% of cases, respectively. Rescheduling was switched off. Figure 13(a) shows the mean imputation error of both methods. In terms of prediction accuracy the slotted scheduler performs 29.5% better. A box plot of the prediction error is presented in Figure 13(b). The box plot clearly shows that in terms of the distribution of errors SS-2T performs better as well. Figure 13(c) shows the number of operational nodes over the duration of the simulation for both methods and for the default network. The packet limit was set to 2,150 packets. Using the GUPTA algorithm, nodes start to die much sooner than when using the proposed algorithm. Table 3 shows that in terms of prediction accuracy and lifetime SS-2T outperforms GUPTA-2T. Figure 13(d) shows how the percentage of errors that are in excess of the error limit varies across the time slots during the first week of operation. It can be seen that, for the GUPTA algorithm, the number of errors varies significantly over the slots. The proposed algorithm performs within the 80% precision limit (P_{lim}) for all time slots.

Figure 14 compares the performance of SS-2T with rescheduling on and off. The initial loss in performance in both cases (days 10–28) is due to the large amount of missing data in the dataset. The algorithm signals for rescheduling during the 11th day of operation but because of the lack of data it was not done till day 26. Overall, the algorithm with rescheduling switched on gives an average of 81% prediction errors which are less than the error limit, while without rescheduling 65% are less than the error limit. The version with rescheduling on requires 58% more packets than the algorithm without rescheduling. Even so, the number of packets transmitted by SS-2T with rescheduling on is four times less than the default network.

Figure 15 shows the results of performance assessment for the two-tier Slotted-Scheduler and the GUPTA algorithms using the RH LUCE dataset. For the GUPTA algorithm the error limit was set to 0.75°C . For the slotted scheduler the error limit and precision limit were set to 2% and 80%, respectively, and rescheduling was switched off. In both cases the packet limit was 2,150. As can be

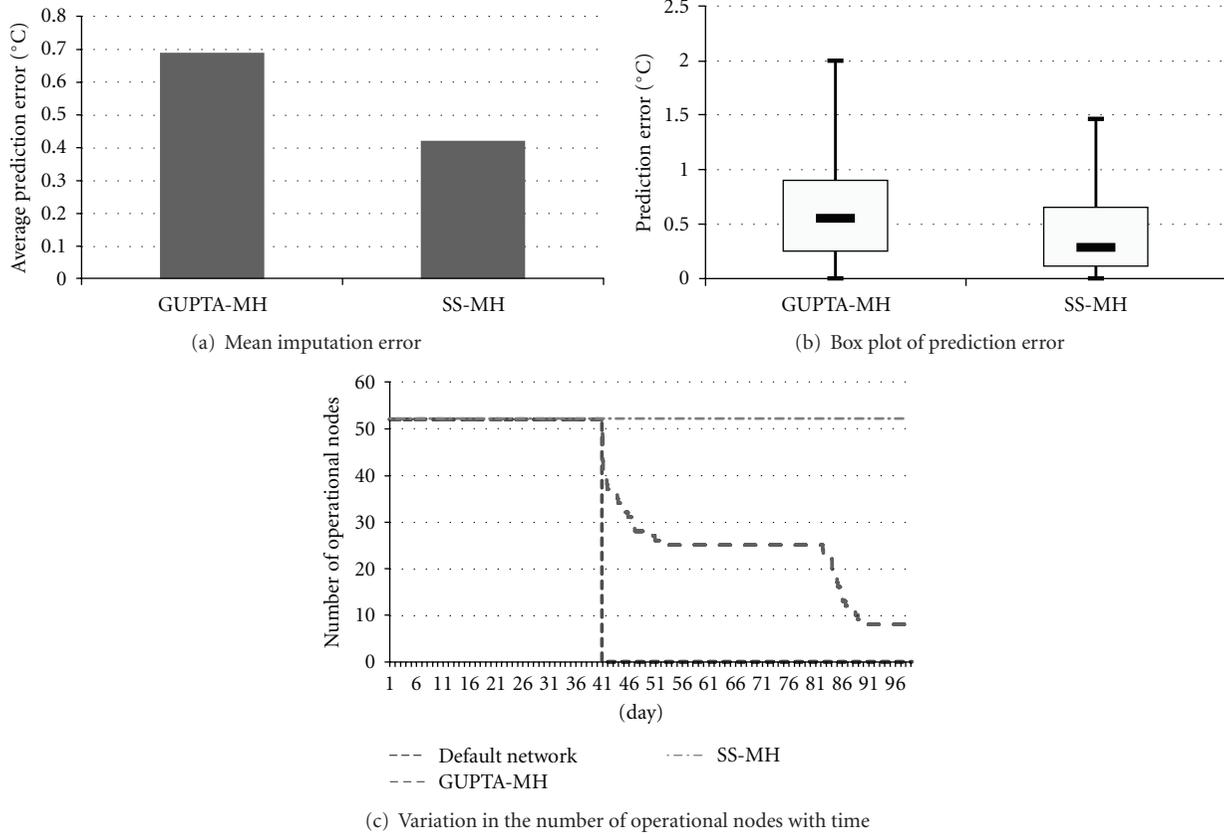


FIGURE 17: Performance of scheduling algorithms, ST LUCE dataset and multihop network.

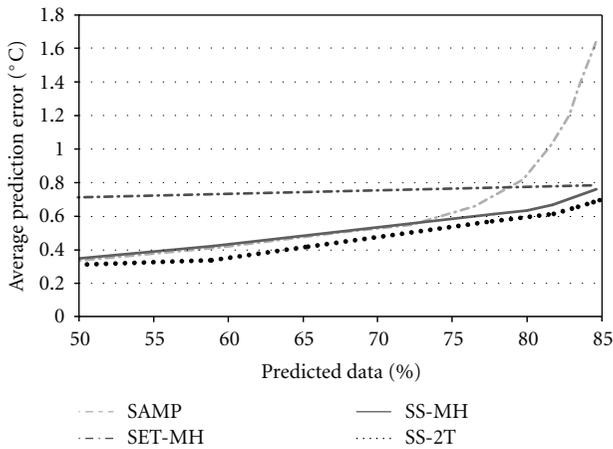


FIGURE 18: Variation of accuracy with percentage of noncollected data.

seen in Figure 15(a), the Slotted-Scheduler provides greater accuracy: 19% better than GUPTA. Figure 15(b) shows that the nodes running the GUPTA schedule die faster. As shown in Figure 15(c) SS-2T also performs within the 80% precision limit for all time slots during the first week of operation. Table 3 summarizes the results obtained.

Figure 16 compares the performance of SS-2T with rescheduling switched on. With rescheduling switched on, the average percentage of prediction errors below the error limit after day 45 is 80% while for rescheduling off it is 75%. In terms of transmitted packets, during the operational phase, the version with rescheduling transmitted 85% more packets than the version without. The re-scheduled version transmits three times less packets than the default network.

6.2. Multihop Network. Figure 17(a) shows the performance of the multihop algorithms for the ST LUCE dataset. The error limits for the GUPTA and slotted scheduler algorithms are 0.1°C and 0.9°C in 80% of cases, respectively, and rescheduling was switched off. The packet limit is 3,700. The accuracy of the slotted scheduler is 38% better than that of the GUPTA algorithm. In terms of the distribution of prediction error, Figure 17(b) shows that SS-MH performs better than the GUPTA algorithm. Figure 17(c) shows that the Slotted-Scheduler also performs better than the GUPTA algorithm in improving the lifetime in terms of both L_{100} and L_{50} . Table 4 summarizes the performance of the algorithms for the ST LUCE dataset for these precision settings and for one other setting. As can be seen, the Slotted-Scheduler performs within the user-specified error limit.

Figure 18 shows how the accuracy of the subsampling (SAMP), multihop subsetting (SET-MH), multihop slotted

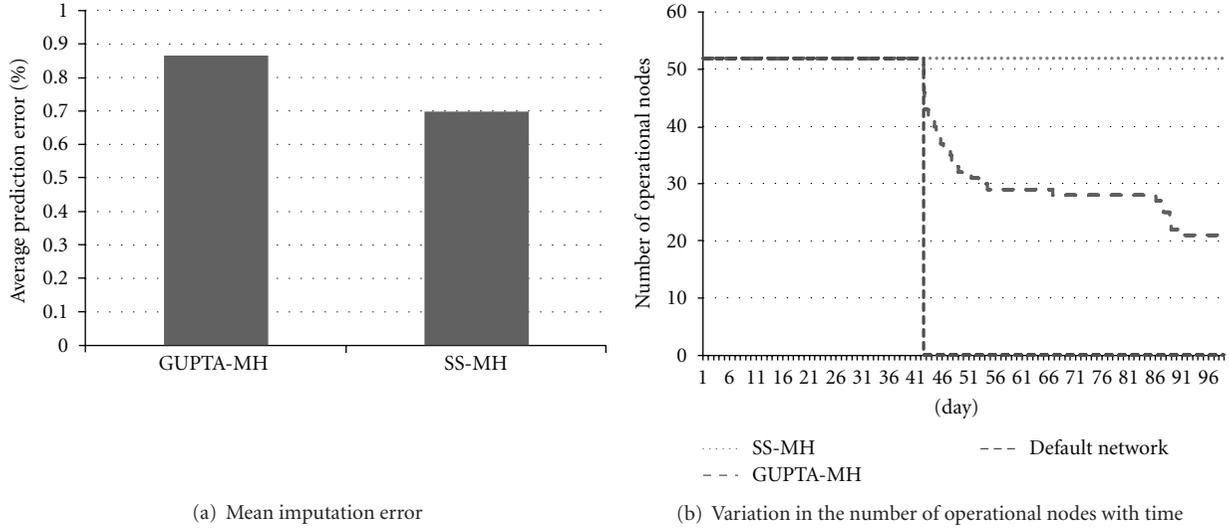


FIGURE 19: Performance of scheduling algorithms, RH LUCE dataset and multihop network.

TABLE 4: Improvement in lifetime by slotted-scheduler (Mhop) and GUPTA (distributed) with respect to default network.

Data Set	Algorithm	Error limit	Packet limit	Average prediction error	$L_{100\%}$	$L_{50\%}$
ST LUCE	GUPTA-MH	0.1	3700	0.69	0%	30%
ST LUCE	SS-MH	0.9	3700	0.42	145%	145%
ST LUCE	GUPTA-MH	0.5	1700	0.75	0%	233%
ST LUCE	SS-MH	1.8	1700	0.67	444%	444%
RH-LUCE	GUPTA-MH	0.1	3700	0.87	0%	107%
RH-LUCE	SS-MH	1	3700	0.70	133%	133%
RH-LUCE	GUPTA-MH	0.5	1400	1.17	0%	306%
RH-LUCE	SS-MH	3	1400	1.0	553%	553%

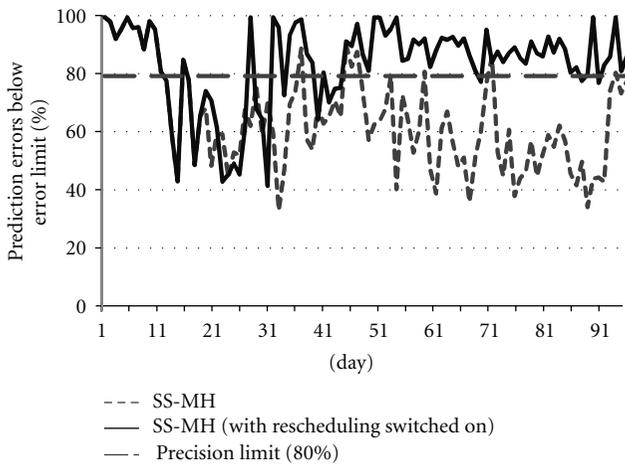


FIGURE 20: Performance of SS-MH with rescheduling on and off, ST LUCE dataset, multihop network.

scheduler (SS-MH), and two-tier Slotted Schedule (SS-2T, rescheduling off) varies with the percentage of imputed data for the ST LUCE dataset. Again, the slotted scheduler performs better than the subseting and subsampling algorithm.

The performance of the multihop slotted scheduler is similar to that of the two-tier algorithm, even though the subsets are constrained in that they must all provide connectivity to the sink for all nodes.

Figure 20 assesses SS-2T with and without rescheduling. Using rescheduling, the average percentage of prediction errors less than the threshold increases from 65% to 84%. This was achieved at the cost of an 83% increase in the number of packets. As in the two-tier case, even though the algorithm signaled a retrain on day 11, it was unable to perform the retrain for several days due to the amount of missing data.

Figures 19(a) and 19(b) show the performance of the multihop algorithms for the RH LUCE dataset. The error limits are 0.1% for the GUPTA algorithm and 1% in 80% of cases for the slotted scheduler algorithm with rescheduling off. The slotted scheduler outperforms the GUPTA algorithm in terms of both accuracy and lifetime. Accuracy and lifetime summaries are provided in Table 4 for two cases. Figure 21 compares the performance of the multihop subsampling, subseting and slotted scheduling algorithms with the two-tier

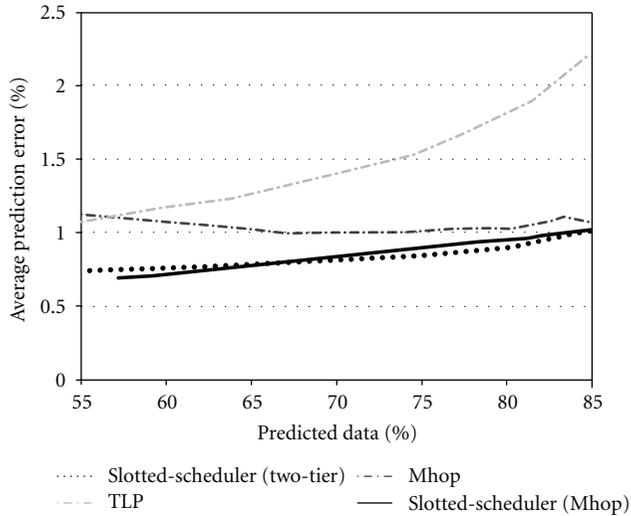


FIGURE 21: Variation of accuracy with percentage of noncollected data.

slotted scheduler. The previous findings are again confirmed. The findings are similar to the two-tier case.

7. Conclusions

Environmental monitoring applications require nodes to continuously transmit data back to the sink. In this paper we have proposed a method which can use the initial collected data to find internode and temporal correlations within the data. It has been shown that the performance of these internode and temporal models varies across time, between data sets and network densities. Herein a novel adaptive scheduling algorithm has been proposed. The algorithm incorporates novel round-robin subset allocation methods for two-tier and multihop networks. When compared to the previously proposed GUPTA algorithm, the two-tier slotted scheduler provides up to 226% longer lifetime and up to 30% greater imputation accuracy. In a multihop network, the slotted scheduling algorithm improves lifetime by up to 553% and can improve accuracy by up to 38% when compared with the GUPTA algorithm. It has been shown that rescheduling can maintain the performance of the system over a long duration of time at the expense of a small increase in cost in terms of the number of transmitted packets. This adds to the importance of using load balancing to lengthen the time it takes for the first node to die so retraining can be done.

Acknowledgment

This research was funded by Enterprise Ireland under Grant CFTD/07/IT/303.

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] V. Shnayder, M. Hempstead, B. R. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 188–200, 2004.
- [3] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies. (IEEE INFOCOM '02)*, vol. 3, pp. 1567–1576, 2002.
- [4] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 95–107, 2004.
- [5] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, 2004.
- [6] F. X. Li, A. Islam, G. C. Perera, and P. K. Kolli, "Real-time urban bridge health monitoring using a fixed wireless mesh network," in *Proceedings of the IEEE Radio and Wireless Symposium (RWW '10)*, pp. 384–387, January 2010.
- [7] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, 2002.
- [8] D. Ingraham, R. Beresford, K. Kaluri, M. Ndo, and K. Srinivasan, "Wireless sensors: oyster habitat monitoring in the bras d'Or lakes," *Distributed Computing in Sensor Systems*, vol. 3560, pp. 399–400, 2005.
- [9] A. Nadamani, P. Basu, and L. Tong, "Extremum tracking in sensor fields with spatio-temporal correlation," in *Proceedings of the IEEE Military Communications Conference (MILCOM '10)*, pp. 1050–1055, 2010.
- [10] R. Brown and V. Swail, "Spatial correlation of marine wind-speed observations," *Atmosphere-Ocean*, vol. 26, pp. 524–540, 1988.
- [11] G. Dubois, M. Saisana, A. Chaloulakou, and N. Spyrellis, "Spatial correlation analysis of nitrogen dioxide concentrations in the area of Milan, Italy," in *Proceedings of the 1st Biennial Meeting of the International Modeling and Software Society*, pp. 176–183, 2002.
- [12] T. Dang, N. Bulusu, and F. W. Rida, "A robust information-driven data compression architecture for irregular wireless sensor networks," *Wireless Sensor Networks*, pp. 133–149, 2007.
- [13] M. Welsh and G. Mainland, "Programming sensor networks using abstract regions," in *Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation*, vol. 1, p. 3, USENIX Association, 2004.
- [14] A. Boulis, S. Ganeriwal, and M. B. Srivastava, "Aggregation in sensor networks: an energy-accuracy trade-off," *Ad Hoc Networks*, vol. 1, no. 2–3, pp. 317–331, 2003.
- [15] M. Cardei, My. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '05)*, vol. 3, pp. 1976–1984, 2005.
- [16] W. Wang, V. Srinivasan, K. C. Chua, and B. Wang, "Energy-efficient coverage for target detection in wireless sensor networks," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 313–322, 2007.
- [17] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in

- Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 32–41, 2002.
- [18] H. Gupta, V. Navda, S. Das, and V. Chowdhary, “Efficient gathering of correlated data in sensor networks,” *ACM Transactions on Sensor Networks*, vol. 4, no. 1, pp. 1–31, 2008.
- [19] S. Yoon and C. Shahabi, “The Clustered AGgregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks,” *ACM Transactions on Sensor Networks*, vol. 3, no. 1, Article ID 1210672, p. 3, 2007.
- [20] A. Jain and E. Y. Chang, “Adaptive sampling for sensor networks,” in *Proceedings of the 1st International Workshop on Data Management for Sensor Networks (DMSN '04)*, pp. 10–16, August 2004.
- [21] P. Tillapart, T. Yeophantong, T. Techachaicherdchoo, T. Thumthawatworn, and U. Udomkul, “Adaptive working schedule modeling for wireless sensor networks,” in *Proceedings of the IEEE Aerospace Conference*, vol. 9, March 2006.
- [22] M. Li, D. Ganesan, and P. Shenoy, “Presto: feedback-driven data management in sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 17, no. 4, pp. 1256–1269, 2009.
- [23] S. B. Roy, G. Das, and S. Das, “Computing best coverage path in the presence of obstacles in a sensor field,” *Lecture Notes in Computer Science*, vol. 4619, pp. 577–588, 2007.
- [24] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong, “Approximate data collection in sensor networks using probabilistic models,” in *Proceedings of the International Conference on Data Engineering*, vol. 2006, p. 48, 2006.
- [25] L. B. Yann-Ael and B. Gianluca, “Round robin cycle for predictions in wireless sensor networks,” in *Proceedings of the 2nd International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '05)*, vol. 2005, pp. 253–258, Melbourne, Australia, 2005.
- [26] J. C. Lim and C. J. Bleakley, “Extending the lifetime of sensor networks using prediction and scheduling,” in *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '08)*, pp. 563–568, December 2008.
- [27] X. Meng, T. Nandagopal, L. Li, and S. Lu, “Contour maps: monitoring and diagnosis in sensor networks,” *Computer Networks*, vol. 50, no. 15, pp. 2820–2838, 2006.
- [28] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong, “Model-driven data acquisition in sensor networks,” in *Proceedings of the 30th International Conference on Very Large Data Bases*, vol. 30, pp. 588–599, VLDB Endowment, 2004.
- [29] D. Apiletti, E. Baralis, and T. Cerquitelli, “Energy-saving models for wireless sensor networks,” *Knowledge and Information Systems*, pp. 1–30, 2010.
- [30] E. Baralis, T. Cerquitelli, and V. D’Elia, “Modeling a sensor network by means of clustering,” in *Proceedings of the 18th International Workshop on Database and Expert Systems Applications (DEXA '07)*, pp. 177–181, September 2007.
- [31] Y. Panthachai and P. Keeratiwintakorn, “An energy model for transmission in Telos-based wireless sensor networks,” in *Proceedings of the International Joint Conference on Computer Science and Software Engineering (JCSSE '07)*, 2007.
- [32] W. Bober and C. Bleakley, “Bailligh: low power cross-layer data gathering protocol for wireless sensor networks,” in *Proceedings of the International Conference on Ultra Modern Telecommunications and Workshops (ICUMT '09)*, pp. 1–7, 2009.
- [33] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [34] “Wireless distributed sensing system for environmental monitoring, Luce deployment,” <http://sensorscope.epfl.ch/index.php/Environmental.Data>.
- [35] G. Lu, B. Krishnamachari, and C. S. Raghavendra, “An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks,” in *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS '04)*, vol. 18, 2004.

Research Article

Network-Coding-Based Cooperative ARQ Medium Access Control Protocol for Wireless Sensor Networks

Angelos Antonopoulos and Christos Verikoukis

Telecommunications Technological Centre of Catalonia (CTTC), Mediterranean Park of Technology (PMT), Building B4, Avenue Carl Friedrich Gauss 7, Castelldefels, 08860 Barcelona, Spain

Correspondence should be addressed to Angelos Antonopoulos, aantonopoulos@cttc.es

Received 20 June 2011; Revised 1 September 2011; Accepted 1 September 2011

Academic Editor: Yuhang Yang

Copyright © 2012 A. Antonopoulos and C. Verikoukis. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We introduce a novel Medium Access Control (MAC) protocol for Automatic Repeat reQuest-based (ARQ-based) cooperative wireless sensor networks. Using network coding techniques, we achieve a better network performance in terms of energy efficiency without compromising the offered Quality of Service (QoS). The proposed solution is compared to other cooperative schemes, while analytical and simulation results are provided to evaluate our protocol.

1. Introduction

Wireless Sensor Networks (WSN) have experienced an impressive growth during the last years. Their inherent restrictions (battery, processing power, etc.) along with their special characteristics (e.g., large network size) constitute great challenges for further research. Moreover, new techniques such as cooperation among nodes [1] and network coding [2] have been introduced to improve the network performance and provide the communication with diversity, robustness, and higher data rates. These new technologies create the need of designing new MAC protocols that exploit the benefits of the aforementioned techniques in order to efficiently use the system resources.

Regarding the cooperative communication, several MAC protocols have been proposed in the literature, classified in two main categories: (i) the cooperative ARQ-based protocols [3, 4] and (ii) the protocols that transform one-hop transmissions to multihop transmissions [5, 6]. However, most of the work is focused on IEEE 802.11 Standard [7], thus not considering the particular traits of WSN.

In the context of IEEE 802.15.4 Standard [8], two MAC schemes have been recently proposed, falling in the first and the second category of cooperative protocols, respectively. In the former, COSMIC [9], the retransmissions are triggered

by the destination after an erroneous packet reception. The relays in the network are enabled to forward the original packets to the destination node, as ARQ defines, using better channel conditions in terms of Packet Error Rate (PER). On the other hand, the latter (WSC-MAC [10]) transforms single one-hop transmissions to multihop transmissions according to the channel conditions. Specifically, when the channel between the relay and the destination is better than the channel between the source and the destination, a two-hop transmission is selected instead of the direct one.

Network coding can be defined as allowing the intermediate nodes in a network not only to forward but also to process the incoming data packets. In the last years, there is a trend towards applying network coding in cooperative communications. In the domain of WSN, Munari et al. [11] have introduced NC-PAN in order to enhance the throughput gain in Time Division Multiple Access (TDMA) high data rate scenarios. However, NC-PAN is not compatible with the non-beacon-enabled mode of IEEE 802.15.4 which adopts Carrier Sense Multiple Access (CSMA).

In this point, let us clarify that IEEE 802.15.4 has two modes of operation: (i) the beacon-enabled mode and (ii) the non-beacon-enabled mode. The former mode (combination of TDMA and CSMA) inherently implies energy saving, since the transceiver of a node is turned off in the sleep mode.

The latter mode uses pure CSMA and supports all types of topologies. Hence, it is of crucial importance to provide the non-beacon-enabled mode with techniques that ensure the efficient use of the energy resources.

In this context, we propose a Network-Coding-based Cooperative ARQ MAC protocol for WSN (NCCARQ-WSN) that coordinates the retransmissions among a set of relay nodes which act as helpers in a bidirectional communication. The novelty of our scheme and the differentiation from the other cooperative protocols of the same category (ARQ-based) lie in the following.

- (1) Network coding techniques are used in order to enhance the system performance.
- (2) Less control packets—and consequently less overhead—are inserted in the network.
- (3) Our protocol operates in CSMA scenarios, hence being compatible with the IEEE 802.15.4 Standard.
- (4) We present an analytical model for the energy consumption in the network from the MAC layer point of view.

Since we have already presented a brief literature review on the related topics, the rest of this paper is organized as follows. In Section 2 we introduce our proposed NCCARQ-WSN MAC protocol along with an operational example. Section 3 presents a detailed mathematical analysis for both QoS metrics and the energy efficiency of our protocol. The validation of the analytical model and the simulation results are provided in Section 4. Finally, Section 5 concludes the paper.

2. Protocol Description

The Network-Coding-based Cooperative Automatic Repeat reQuest MAC protocol for Wireless Sensor Networks (NCCARQ-WSN) has been designed to coordinate the channel access among a set of relays that support bidirectional communications between pairs of sensor nodes. The first goal of NCCARQ-WSN is to enable the IEEE 802.15.4 stations to request cooperation by the neighboring nodes upon an erroneous reception of a data packet. The second design goal of our proposed protocol is to allow the helper nodes to perform network coding techniques to the packets to be transmitted before relaying them.

Two fundamental requirements are needed for the efficient operation of the NCCARQ-WSN protocol: (i) all nodes in the network should operate in a promiscuous mode, that is, they have to be able to listen to all ongoing transmissions and cooperate if requested and (ii) they should store a copy of any received data packet (regardless of its destination address) until the correct reception of this packet is acknowledged by the intended destination. Hence, the relay set is under saturated conditions, since the nodes have always cooperative packets to send in their buffers.

In NCCARQ-WSN, a cooperation phase is initiated once a packet is not received correctly by the destination. Several error detection mechanisms such as Cyclic Redundancy Code (CRC) can be applied in order to perform error control

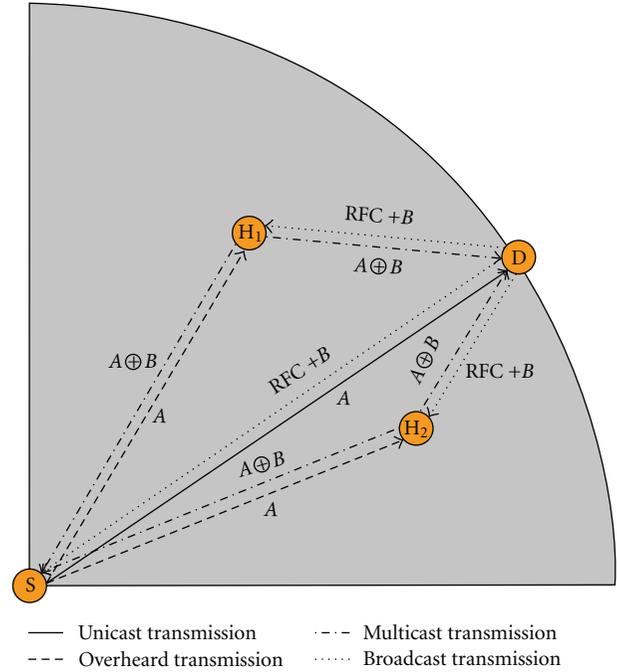


FIGURE 1: NCCARQ-WSN operation.

to the received messages. Therefore, the destination station initiates the cooperation phase by broadcasting a Request for Cooperation (RFC) control packet after a Long Interframe Space (LIFS) period of time, as it is defined in the IEEE 802.15.4 Standard. Furthermore, in the special but not rare case of bidirectional traffic, that is, when the destination station has a data packet for the source station, the packet is broadcasted piggy backed to the RFC message.

The stations that receive the RFC packet are potential candidates to become active relays for the communication process. Therefore, the relay set is formed upon the reception of the RFC and the participants stations get ready to forward their information. Since the partners have already stored the packets that destined both to the destination (so-called cooperative packet) and to the source (so-called piggy-backed packet), they create a new coded packet by combining the two existing data packets, using the XOR method. Accordingly, the active relays will try to get access to the channel in order to transmit the network-coded (NC) packet. A simple scenario that subjects to the initial principles of NCCARQ-WSN is depicted in Figure 1.

In this point, we have to state that NCCARQ-WSN is backwards compatible with IEEE 802.15.4 Standard, as it uses the same frame structure and follows the same principles with the standard. However, some modifications have been made in order for the protocol to efficiently exploit the advantages of both cooperative and network coding techniques.

- (1) Each network-coded packet forwarded by the relays requires an ACK packet to guarantee a reliable communication.

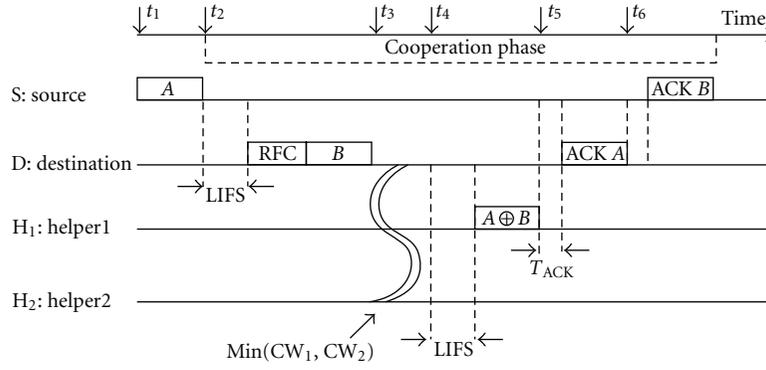


FIGURE 2: NCCARQ-WSN example of frame sequence.

- (2) For bidirectional traffic, data packets can be transmitted along with RFC packets without taking part in the contention phase.
- (3) Since the subnetwork formed by the relay set operates in saturated conditions, it is necessary to execute a backoff mechanism at the beginning of the cooperation phase to minimize the probability of a certain initial collision.

Once the source and the destination receive the NC packet from the relay, they are able to decode it and extract the respective original data packets. Subsequently, they acknowledge the received data packet by transmitting the respective ACK, thus terminating the cooperation phase. Receiving the acknowledgment packet, the relays are informed that the particular communication has been completed, hence becoming able to erase the packets of their buffers. In case that the received coded packets can not be decoded after a certain maximum cooperation timeout due to transmission errors, the relays are obliged to forward again the NC packet.

2.1. Operational Example. In this subsection, we provide an example of operation of NCCARQ-WSN in order to clarify our proposed access protocol. A simple network topology with four stations is considered, all of them in the transmission range of each other. A source station (S) transmits a data packet (A) to a destination station (D) that does also have a packet (B) destined to the source station. Furthermore, there are two helper nodes (H₁ and H₂) that support this particular bidirectional communication. The entire procedure is depicted in Figure 2 and explained as follows.

- (1) At instant t_1 , station S sends the data packet A to station D.
- (2) Upon reception, at instant t_2 , station D fails to demodulate the data packet, thus broadcasting an RFC packet asking for cooperation of the neighboring stations (H₁ and H₂ in this example) along with the data packet B, destined to the station S.
- (3) The reception of the RFC (t_3) triggers the stations H₁ and H₂ to become active relays and set up their back-

off counters (CW_1 and CW_2 , resp.) in order to participate in the contention phase.

- (4) At instant t_4 , the backoff counter of H₁ expires and H₁ transmits the coded packet $A \oplus B$ to the nodes S and D simultaneously.
- (5) At instant t_5 , the station D retrieves the original packet A and sends back an ACK packet since it is able to decode properly the XOR-ed packet.
- (6) At instant t_6 , the node S acknowledges the packet B since it is able to decode properly the coded packet $A \oplus B$.

3. Protocol Analysis

3.1. Delay Analysis. The application of network coding techniques in our proposed scheme implies the simultaneous transmission of more than one packet in the network. Therefore, we analytically estimate the expected time that is needed for two packets to be exchanged under the NCCARQ-WSN protocol. The total time that is elapsed from the initial transmission until the correct reception in the destinations can be defined as

$$\mathbf{E}[T_{\text{total}}] = \mathbf{E}[T_{\text{D}}] + \mathbf{E}[T_{\text{COOP}}], \quad (1)$$

where $\mathbf{E}[T_{\text{D}}]$ represents the average time for a direct transmission of a single data packet from the source to destination and $\mathbf{E}[T_{\text{COOP}}]$ corresponds to the average time required for a cooperative transmission via relays to be completed.

Since $\mathbf{E}[T_{\text{D}}]$ has a value that is easy to be estimated depending on the network's configuration, we focus our analysis on the term $\mathbf{E}[T_{\text{COOP}}]$ in order to derive a closed-form expression for the system's delay. The average time that is spent during the cooperation phase can be defined as

$$\mathbf{E}[T_{\text{COOP}}] = \mathbf{E}[T_{\text{min}}] + \mathbf{E}[T_{\text{CONT}}], \quad (2)$$

where $\mathbf{E}[T_{\text{min}}]$ is the minimum average delay in case of perfect scheduling among the relays, that is, contention-free scheme. On the other hand, the term $\mathbf{E}[T_{\text{CONT}}]$ is used to denote the additional delay that is caused due to the

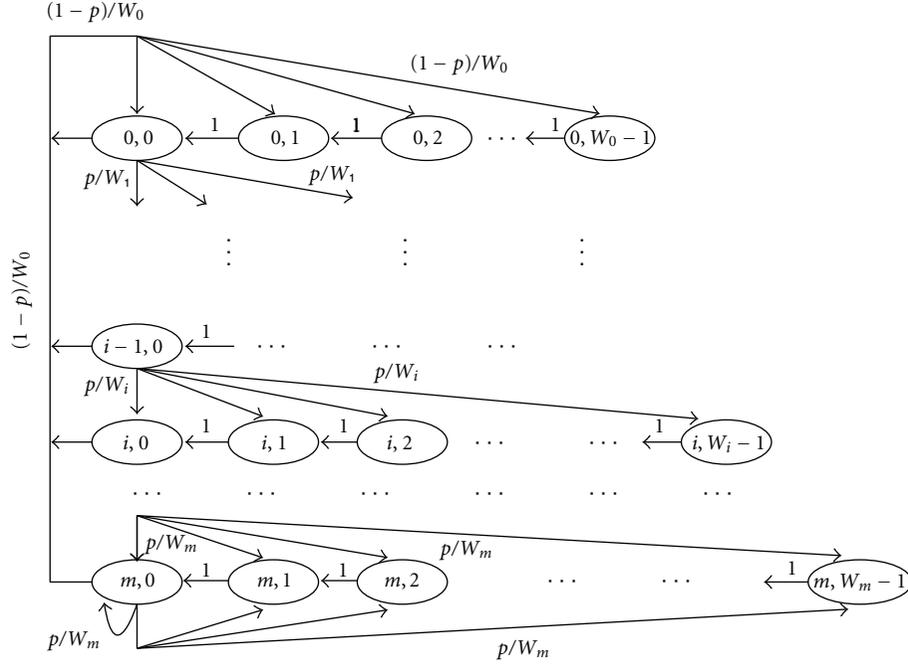


FIGURE 3: The 2-dimensional Markov model state transition diagram.

contention phase which has been adopted in our protocol in order for the probability of collisions to be minimized.

The expected number of retransmissions ($E[r]$) that are required in order to properly demodulate the coded packet at the destination nodes is a factor that affects the time needed for a packet to be delivered. It is directly connected with the packet error rate between the relays and the destination (PER_{R-D}) and could be mathematically expressed as

$$E[r] = \frac{1}{1 - PER_{R-D}}. \quad (3)$$

However, in our scheme two packets are sent at the same time via different channels, and, as a result, the number of retransmissions can be calculated as

$$E[r] = \frac{1 + ((1 - PER_{R-S}) \cdot PER_{R-D} / (1 - PER_{R-D})) + ((1 - PER_{R-D}) \cdot PER_{R-S} / (1 - PER_{R-S}))}{1 - PER_{R-S} \cdot PER_{R-D}}. \quad (4)$$

Therefore, the term $E[T_{\min}]$ can be calculated as

$$\begin{aligned} E[T_{\min}] &= T_{LIFS} + T_{CCA} + T_{RFC} + T_B + T_{ONC} \\ &+ E[r] \cdot (T_{LIFS} + T_{CCA} + T_{A\oplus B} + T_{T_{ACK}}) \\ &+ T_{ACK} + T_{T_{ACK}} + T_{ACK}, \end{aligned} \quad (5)$$

where T_{RFC} and T_{ACK} are the transmission times for the RFC and the ACK packet, respectively. Furthermore, $T_{A\oplus B}$ is the time required to retransmit a coded packet, while T_{ONC} is the time that a relay needs in order to perform network coding techniques. Finally, T_{LIFS} , $T_{T_{ACK}}$, and T_{CCA} are the duration of a LIFS silence period, a T_{ACK} period, and a Clear Channel Assessment (CCA) period, respectively.

Moreover, the term $E[T_{\text{CONT}}]$ can be defined as

$$E[T_{\text{CONT}}] = E[r] \cdot E[T_c], \quad (6)$$

where $E[T_c]$ represents the average time required to transmit a single packet during the contention phase among all

the relays. In order to compute this value we need to model the backoff counter of each of the relays with the Markov Chain presented in [12] (Figure 3), since the formed subnetwork acts as a saturated IEEE 802.15.4 non-beacon-enabled network despite the modifications in the access rules. According to this model, the probability τ that a station transmits in a randomly chosen slot is given by

$$\begin{aligned} \tau &= \sum_{i=1}^m b_{i,0} = \frac{b_{0,0}}{1-p} \\ &= \frac{2(1-2p)}{(1-2p)(W+1) + pW(1-(2p)^m)}, \end{aligned} \quad (7)$$

where

$$b_{0,0} = \frac{2(1-2p)(1-p)}{(1-2p)(W+1) + pW(1-(2p)^m)} \quad (8)$$

and the probability of a collision p as a function of τ is given by

$$p = 1 - (1 - \tau)^{n-1}. \quad (9)$$

In the formulas (7)-(8), $b_{i,k}$ represents the steady-state probability of the state $\{i, k\}$, W is the size of the contention window, m denotes the number of the backoff stages, and n corresponds to the number of the relays in the network.

Furthermore, the probability that at least one relay attempts to transmit can be expressed as

$$p_{tr} = 1 - (1 - \tau)^n \quad (10)$$

and the probability of a successful transmission, that is, one station transmits conditioned on the fact that at least one station transmits is given by

$$p_{s|tr} = \frac{n\tau(1 - \tau)^{n-1}}{1 - (1 - \tau)^n}. \quad (11)$$

Moreover, the probabilities of having an idle (p_i), successful (p_s), or collided (p_c) slot can be written as

$$\begin{aligned} p_i &= 1 - p_{tr}, \\ p_s &= p_{tr} \cdot p_{s|tr}, \\ p_c &= p_{tr}(1 - p_{s|tr}). \end{aligned} \quad (12)$$

Considering the above probabilities, and given that the average number of slots that we have to wait before having a successful transmission can be represented as

$$\mathbf{E}[N] = \sum_{k=0}^{\infty} k(1 - p_s)^k p_s = \frac{1}{p_s} - 1, \quad (13)$$

the total contention time can be written as

$$\mathbf{E}[T_c] = \mathbf{E}[N] \cdot \mathbf{E}[T_{\text{slot}|\text{non_successful_slot}}]. \quad (14)$$

Applying Bayes' theorem we are able to estimate the average duration of a slot, given that the specific slot is either idle or collided:

$$\mathbf{E}[T_{\text{slot}|\text{non_successful_slot}}] = \left(\frac{p_i}{1 - p_s}\right)\sigma + \left(\frac{p_c}{1 - p_s}\right)T_{\text{col}} \quad (15)$$

with σ representing the duration of an idle slot, while T_{col} corresponds to the time of a collision and in our scheme is equal to

$$T_{\text{col}} = T_{\text{LIFS}} + T_{\text{CCA}} + T_{\text{A}\oplus\text{B}} + T_{\text{TACK}}. \quad (16)$$

Therefore, using (13)–(15), the formula (6) can be rewritten as

$$\mathbf{E}[T_{\text{CONT}}] = \mathbf{E}[r] \cdot \left(\frac{1}{p_s} - 1\right) \left[\left(\frac{p_i}{1 - p_s}\right)\sigma + \left(\frac{p_c}{1 - p_s}\right)T_{\text{col}} \right]. \quad (17)$$

Finally, we are able to derive a closed-form formula and compute the total delay for two packets to be exchanged in the system by exploiting (2), (5), and (17).

3.2. Throughput Analysis. The total throughput of the network can be defined as the sum of the throughput that is produced by the successful direct transmissions plus the throughput derived by the cooperation phase after erroneous packet receptions. This can be mathematically expressed as

$$S_{\text{total}} = S_{\text{D}} + S_{\text{COOP}}, \quad (18)$$

where

$$S_{\text{D}} = (1 - \text{PER}_{\text{S-D}}) \cdot \frac{\mathbf{E}[P]}{\mathbf{E}[T_{\text{D}}]}, \quad (19)$$

$$S_{\text{COOP}} = 2 \cdot \text{PER}_{\text{S-D}} \cdot \frac{\mathbf{E}[P]}{\mathbf{E}[T_{\text{total}}]}. \quad (20)$$

In the above expressions, the parameters $\mathbf{E}[T_{\text{D}}]$ and $\mathbf{E}[T_{\text{total}}]$ have been already defined. Furthermore, the packet error rate between the source and the destination is given by $\text{PER}_{\text{S-D}}$, while $\mathbf{E}[P]$ denotes the average packet payload. In this point, it must be clarified that the coefficient 2 in formula (20) is mandatory, since two packets are delivered in each particular transmission.

Thus, having obtained a closed-form expression for $\mathbf{E}[T_{\text{total}}]$ and since $\mathbf{E}[P]$, $\mathbf{E}[T_{\text{D}}]$, and $\text{PER}_{\text{S-D}}$ are known parameters, we are able to compute the theoretical system's throughput.

3.3. Energy Performance Analysis. Having analyzed the operation of the proposed NCCARQ-WSN protocol, we derive a closed-form expression that describes the power consumption in the network:

$$\mathcal{E}_{\text{total}} = \mathcal{E}_{\text{S}} + \mathcal{E}_{\text{COOP}}, \quad (21)$$

where $\mathcal{E}_{\text{COOP}}$ and \mathcal{E}_{S} represent the energy consumption during the cooperative phase and the initial transmission from the source, respectively. The term $\mathcal{E}_{\text{COOP}}$ could be further expressed as

$$\mathcal{E}_{\text{COOP}} = \mathcal{E}_{\text{min}} + \mathcal{E}_{\text{CONT}}, \quad (22)$$

where \mathcal{E}_{min} denotes the energy waste in a perfect scheduled cooperative phase and $\mathcal{E}_{\text{CONT}}$ is the energy that is consumed during the contention phase (i.e., idle and collided slots).

In order to clarify the above equations, we try to compute each term analytically. We consider three different modes:

- (1) *transmission mode*; when the node is transmitting data/control packets,

- (2) *reception mode*; when the node is receiving data/control packets,
- (3) *idle mode*, when the node is sensing the medium without performing any action.

The power levels associated to each mode are P_T , P_R , and P_I , respectively. Furthermore, the relationship between energy and power is given by $\mathcal{E} = P \cdot t$, where the terms \mathcal{E} , P , and t represent the energy, the power, and the time, respectively.

We recall that the network consists of a source, a destination, and a set of n relays. Therefore, considering the network's topology, we have

$$\begin{aligned}
\mathcal{E}_S &= (n+2) \cdot P_I \cdot (T_{LIFS} + T_{CCA}) + P_T \cdot T_A \\
&\quad + (n+1) \cdot P_R \cdot T_A, \\
\mathcal{E}_{\min} &= (n+2) \cdot P_I \cdot (T_{LIFS} + T_{CCA}) + P_T \cdot (T_{RFC} + T_B) \\
&\quad + (n+1) \cdot P_R \cdot (T_{RFC} + T_B) \\
&\quad + (n+2) \cdot P_I \cdot T_{ONC} + \mathbf{E}[r] \\
&\quad \cdot ((n+2) \cdot P_I \cdot (T_{LIFS} + T_{CCA}) \\
&\quad \quad + P_T \cdot T_{A \oplus B} + 2 \cdot P_R \cdot T_{A \oplus B} \\
&\quad \quad + (n-1) \cdot P_I \cdot T_{A \oplus B} \\
&\quad \quad + (n+2) \cdot P_I \cdot T_{T_{ACK}}) \\
&\quad + 2 \cdot P_T \cdot T_{ACK} + (n+2) \cdot P_I \cdot T_{T_{ACK}} \\
&\quad + 2 \cdot (n+1) \cdot P_R \cdot T_{T_{ACK}}.
\end{aligned} \tag{23}$$

The above equation (23) is based on the following principles.

- (i) All stations remain idle during the LIFS, CCA, and T_{ACK} times.
- (ii) The relays that lose the contention phase turn in idle mode.
- (iii) When a station transmits a packet (control or data), the rest of the stations are in promiscuous mode, thus capturing the packets.

Computing the energy consumed during the contention phase constitutes one of the most challenging parts in this analytical model. The total energy wasted during this phase derives from the energy that is spent during both the idle slots and the collisions as well. Let us start by defining that

$$\mathcal{E}_{\text{CONT}} = \mathbf{E}[r] \cdot \mathcal{E}_C, \tag{24}$$

where \mathcal{E}_C represents the average energy required to transmit an NC packet during the contention phase among all the relays. In order to calculate the energy consumed during the collisions, we have to estimate the average number of stations that transmit a packet simultaneously. The probability p_k that exactly k stations are involved in a collision is

$$p_k = \frac{\binom{n}{k} t^k (1-t)^{n-k}}{1 - (1-t)^{n-(k-1)}}. \tag{25}$$

Therefore, the expected number $\mathbf{E}[K]$ of stations that are involved in a collision is

$$\mathbf{E}[K] = \sum_{k=2}^n k \cdot p^k = \sum_{k=2}^n k \cdot \frac{\binom{n}{k} t^k (1-t)^{n-k}}{1 - (1-t)^{n-(k-1)}}. \tag{26}$$

During the idle slots, all the stations in the network remain idle. On the other hand, during the collisions, more than one relay is in transmission mode, two stations (the source and the destination) are in reception mode, while the rest of the relays are in idle mode. Considering the probabilities that we have derived regarding the contention phase (p_c , p_i), the above assumptions can mathematically be expressed as

$$\begin{aligned}
\mathcal{E}_C &= p_i \cdot ((n+2) \cdot P_I \cdot \sigma) \\
&\quad + p_c \cdot (\mathbf{E}[K] \cdot P_T \cdot T_{\text{col}} + (n - \mathbf{E}[K]) \cdot P_I \cdot T_{\text{col}} \\
&\quad \quad + 2 \cdot P_R \cdot T_{\text{col}}),
\end{aligned} \tag{27}$$

where all the parameters have been already defined. Thus, combining (21)–(24) and (27), we are able to estimate the total amount of the energy that is consumed in our protocol.

4. Performance Evaluation

In order to validate our analysis and further evaluate the performance of NCCARQ-WSN we have developed a time-driven C++ simulator that executes the rules of the protocol. Here we present the simulation setup along with the results of our experiments.

4.1. Simulation Scenario. The network under simulation consists of a pair of transmitter-receiver (both nodes transmit and receive data) and a set of relay nodes that facilitate the communication, all of them in the transmission range of each other. In our experiments we consider saturated conditions, that is, the nodes have always packet to send in their buffers. Additionally, the relay nodes are capable of performing network coding techniques to their buffered packets before relaying them. In order to focus on the impact of both network coding and cooperative communication, we have made the following assumptions.

- (1) The traffic is bidirectional, that is, the destination node has always a packet destined back to the source node.
- (2) Original transmissions from source to destination are always received with errors ($\text{PER}_{S-D} = 1$), thus initiating a cooperative phase.
- (3) The channel between the source and the destination is error symmetric, that is, $\text{PER}_{S-D} = \text{PER}_{D-S}$.
- (4) The channel between the source and the relays is error free, that is $\text{PER}_{R-S} = 0$.

The configuration parameters of the network are summarized in Table 1 considering the IEEE 802.15.4 physical layer. The relay set consists of five (5) nodes, each of them

TABLE 1: System parameters.

Parameter	Value	Parameter	Value	Parameter	Value
Data packets	100 bytes	Data rate	256 kb/s	Backoff unit	320 μ sec
MAC header	9 bytes	T_{ACK}	192 μ sec	P_T	15 mW
PHY header	6 bytes	CCA	128 μ sec	P_R	35 mW
ACK, RFC	11 bytes	LIFS	640 μ sec	P_I	712 μ W

implements a backoff counter starting with a contention window $CW_{\min} = 8$. The time for applying network coding (T_{ONC}) to the data packets is considered to be negligible, since the coding takes place between only two packets. Based on hardware specifications and since different power modes are allowed, we have chosen the following power levels for our scenarios: $P_T = 15$ mW (We use the maximum of the transmission power levels that are offered in IEEE 802.15.4.), $P_R = 35$ mW, and $P_I = 712 \mu$ W [13].

In order to evaluate our approach, we compare our scheme with a simple cooperative ARQ scheme (so-called CARQ-WSN), where the bidirectional communication takes place in two steps. In the first step, the source sends a packet to the destination and, upon the erroneous reception, the destination broadcasts the RFC packet, thus triggering the relays to retransmit the packet. In the second step, the destination transmits its own packet to the source and the same procedure as in the first step is repeated, thus consuming valuable network resources. In both steps, the relays take part in the contention phase in order to access the medium and transmit their packets.

The delay and the throughput, as they have been defined in Sections 3.1 and 3.2, respectively, are the metrics that we use in order to evaluate the QoS performance of our protocol. Moreover, in order to evaluate the energy performance of our proposed protocol we use the energy efficiency metric [14]. It is denoted by η and defined as:

$$\eta = \frac{\text{total amount of useful data delivered (bits)}}{\text{total energy consumed (Joule)}}. \quad (28)$$

Before proceeding to the simulation results, it is worth mentioning that the definition in (28) inherently implies that network coding benefits the energy efficiency of a protocol, as the number of the delivered bits increases by combining multiple data packets.

4.2. Simulation Results. Figure 4 shows that the numerical and simulation results are almost perfectly matched, thus verifying our analysis. Comparing with simple cooperative schemes which have the advantage of spatial diversity through relays without any network coding capability, we can achieve an enhancement in the network's throughput up to 80%. We can see that the throughput in NCCARQ-WSN for one retransmission (the minimum number when the initial transmission contains errors) is 99 kb/s while in simple cooperative schemes, the throughput is approximately 66 kb/s. Furthermore, upon the increase in the number of required retransmissions (x -axe), the throughput gain increases as well. This significant improvement makes sense since the

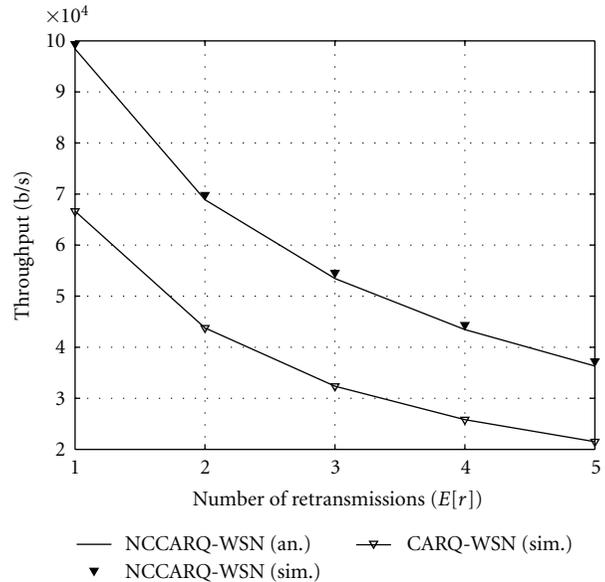


FIGURE 4: System throughput (NCCARQ-WSN versus CARQ-WSN).

number of total transmissions in NCCARQ-WSN protocol is lower compared to CARQ-WSN, as the packets are sent coded, while the number of RFC packets is also decreased. Moreover, in NCCARQ-WSN the cooperation phase is initiated only once when the traffic is bidirectional, thus saving time compared to other cooperative schemes where the cooperation takes place upon every erroneous packet reception.

Figure 5 presents the packet delay in both Network-Coding-based and simple Cooperative ARQ MAC protocols. In this point, we must recall that two packets are delivered to their respective destinations in each transmission cycle of NCCARQ-WSN. Hence, in order to be accurate, we compare the delay in NCCARQ-WSN with the time required for two packets to be exchanged in CARQ-WSN.

As it can be observed, we can achieve significantly lower packet delay by using network coding techniques. Specifically, the average time that is required for two packets to be transmitted using CARQ-WSN is 0.024 sec in channels where one retransmission is necessary, reaching up to 0.074 sec when five retransmissions are required. On the other hand, the delay values in NCCARQ-WSN are 0.016 and 0.042 sec for one and five retransmissions, respectively. This difference can be rationally explained considering the operation of our proposed NCCARQ-WSN scheme, where some data

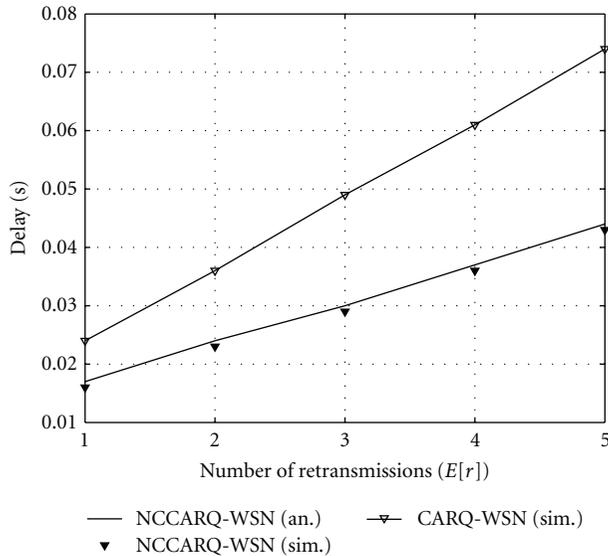


FIGURE 5: Packet delay (NCCARQ-WSN versus CARQ-WSN).

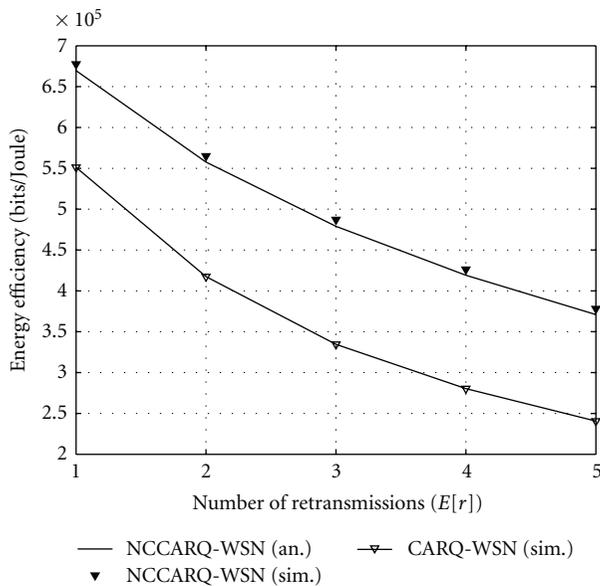


FIGURE 6: Energy efficiency (NCCARQ-WSN versus CARQ-WSN).

packets are sent to the relay (attached to the RFC message), thus avoiding the erroneous channel. Furthermore, in our proposed scheme we manage to reduce the backoff phases by sending two packets simultaneously, while in simple cooperative protocols the relays have to participate in the contention phase for each packet that has to be retransmitted. Therefore, we are able to enhance the packet delay, since the time that is spent in idle slots and collisions is significantly reduced, especially as the number of required retransmissions grows.

Figure 6 shows that our analysis verifies the simulation results with regard to the energy performance. Comparing our proposed network-coding-based scheme with simple cooperative protocols for different number of retransmissions (and consequently different PER between the relays and the

destinations), we observe that our scheme is more energy efficient than non-network-coding-based schemes, since more bits are delivered over the same amount of energy consumed. Keeping constant the data packet length (100 bytes) the energy efficiency of NCCARQ-WSN decreases as the number of relay retransmissions grows. However, the difference with simple cooperative schemes remains steadily over 30%.

5. Conclusion

In this paper, a novel network-coding-based MAC protocol for cooperative wireless sensor networks has been presented. Compared to simple cooperative ARQ-based MAC protocols, the proposed solution has been proven to be up to 50% more energy efficient, while the provided quality of service in terms of throughput and delay is not compromised. In order to optimize the energy management in the network, MAC schemes have to be combined with energy-aware routing solutions, while sleep modes techniques for inactive nodes should be considered as an extra option. Our future research will be focused on such issues.

Acknowledgments

This work has been funded by the Research Projects VITRO(257245), WSN4QoL(286047), GREENET(PITN-GA-2010-264759), and CO2GREEN(TEC2010-20823).

References

- [1] T. M. Cover and A. A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 572–584, 1979.
- [2] 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.
- [3] K. Lu, S. Fu, and Y. Qian, "Increasing the throughput of wireless LANs via cooperative retransmission," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference, (GLOBECOM '07)*, pp. 5231–5235, November 2007.
- [4] J. Alonso-Zárate, E. Kartsakli, C. Verikoukis, and L. Alonso, "Persistent RSCMA: a MAC protocol for a distributed cooperative ARQ scheme in wireless networks," *Eurasip Journal on Advances in Signal Processing*, vol. 2008, Article ID 817401, 2008.
- [5] T. Guo and R. Carrasco, "CRBAR: cooperative relay-based auto rate MAC for multirate wireless networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, Article ID 5351713, pp. 5938–5947, 2009.
- [6] X. J. Zhu and G. S. Kuo, "Cooperative MAC scheme for multi-hop multi-channel wireless mesh networks," in *Proceedings of the 68th Semi-Annual IEEE Vehicular Technology, (VTC '08)*, September 2008.
- [7] Wireless Medium Access Control and Physical Layer WG, IEEE Draft Standard P802.11, "Wireless LAN," IEEE Std. Department D3, January 1996.
- [8] IEEE Standard for Information Technology and Telecommunications and Information Exchange between Systems, "Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal

- Area Networks (LR-WPANs), IEEE Std 802.15.4,” pp.1-670, 2003.
- [9] A. B. Nacef, S. Senouci, Y. Ghamri-Doudane, and A.-L. Beylot, “COSMIC: a cooperative MAC protocol for WSN with minimal control messages,” in *Proceedings of the 4th IFIP International Conference on New Technologies, Mobility and Security, (NTMS '11)*, pp. 1–5, February 2011.
 - [10] B. Mainaud, V. Gauthier, and H. Afifi, “Cooperative communication for wireless sensors network : a mac protocol solution,” in *Proceedings of the 1st IFIP Wireless Days, (WD '08)*, pp. 1–5, November 2008.
 - [11] A. Munari, F. Rossetto, and M. Zorzi, “Hybrid cooperative-network coding medium access control for high-rate wireless personal area networks,” in *Proceedings of the IEEE International Conference on Communications, (ICC '10)*, pp. 1–6, May 2010.
 - [12] Y. Fujii, D. Umehara, S. I. Denno, M. Morikura, and T. Sugiyama, “Saturation throughput analysis of unslotted CSMA-CA networks,” in *Proceedings of the 25th International Technical Conference on Circuits/Systems, Computers and Communications*, pp. 688–691, Pattaya, Thailand, July 2010.
 - [13] B. Otal, *Optimization of wireless ambient and body sensor networks for medical applications*, Ph.D. Dissertation, Barcelona, Barcelona, Brazil, March 2010.
 - [14] M. Zorzi and R. R. Rao, “Energy constrained error control for wireless channels,” in *Proceedings of the Global Telecommunications Conference (GLOBECOM '96)*, vol. 2, pp. 1411–1416, London, UK, November 1996.

Research Article

An Efficient Data-Gathering Scheme for Heterogeneous Sensor Networks via Mobile Sinks

Po-Liang Lin and Ren-Song Ko

*Department of Computer Science and Information Engineering, National Chung Cheng University,
168 University Road, Min-Hsiung Chia-Yi 621, Taiwan*

Correspondence should be addressed to Ren-Song Ko, korensen@cs.ccu.edu.tw

Received 16 June 2011; Revised 27 August 2011; Accepted 27 August 2011

Academic Editor: Yuhang Yang

Copyright © 2012 P.-L. Lin and R.-S. Ko. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Typical Wireless Sensor Networks (WSNs) use static sinks to collect data from all sensor nodes via multihop forwarding. This results in the hot spot problem since the nodes close to the sink have a tendency to consume more energy in relaying data from other nodes. Many approaches using mobile sinks have been proposed to prevent this problem, but these approaches still suffer from the buffer overflow problem due to the limited memory capacity of the sensor nodes. This paper proposes an approach in which the mobile sink traverses a subset of nodes. Given the characteristics of wireless communication, such an approach can effectively alleviate the buffer overflow problem without incurring additional energy consumption. To further alleviate the buffer overflow problem, we propose the *Allotment Mechanism* which allows nodes with different data sampling rates to share their memory and, thus, extend the overflow deadline. Finally, the effectiveness of the proposed approach is verified via the GloMoSim network simulator. The results show that our approach incurs fewer buffer overflows than other data-gathering schemes.

1. Introduction

Wireless Sensor Networks (WSNs) play an important role in a wide range of applications [1, 2], such as routine data collection [3, 4], distant unfriendly location exploration [5, 6], emergency response [7, 8], and hazard detection [9]. A WSN may consist of hundreds or thousands of sensor nodes that integrate sensors with limited onboard processing and wireless communication capabilities. With the available technology, the sensors are usually battery powered and have limited memory. It is not economically or technically feasible to recharge or replenish their energy. In addition, hardware constraints limit the amount of sensed data that can be stored in a sensor node. Data may, therefore, need to be discarded if it cannot be processed before the buffer overflows [10]. Therefore, buffer overflow and network lifetime are often seen as the two most important issues in designing a WSN.

Typical WSNs have static data sinks for data gathering, in which sensor nodes transmit data to sinks via multihop forwarding [11–13]. In other words, a node not only transmits the information sensed by itself but also relays

the data packets generated by others. Therefore, nodes near sinks have a tendency to consume more energy because they may need to relay data for nodes at a distance from sinks. Multihop forwarding may create a hot spot problem which may exhaust the nodes near sinks, leaving the sinks isolated from the rest of the network, and the networks will cease to function since the data sensed by the remote nodes cannot be forwarded to the sinks, thus, limiting the utility of the remaining working nodes.

Many studies have proposed the use of mobile sinks for sensor networks [14, 15] to solve the hot spot problem and improve energy efficiency in data gathering. Instead of waiting for data, a mobile sink (MS) can travel around the network, approach individual nodes, and collect data from them. For example, in [16], an MS can randomly roam around the region of interest (ROI) and collect data from sensor nodes. The authors of [17, 18] considered the mobility planning problem which involves determining a good traverse path or rendezvous point for the MS to optimize the energy consumption used in communicating with each sensor node. Note that, intuitively, the MS can approach

each node to avoid the need for multihop forwarding, and thus minimize communication energy consumption. The limited memory size of sensor nodes means a node can hold or buffer data before memory is full, and a node may be unable to hold its generated data in memory for collection by the MS if the MS traverses each node inappropriately. As a consequence, such a buffer overflow problem may result in information loss. Determining a traverse path within a given deadline is known as the *Traveling Salesman Problem* (TSP) and its complexity is NP complete. Rather than searching for a moving path without any buffer overflow, Somasundara et al. [32] proposed a mobile element scheduling method to collect data with dynamic deadlines with the goal of minimizing the number of missed deadlines. They used the moving cost of the MS and the buffer overflow time of the sensor nodes to determine the visiting sequence of sensor nodes. Nevertheless, this scheduling method requires the MS to visit all sensor nodes. A large number of nodes spread over the ROI would result in a long moving path, reducing MS efficiency, and some nodes may still suffer the buffer overflow problem. Furthermore, once a buffer overflow occurs, the data is simply dropped. Salvaging the data is not considered.

This paper tries to address the energy consumption problem of data gathering and avoid the buffer overflow problem by using a hybrid approach that combines one-hop data forwarding and MS. Note that traditional static sink approach has the hot spot problem mentioned above, but MS has the minimal buffer overflow problem. On the other hand, using MS to traverse each node alleviates the hot spot problem at the cost of incurring buffer overflow. By combining data forwarding and MS, we can designate several rendezvous points for the MS to visit. Once the MS reaches a rendezvous point, the sensor nodes close to the rendezvous point will send their buffered data to the MS via data forwarding. Increasing the number of rendezvous points decreases the severity of the hot spot problem but increases the severity of the buffer overflow problem. That is the number of hops for data forwarding and the number of rendezvous points are a tradeoff between the hot spot problem and the buffer overflow problem and, thus, a key challenge to data gathering. In this paper, the maximal number of hops to the closest rendezvous point is restricted to one, and our proposed approach is then reduced to determine an appropriate path to traverse these rendezvous points within the data overflow deadlines. The WSN application considered here is environmental monitoring or surveillance with heterogeneous sensor nodes, in which the nodes may have different hardware, sensing capability, or data sampling rates. The proposed approach requires an MS, a robot, or a vehicle equipped with a transceiver to collect the data from the nodes that are within its one-hop communication range. Besides, a node will buffer the generated data in its memory if there is no MS within its one-hop communication range. The objective is to have the MS to collect the data from nodes before their buffers are full. We achieve this objective by minimizing the rendezvous points and determining a good traverse path.

Our proposed approach for efficient data gathering can be summarized as follows.

- (1) We cluster the sensor nodes by constructing a dominating set [20, 21] and assign the dominating nodes as rendezvous points. By definition, a node is either a dominating node or within a one-hop communication range of its closest dominating node. After that, the MS only needs to traverse the dominating nodes for data gathering. There is no need for dominating nodes to buffer the data of its one-hop neighbors. When MS arrives at a dominating node, it will collect the data from the dominating node and its one-hop neighbors via one-hop data transmission.
- (2) We also propose an *Allotment Mechanism* to alleviate the buffer overflow problem, specifically for sensor nodes with high sampling rates. The *Allotment Mechanism* will average the amount of sensed data in the cluster. Nodes with higher sampling rates may temporarily buffer their data in the memory of the node with a lower sampling rate in the same cluster. Doing so can extend the time to buffer overflow for the nodes with the higher sampling rates, and the overall buffer overflow problem is alleviated.

After determining rendezvous points, we consider two algorithms for the rendezvous point traversing problem with deadline constraints, namely the *Dominating-Based Minimum Weighted Sum First* (DMWSF) algorithm and the *Dominating-Based Traveling Salesman Approximation* (DTSP) algorithm. The objective is to minimize the number of missed deadlines for a given data sampling rate, or the dual problem, to maximize the data sampling rates without any missed deadlines.

To verify the effectiveness of the proposed approach, we conducted simulations with the GloMoSim simulator [22] and compared the results with those from other related algorithms. The results illustrate that the proposed approach significantly outperforms other algorithms in terms of providing longer network lifetime and less buffer overflow.

The rest of this paper is organized as follows. Section 2 discusses the related work with regard to data-gathering schemes via MS. The sensor network model, assumptions, and notations are introduced in Section 3, and the proposed algorithms are described in Section 4. Section 5 describes and discusses the simulation environment, simulation parameters, and simulation results. Conclusions are given in the last section.

2. Related Work

As mentioned earlier, MS data-gathering schemes may avoid the hot spot problem and prolong the network lifetime. Many approaches have been proposed for WSN [19, 23, 24], and one major design challenge is to plan a traverse path without missing any deadlines. One intuitive solution is to traverse each sensor node in the shortest distance so that the network can tolerate buffer overflow problems with a high sampling rate [25]. Wohlers et al. [26] pointed out that whether a data-gathering approach is energy efficient depends on the application characteristics, including the

mobility patterns of sinks and the desired freshness of the collected data.

Several researchers [27, 28] have proposed using either a tree, cluster, or grid structure to level down the large-scale sensor network. In [29] an energy-efficient routing protocol Multitier Grid Routing Protocol (MGRP) is proposed which introduces a special hybrid multi-tier structure for data dissemination. They form an optimized cluster which transmits reliable data to its higher tier cluster head, with the uppermost cluster head from neighbor grids further negotiating to construct the data d-tree from which the mobile sink can access and send queries. After reducing the solution space, some proposed methods construct the optimal traverse path for mobile sink. In [30] this problem is addressed by minimizing the traverse length of the mobile elements (MEs), seeking to obtain optimal scheduling for the MEs based on the minimal stop point set to minimize their traverse distance.

In [16, 31], Shah et al. proposed a data-gathering scheme using a mobile sink called Mobile Ubiquitous LAN Extensions (MULEs) which is basically a moving observer, such as an animal or human being, carrying a transceiver. The mobile observer wanders in the ROI, collecting and transmitting information to access points for further processing and analysis. For a performance comparison with our proposed approach, this approach is also implemented in the GloMoSim simulator and denoted as Random_Waypoint.

As mentioned above, one possible traverse path for alleviating the buffer overflow problem is the one with the shortest distance, a TSP with NP-complete complexity. Therefore, instead of permuting the node visiting order to find the minimal traverse path, Somasundara et al. [32] proposed an approach called k -lookahead in which only k nodes (k is less than 10) are permuted at a time and the first k node visited is the one with the minimum cost among k nodes. Furthermore, they also proposed a mobile element scheduling method to collect data in [32]. The MS calculates a weighted value and uses the Minimum Weighted Sum Value First (MWSF) algorithm to determine the next destination. The weighted value consists of two factors, cost and deadline, in which the cost is the distance between the MS and the potential destination node and the deadline is the time to buffer overflow for the potential destination node. If the weighted value is dominated by the cost, the MS will traverse the nodes by the shortest distance; if dominated by the deadline, the MS will visit the node with the earliest deadline first. This algorithm is also called the Early Deadline First (EDF) algorithm in [32]. The performance results in [32] show that the buffer overflow ratio of MWSF is similar to k -lookahead for $k = 6$, while MWSF is computationally inexpensive.

Ma and Yang [17] introduced an energy-efficient data-gathering scheme with an MS called SenCar. The fundamental idea is to determine an energy-efficient traverse path via a bisection method. Given starting and ending points, the MS can move straight across the ROI to gather the data from sensor nodes via multihop forwarding. While a straight line may not be an energy-efficient moving path, it is possible to find a turning point in the middle. As a consequence, the

moving path moves from the starting point to the turning point and then to the ending point, and the MS will gather the data along the path using less energy. Furthermore, more new turning points can be recursively added between the starting point, turning points, and the ending point for better energy efficiency. Note that, though more turning points may reduce the number of hops for data forwarding, which in turn reduces energy consumption, it will increase the traverse distance and the MS will need more time to cross the ROI, which may produce the buffer overflow problem. We also include this approach, denoted as BISECTOR, in our simulation for performance comparison.

Xing et al. [18] proposed a rendezvous-based data-gathering scheme. A subset of nodes is designated as rendezvous points that will buffer and aggregate data originating from other sensor nodes. The MS will traverse each rendezvous point to collect these data. Conceptually, these rendezvous points serve as temporary static sinks from which the MS collects data; thus, this approach basically distributes the hot spot problem over these rendezvous points. Rao and Biswas [33] introduced a distributed ant-based TSP mechanism to determine a traverse path for MS such that all the chosen rendezvous point are visited.

Saad et al. [34] proposed a hierarchical structure for large-scale sensor networks via a clustering algorithm. Cluster heads are randomly selected in time-driven scenarios, and the MS will traverse these cluster heads to minimize the energy consumption on multihop forwarding. However, this approach does not consider the buffer overflow problem.

This paper considers heterogeneous WSNs, in which sensor nodes may have different data sampling rates and, thus, different levels of severity of the buffer overflow problem. To verify the effectiveness of our proposed approach, several other approaches, namely, Random_Waypoint, BISECTOR and MWSF, are implemented in the simulation for performance comparison for factors including network lifetime and number of buffer overflows.

3. Network Model and Assumptions

In this paper, a heterogeneous WSN with different data sampling rates can be modeled as follows.

- (1) A connected graph consists of n nodes with unique ID $1, \dots, n$ and an MS that moves around the ROI.
- (2) Sensing operations of sensor nodes, such as temperature and humidity, are determined prior to deployment and set with given sampling rates (which may be different). The vector $S[1, \dots, n]$ denotes the sampling rates (i.e., bytes per unit of time) of the sensor nodes.
- (3) Each sensor node has a limited memory size (which may be different) represented by the vector $M[1, \dots, n]$.
- (4) All sensor nodes and the MS have the same communication range, denoted as R_C .
- (5) The MS is aware of the position of each sensor node.

TABLE 1: The notation list.

Description	Notation
Overflow time	T_o
Maximum overflow time	$T_{o,max}$
Sample rate	S
Memory size	M
Election timer	T_{elk}
Maximum election timer	$T_{elk,max}$
Timer to overflow time ratio	η
Overflow time with <i>Allotment Mechanism</i>	D_o
Allotment data	AD

- (6) The MS is initially stopped, and all nodes know how to send data to the MS (In this case, the MS acts like a static sink.) In addition, once the MS begins moving, it does so with a constant speed, v m per unit of time.

In addition to the WSN model above, the following assumptions are made.

- (1) Any two nodes can directly communicate via bi-directional wireless links if their Euclidean distance is not greater than R_C .
- (2) All nodes have their clocks synchronized.
- (3) The actual data transfer time from a node to the MS is negligible when calculating the buffer overflow time.

Based on the model and assumptions above, it is easy for the MS to derive the following information.

- (1) Since the MS knows the position of each node, it can derive the distance for each pair of nodes, denoted as D_{ij} for node i and j . Furthermore, the matrix $\text{cost}[1, \dots, n][1, \dots, n]$ that denotes the time needed for the MS to move from one node to another is defined as

$$\text{cost}[i][j] = \frac{D_{ij}}{v}. \quad (1)$$

- (2) The buffer overflow time or the time to fill the memory of node i , denoted as $T_o[i]$, is defined as

$$T_o[i] = \frac{M[i]}{S[i]}. \quad (2)$$

Table 1 lists the notations used in this paper.

Given the model and assumptions above, this paper addresses two research issues:

- (1) reducing the number of hops from sensor nodes to the MS, which in turn will reduce the energy consumption of sensor nodes and
- (2) determining a traverse path for MS to alleviate the buffer overflow problem, that is, minimizing the number of missed buffer deadlines.

The proposed approaches are described in detail in the next section.

4. Algorithms

4.1. Overview. One intuitive way to reduce communication energy consumption is to have the MS approach in each node to collect data. However, due to the NP completeness of TSP, it is infeasible to both traverse each node and meet the buffer overflow deadline, particularly in large-scale networks. Note that, in WSN, sensor nodes communicate with each other via wireless communication. As long as the MS is in the communication range of a sensor node, it can collect the sensor node's data. Therefore, instead of visiting each node, we designate some sensor nodes as rendezvous points so that each node is located within a one-hop communication range of at least one of these rendezvous points. By only traversing these rendezvous points, the MS can collect all data from the sensor nodes via wireless communication without expending additional energy on communication. One possible set of such rendezvous points is the dominating set of WSN in which, by definition, each node is either a dominating node or within a one-hop range of a dominating node [35, 36]. With fewer nodes to visit, it is easier to plan the MS's traverse path and meet the buffer overflow deadlines.

Moreover, sensor nodes may have different sampling rates and, thus, different buffer overflow deadlines. To further extend the deadlines of nodes with higher sampling rates and, thus, alleviate the buffer overflow problem, we allow the nodes with higher sampling rates to temporarily buffer their data in the memory of their one-hop neighbors that have lower sampling rates. Basically, our proposed approach consists of three modules.

- (1) The first module reduces the scale of the WSN so that the MS may collect all data without traversing all nodes and without sensor nodes needing to relay data from other nodes. This is achieved by letting each sensor node run the *Time Delay-Based Dominating (TDD) algorithm* and use its own buffer overflow time to select dominating nodes.
- (2) Sometimes analyzing variations in the ROI may require maintaining the completeness of data collected by sensor nodes. Therefore, the *Allotment Mechanism* is proposed here to alleviate the buffer overflow problem by letting nodes temporarily buffer their data in the memory of their one-hop neighbors.
- (3) The last module determines the MS's traverse path so that missed buffer overflow deadlines can be avoided or minimized. Note that, though the problem scale is reduced to the dominating set, the complexity is still NP complete. Therefore, instead of finding the optimal path, this paper considers a heuristic algorithm and an approximation algorithm, namely the *Dominating-Based Minimum Weighted Sum First (DMWSF) algorithm* and the *Dominating-Based Traveling Salesman Approximation (DTSP) algorithm* respectively.

Each module will be described in detail in the following subsections.

4.2. Time Delay-Based Dominating (TDD) Algorithm. Some rendezvous-based data-gathering schemes, such as [18, 37], utilize rendezvous points as static sinks. The data collected by other nodes will be forwarded to these rendezvous points via multihop forwarding for the MS to collect. Thus, the hot spot problem still exists and is merely transferred from a centralized data sink to these rendezvous points. For example, some rendezvous-based schemes ask the rendezvous points to buffer data originated from other nodes and then forward the buffered data to the MS when it arrives. Such an approach raises both the hot spot problem and the buffer overflow problem due to the limited memory size of the rendezvous points. Note that the main cause of the hot spot problem is the multihop forwarding in which the nodes closest to the sinks are more likely to relay data from other nodes and, thus, consume more energy for communication. Thus, we consider using the dominating set as the rendezvous points, not only to decrease the scale of TSP but also to eliminate the hot spot problem.

Definition 1. A dominating set of a graph $G = (V, E)$ is a subset $S \subseteq V$ of the nodes such that, for all nodes $u_1 \in V$, either $u_1 \in S$ or a neighbor u_2 of u_1 is in S . Here, if the distance between u_2 and u_1 is less than R_C , u_2 and u_1 are neighbors. Besides,

- (i) $u_1 \in V$ is the dominating header (DH) if $u_1 \in S$;
- (ii) $u_1 \in V$ is the dominating member (DM) if u_1 is a neighbor of $u_2 \in S$;
- (iii) a DH and its DM form a dominating cluster (DC).

Referring to Algorithm 1, we propose a time-based algorithm TDD to determine DH, DM, and DC based on sensor node's timers, T_{elk} . The main purpose of this algorithm is to reduce information exchange and to construct DC in a fixed amount of time denoted as $T_{\text{elk_max}}$. To integrate this with *Allotment Mechanism*, described in the next subsection, we use buffer overflow time to define timers in dominating header selection; that is,

$$T_{\text{elk}}[i] = \eta \times T_o[i]. \quad (3)$$

To guarantee that TDD will terminate in $T_{\text{elk_max}}$, η can be determined by

$$\eta = \frac{T_{\text{elk_max}}}{T_{o_max}}. \quad (4)$$

All the notations mentioned here are listed in Table 1. Note that the parameters η and T_{o_max} are prerequisites for TDD and need to be given to each sensor node in advance. After deploying the sensor nodes in the ROI, each sensor node timer starts to countdown. If a sensor node does not receive any DH declaration message from its neighbors before its election timer expires, it will declare itself to be DH and broadcast a DH declaration message to its neighbors. The DH declaration message contains the buffer overflow time of the DH. Thus, if a sensor receives more than one DH declaration, it may select the DH with the least buffer overflow time or resort to some tie-breaking mechanism if

```

(1) set timer =  $T_{\text{elk}}[i]$ 
(2) while timer  $\neq 0$  do
(3)   timer --
(4) end while
(5) if no DH declaration message from its neighbors then
(6)   declare as DH and broadcast a DH declaration
      message to its neighbors
(7) else
(8)   declare as DM and reply to its DH neighbor with a
      DM declaration message
(9) end if

```

ALGORITHM 1: Time delay-based dominating algorithm.

the buffer overflow times are same. On the other hand, if a sensor node receives a DH message from its neighbors, it will declare itself to be DM and reply to its DH neighbor with a DM declaration message. The TDD algorithm is presented in Algorithm 1.

Note that the MS is initially stopped and all nodes know how to send data to the MS. After the DHs are selected, they may notify the MS with all their DMs. In addition, the MS has knowledge about all nodes and, thus, knows the TDD results of all sensors in maximum election time, $T_{\text{elk_max}}$. The MS will then start to traverse DMs to collect data. To cope with packet loss due to unreliable communication conditions, it is possible to assign the MS a timer with the expiration time greater than $T_{\text{elk_max}}$. Once expired, the MS begins to traverse the DHs it knows regardless of information incompleteness. Besides, the MS will visit the nodes for which TDD results are unknown to the MS to collect their TDD results. In other words, it is still possible for the MS to know all the DHs.

From the definition of the election timer, a DH obtained from TDD will have a shorter buffer overflow time than its neighbors; that is, the DH has the critical buffer overflow time of its DC. Thus, the buffer overflow time for a DC can be represented by the buffer overflow time of its DH, and the MS only needs to consult each DH's buffer overflow time to plan a traverse path which minimizes the number of missed buffer deadlines. As long as the MS moves to each DH before its buffer overflow deadline, it can collect the data from the DH and all its DMs without any data loss.

Note that if a DH's DH declaration message is delayed or lost during the setup of the dominating clusters, its neighbors may declare themselves to be DHs after election timers expire, thus, increasing the number of clusters and the complexity of the traverse path planning. However, this problem may be alleviated by pruning the dominating clusters, for example, by modifying the approaches proposed in [35] in which we may use buffer overflow time instead of node ID to determine which cluster will be pruned. On the other hand, when a DM declaration message is delayed or lost, a DH may not recognize its neighbor as a DM, which may cause the *Allotment Mechanism* described later fail since it does not have the complete information of its neighbors' data-sampling rates. This problem may be recovered when

the MS visits the DH; all its DMs have a chance to update their membership while sending data to the MS.

4.3. Allotment Mechanism. Note that the buffer overflow deadline of a DC is determined by the deadline of the node with the highest sampling rate, that is, the DH. To further extend the DH's deadline and, thus, alleviate the buffer overflow problem, we introduce the *Allotment Mechanism* which allows a DH to temporarily buffer its data in the memory of its DMs. The basic idea is to select the first k highest sampling rate nodes to share their memory with the DH.

For the purposes of discussion, we sort m member nodes of a given DC in descending order of data sampling rates and relabel them as u_0, u_1, \dots, u_m ; that is, the sampling rate of u_i is not lower than that of u_{i+1} . With the *Allotment Mechanism*, the DH, that is, u_0 , may distribute its data to u_1, u_2, \dots, u_k , so its buffer will not be filled so quickly, thus, extending its buffer overflow time, along with the DC.

Note that the deadline of a DC is determined by the deadline of its DH, for example, node u_0 , which is $T_o[u_0]$ defined in (2). With the *Allotment Mechanism*, the buffer overflow time of the DH or DC is no longer $T_o[u_0]$ and is extended to $D_o[u_0]$, defined as

$$D_o[u_0] = \frac{\sum_{j=0}^k M[u_j]}{\sum_{j=0}^k S[u_j]}. \quad (5)$$

Referring to Table 1, M is the memory size of the sensor nodes. Note that (5) is derived from the fact that the total memory of the DH, and k DMs is shared, and, thus, the overflow deadline is the total memory size divided by total sampling rates of DH and k DMs.

Consider the example depicted in Figure 1. u_0 is DH and has three DMs, u_1, u_2 , and u_3 , in the same DC. The order of the DMs' sampling rates is $u_0 > u_1 > u_2 > u_3$. If we share the memory of 2 DMs in the *Allotment Mechanism*, the memory of node u_1 and u_2 will be shared. Note that u_0 has the highest sampling rate in the DC, followed by u_1 and u_2 . If the memory of u_0 is almost filled, it may free some of its memory space by temporarily storing some of its data into u_1 's or u_2 's memory. Thus, the space released in u_0 can be used to store more data and extend the buffer overflow deadline.

A DH can easily determine the order of its DMs' sampling rates, since each DM will return a DH with the DM declaration message once the election timer expires. The election timer is proportional to the buffer overflow time, and, thus, the order of the DM declaration message mirrors the order of the sampling rates. Therefore, the first k DMs sending the DM declaration messages will share memory with the DH.

Basically, u_0 calculates the quantity of allotment data of the k members, u_1, u_2, \dots, u_k , by the following:

$$AD[u_j] = (T_o[u_j] - D_o[u_0]) \times S[u_j], \quad 1 \leq j \leq k. \quad (6)$$

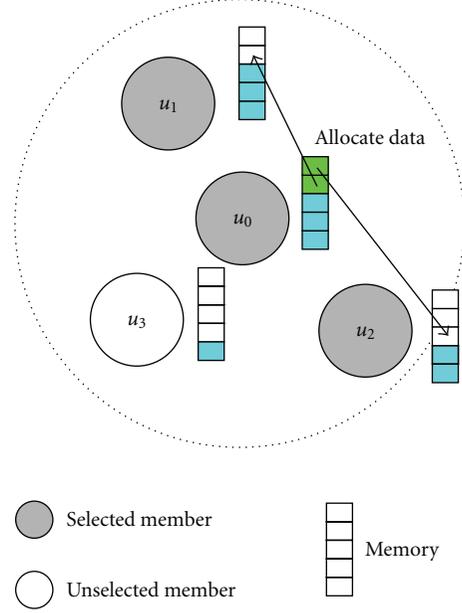


FIGURE 1: An example of the *Allotment Mechanism*. The DH, u_0 , of a DC may distribute its data to some DMs, u_1 and u_2 , in the same DC to free some of its memory space.

Thus, if the $AD[u_j]$ is positive, u_j can temporarily save $AD[u_j]$ bytes data for other sensor nodes; on the other hand, if the $AD[u_j]$ is negative, u_j needs to forward the $AD[u_j]$ bytes data to other sensor nodes. Which nodes will cooperate with u_j for sharing memory is decided by node u_0 through the allotment algorithm, a flowchart of which is presented in Figure 2. Note that the DH in each dominating set will first select argument k to calculate the AD array and then execute the allotment algorithm.

The following theorem says no DM of u_0 has a buffer overflow time less than $D_o[u_0]$. Thus, we can use $D_o[u_0]$ to represent the buffer overflow time of a DC. That is, the *Allotment Mechanism* can more evenly distribute the data to the memory of nodes in a heterogeneous WSN.

Theorem 2. For a given DC, the buffer overflow time of u_j , $j \neq 0$, is not less than the buffer overflow time of u_0 .

Proof. If u_j is not selected to share its memory; that is, $k < j \leq m$, we have

$$\frac{M[0]}{S[0]} \leq \dots \leq \frac{M[u_k]}{S[u_k]} \leq \frac{M[u_j]}{S[u_j]}. \quad (7)$$

Therefore,

$$D_o[u_0] = \frac{\sum_{i=0}^k M[u_i]}{\sum_{i=0}^k S[u_i]} \leq \frac{M[u_j]}{S[u_j]} = T_o[u_j]. \quad (8)$$

On the other hand, the proof is trivial if u_j is designated to share its memory since its buffer overflow time is $D_o[u_0]$. \square

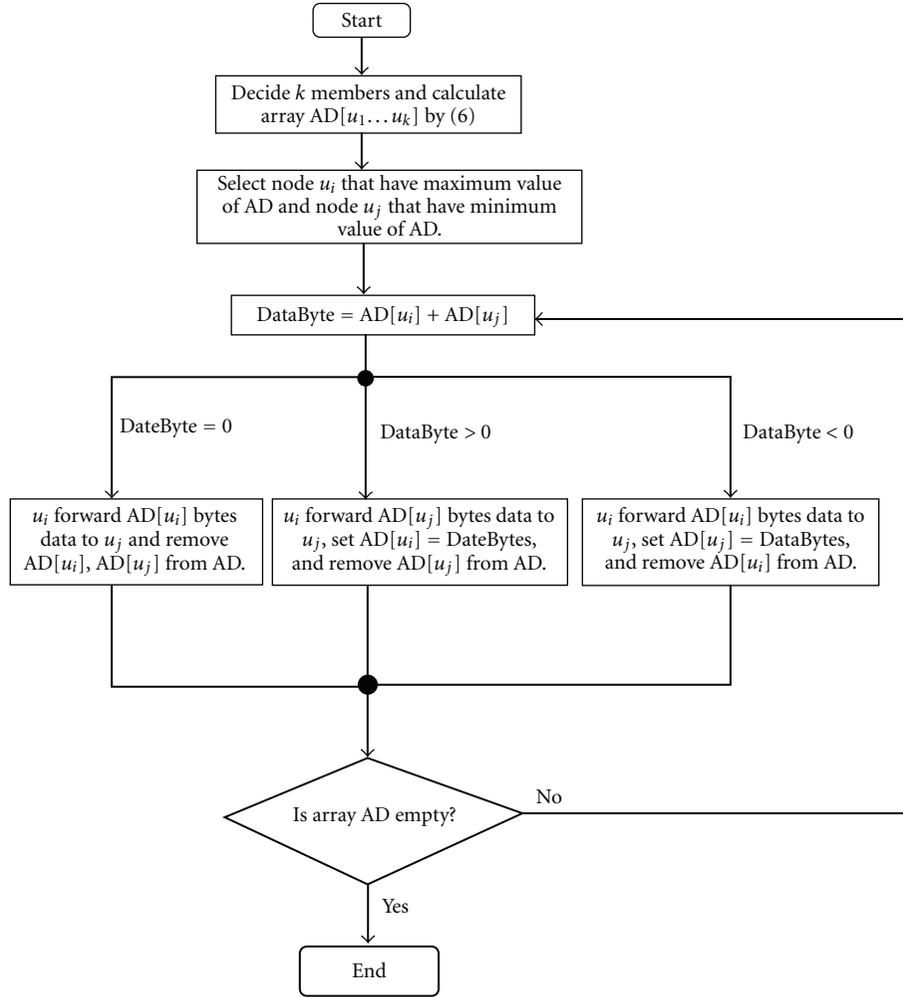


FIGURE 2: Flowchart of the allotment algorithm.

Note that the reason that the proposed *Allotment Mechanism* does not let a *DH* (i.e., u_0) store data on *DMs* with lower sampling rates is that the buffers of these sensors sharing memory cannot be filled slower than the buffer of the sensor having the second highest sampling rate, that is, u_1 . Otherwise, the most critical node of the *DC* becomes u_1 , not u_0 . Therefore, it is necessary to take u_1 into consideration. However, if both u_0 and u_1 store data on *DMs* with the lower sampling rates, it is necessary to consider u_2 for the same reason. Thus, we simply have sensors with higher sampling rates share their buffers to minimize management overhead.

The larger k allows more *DMs* to share their memory at the cost of greater data management and communication overhead, for example, to determine which node has free memory to share and transmit the data via wireless communication. As a result, the *DH* may consume more energy on computation and communication. Therefore, due to the sensor nodes' limited communication and computation resources, k cannot be too large in practical applications. With the *Allotment Mechanism*, every *DC* has a larger buffer overflow time which gives the *MS* more time to collect data or allows nodes to operate with higher sampling rates.

4.4. Traversing Algorithms for Mobile Sinks. Based on the *DH* determined by the TDD algorithm and the buffer overflow time derived from the *Allotment Mechanism*, the *MS* needs to schedule a good traverse path to visit every *DH* in the WSN with the minimum number of missed deadlines. To achieve this goal, we consider a heuristic algorithm and an approximation algorithm, namely, DMWSF and DTSP. Details of these algorithms will be discussed in the following subsections.

4.4.1. Dominating-Based Minimum Weighted Sum First (DMWSF) Algorithm. In the literature [32], the authors use the MWSF algorithm to determine a traverse path for the *MS* to collect data from all sensor nodes. To integrate with TDD and the *Allotment Mechanism*, we modify MWSF as the DMWSF algorithm, in which the *MS* will visit *DHs* only with D_o derived from the *Allotment Mechanism*. With fewer nodes to visit, the *MS* computation load required to find the next visiting *DH* is alleviated. In addition, the increased buffer overflow time D_o results in fewer missed deadlines.

Two factors are considered in the DMWSF algorithm to find the next *DH* to visit. One is the buffer overflow time of

```

(1) while true do
(2)   Calculate the weight value for the current DH and
      every other DH by (9).
(3)   Choose the DH which has the smallest weight value.
(4)   Move to the selected DH and collect all data from the
      DH and its DMs.
(5) end while

```

ALGORITHM 2: DMWSF algorithm.

each *DH*, and the other is the distance between the MS and each *DH*. The MS calculates the weight value for each *DH* based on the following:

$$\text{weight}[i][j] = \alpha \times (D_o[j] - D_o[i]) + (1 - \alpha) \times \text{cost}[i][j]. \quad (9)$$

Here node i is the *DH* that the MS is currently visiting. $\text{Weight}[i][j]$ represents the weight value from current node i to the next *DH*, node j . Because sensor nodes have their clocks synchronized, the difference between the buffer overflow time of nodes i and j may represent the imminence of node j 's *DC*. A negative value of the difference means the missed deadline of j 's *DC*, and it will show the stress when the difference is tiny. On the other hand, the $\text{cost}[i][j]$, defined in Section 3, is the distance between node i and j . α has a value between 0 and 1 and is used to adjust the weight between the distance factor and the buffer overflow time factor. If α is bigger than 0.5, the MS decides the next visiting *DH* mainly based on the buffer overflow deadline; otherwise, the MS considers the next *DH* having shorter distance. Note that, when α is equal to 1, DMWSF is equivalent to the Early Deadline First algorithm.

After calculating the weight value for every *DH* by (9), the MS will move to the *DH* with the smallest weight value and collect the sensing data from the *DH* and its *DMs*. When finished, the whole process will start again (refer to Algorithm 2).

Figure 3 illustrates an example of the DMWSF algorithm. Figure 3(a) shows all the *DCs* as determined by the TDD algorithm, and the MS will use *DHs* as rendezvous points. The MS is located close to the position of node A . After calculating $\text{weight}[A][B]$ and $\text{weight}[A][C]$, the MS will select B as the next visiting node if $\text{weight}[A][B] < \text{weight}[A][C]$, in Figure 3(b).

The DMWSF algorithm is designed to be integrated with the TDD algorithm, thus, alleviating a problem in the MWSF algorithm in which the MS needs to visit all the sensor nodes without requiring much additional communication. With the *Allotment Mechanism*, it may further reduce the buffer overflow problem by having nodes sharing their memory to buffer data, thus, reducing the number of missed deadlines. The simulation results are presented and discussed in detail in Section 5.

4.4.2. Dominating-Based Traveling Salesman Approximation (DTSP) Algorithm. An intuitive approach to traverse all *DHs*

```

(1) select a node  $r \in V$  to be a root node
(2) construct a minimum spanning tree  $T$  for  $G$  from root
       $r$  using MST-PRIM( $G, w, r$ )
(3) let  $L$  be the sequence of vertices in the preorder tree visit
      of  $T$ 
(4) return the traverse path  $H$  that visits the vertices in the
      order  $L$ 

```

ALGORITHM 3: Dominating-based TSP(G, w).

while minimizing the number of missed deadlines is to traverse them in the minimum amount of time. Thus, with a constant moving speed, we want the MS to traverse all *DHs* via the shortest distance. The second traversing algorithm uses a well-known approximation algorithm for TSP [38]. To allow the integration of the TDD and *Allotment Mechanism*, the approximation algorithm is modified and denoted as DTSP, as illustrated in Algorithm 3. Here the traversing problem is modeled as a complete undirected graph $G(V, E)$, and the V is the set of *DHs* and the weight $w[i][j]$ of each edge in E is the derived from (9) for *DH* i and j . Algorithm 3 is widely recognized to be a 2-approximation algorithm since the problems considered are defined in the Euclidean space in which the triangle inequality is satisfied. Note that line 2 of Algorithm 3, *MST-PRIM*, is used to construct a minimum spanning tree for a given graph G . It starts with the root vertex r and chooses the minimum weight edge by w [39].

The MS will continue to visit all *DHs* along the path H . The simulation results are presented and discussed in Section 5.

5. Simulation Environment and Results

To verify the effectiveness and performance of the proposed approach, we conducted various simulations with the GloMoSim [22] network simulator. GloMoSim is a scalable network simulation tool for wired and wireless networks which easily allows the addition of new protocols or the modification of supporting protocols. The algorithms to be compared include Random_Waypoint [16, 31], BISECTOR [17], and MWSF [32]. To serve as a baseline comparison, we also include the data-gathering scheme via static sink, denoted as STATIC.

5.1. Simulation Environment. The objective of our simulations is to compare the number of missed data overflow deadlines and the network lifetime. The network considered is a heterogeneous sensor network with various data-sampling rates. The parameters for the simulation environment are as follows:

- (1) ROI: a 500 m \times 500 m square,
- (2) Node deployment: 50 \sim 300 sensor nodes uniformly and randomly deployed over the ROI.
- (3) data sampling rate: each sensor node is randomly assigned a sampling rate between 0 and 25 (bytes per

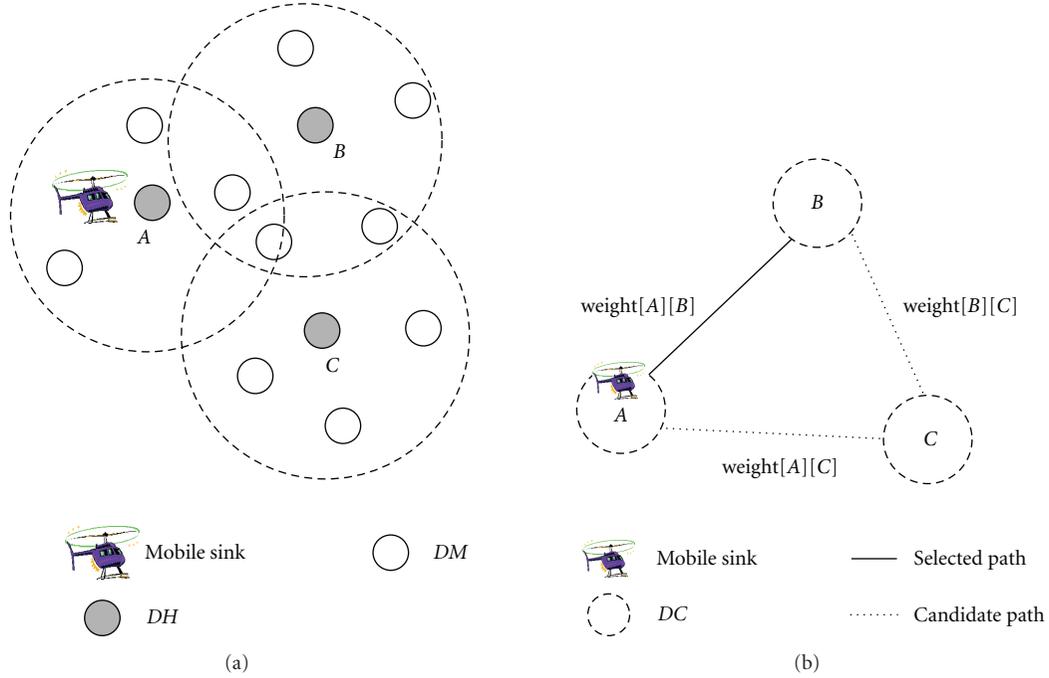


FIGURE 3: An example of the DMWSF Algorithm. (a) All sensor nodes are partitioned into DCs. (b) The MS selects B as the next visiting node since $\text{weight}[A][B] < \text{weight}[A][C]$.

unit of time) and a memory size between 5000 and 10000 bytes,

- (4) energy and communication: the initial energy of each node is 8 Joules and the transmission range is 100 m,
- (5) mobile sink: the MS has the same transmission range as the sensor nodes, and its velocity is 5 m per unit of time. Memory size and energy are unlimited for the MS.

The medium access control (MAC) protocol applied in our simulations is CSMA. The transmission rate for each node is 19.2 Kb per unit of time. The simulation duration is 10000 units of time.

5.2. Energy Consumption Model. Since wireless communication plays a major role in sensor node energy consumption, we only consider the energy consumed for communication and use the energy consumption model adopted in [23], in which the energy needed to transmit a l -bits packet over a distance d is.

$$E_{tx}(d) = E_{Tx.elec} \times l + E_{amp} \times l \times d^2, \quad (10)$$

and the energy needed to receive a l -bits packet is,

$$E_{rx} = E_{Rx.elec} \times l. \quad (11)$$

The values of the parameters are listed in Table 2.

5.3. The α Value . The α value of (9) controls the priority of the data overflow time and the inter-distance of the DHs. If $\alpha > 0.5$, the buffer overflow time will have a higher

TABLE 2: Energy consumption for wireless communication.

Operation	Energy consumption
Transmit electronics ($E_{Tx.elec}$)	50 nJ/bit
Receive electronics ($E_{Rx.elec}$)	
Transmit amplifier (E_{amp})	100 pJ/bit/m ²

priority than the interdistance, and vice versa. To determine which factor has a greater impact on performance, we first conducted simulations to evaluate how α affects the maximum tolerable sampling rate, that is, the maximum sampling rate without any missed deadlines, and the traverse distance.

100 sensor nodes were uniformly and randomly deployed over the ROI. Other parameters of each sensor node are described in Section 5.1. Figure 4 illustrates the simulation results for various α values from 0.05 to 0.9. In Figure 4(a), the y -axis is the maximum tolerable sampling rate (bytes per unit of time). This shows that the WSN's maximum tolerable sampling rate is the lowest when $\alpha = 0.05$ and 0.9 and the highest when α is between 0.4 and 0.5. Thus, to effectively avoid the buffer overflow problem, we should simultaneously consider both factors (i.e., buffer overflow time and interdistance), as ignoring one of these factors will lead to the most severe buffer overflow problem.

In Figure 4(b), the y -axis is the traverse distance of the MS. This shows that the traverse distance is shorter when α is smaller. This is not surprising since the interdistance has higher priority over buffer overflow time when α is small. Based on these results, we set $\alpha = 0.5$ in the

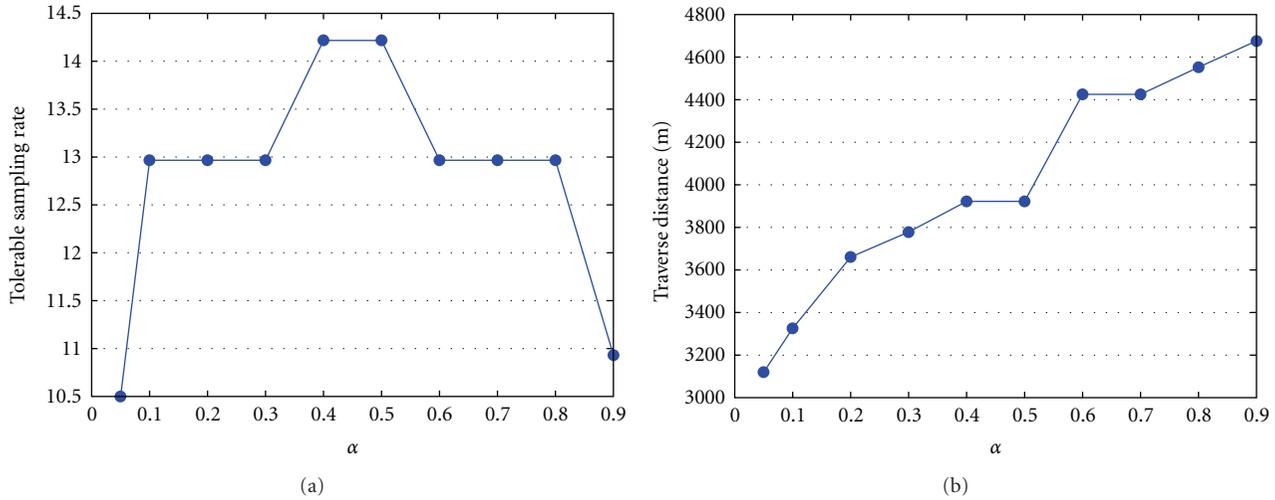


FIGURE 4: (a) The WSN's maximum tolerable sampling rate (bytes per unit of time) is the lowest when $\alpha = 0.05$ and 0.9 and the highest when α is between 0.4 and 0.5 . (b) A smaller α results in shorter traverse distances for the MS.

subsequent simulations so the tolerable sampling rate is maximized without much significantly affecting the MS's traverse distance.

5.4. Maximum Tolerable Sampling Rate and Traverse Distance.

Figure 5(a) compares the maximum tolerable sampling rate against the number of nodes for various data-gathering algorithms, namely, BISECTOR, MWSF, DMWSF with various k values and DTSP with various k values. Here the number of nodes is between 50 and 300, and k is the number of DMs that share their memory space for the *Allotment Mechanism*. When $k = 0$, the *Allotment Mechanism* is not applied.

As seen in Figure 5(a), the maximum tolerable sampling rate decreases (i.e., results in more severe buffer overflow problems) as the number of nodes increases. This is not surprising since there may be more DHs to visit as the number of nodes increases and the sampling rate needs to be reduced to meet all the deadlines. Besides, the larger value of k (i.e., $k = 2$) will lead to a higher maximum tolerable sampling rate, which indicates that the *Allotment Mechanism* can actually alleviate the buffer overflow problem. However, the performance is improved more significantly when the k value increases from zero to one than when it increases from one to two, which suggests that $k = 1$ may be an appropriate value for the *Allotment Mechanism* since DHs have more energy consumption overhead for larger k .

In general, DMWSF and DTSP provide similar performance. On the other hand, the BISECTOR algorithm gathers data by designating turning points. Sensor nodes need to transmit data to one of these turning points via multihop forwarding, which leads to the hot spot problem. This can be alleviated by using lots of turning points (i.e., more than the DHs used in DMWSF and DTSP), to meet all buffer overflow deadlines. Thus, because BISECTOR has more points to traverse, its maximum tolerable sampling rate is lower than that of DMWSF or DTSP. Finally, MWSF has the lowest maximum tolerable sampling rate since it needs to traverse

all sensor nodes. Note that for each simulation nodes are deployed over the same ROI. That is, in MWSF, the MS simply travels the entire ROI to visit every node. Thus, the maximum tolerable sampling rate is independent of the number of nodes but dependent on the size of the ROI.

Figure 5(b) compares the MS's traverse distance against the number of nodes for various data-gathering algorithms. Intuitively, if an algorithm has a shorter traverse distance for the MS, it may have a more tolerable higher sampling rate without any buffer overflow. In MWSF, the MS needs to visit all sensor nodes and, thus, has the longest traverse distance. Note that, in [32], sensor nodes are deployed within concentric circles in which the nodes in the innermost region have the lowest sampling rates and the outer regions can have higher sampling rates. Such a deployment and sampling rate designation allows for a short traverse path. However, in general deployment, MWSF does not provide short traverse distances for the MS.

Figure 5(b) also shows that the traverse distances of BISECTOR, DMWSF, and DTSP increase slightly with the number of nodes. This is because nodes are deployed on the same ROI, and increasing the number of nodes over the same region increases the density but may only slightly increase the number of DHs or turning points. Thus, the traverse distances do not increase significantly. DMWSF and DTSP outperform BISECTOR and MWSF in this metric as well.

Therefore, the dominating-based algorithms, DMWSF and DTSP together with TDD and the *Allotment Mechanism*, can alleviate the buffer overflow problem for randomly deployed heterogeneous sensor networks.

5.5. Lifetime and Energy Consumption. In this section, we compare energy consumption performance for the various algorithms. The ROI contains 50–200 sensor nodes deployed uniformly and randomly. Each sensor node has a data-sampling rate of either 2 or 4 bytes per unit of time. In addition to the data-gathering algorithms mentioned in the

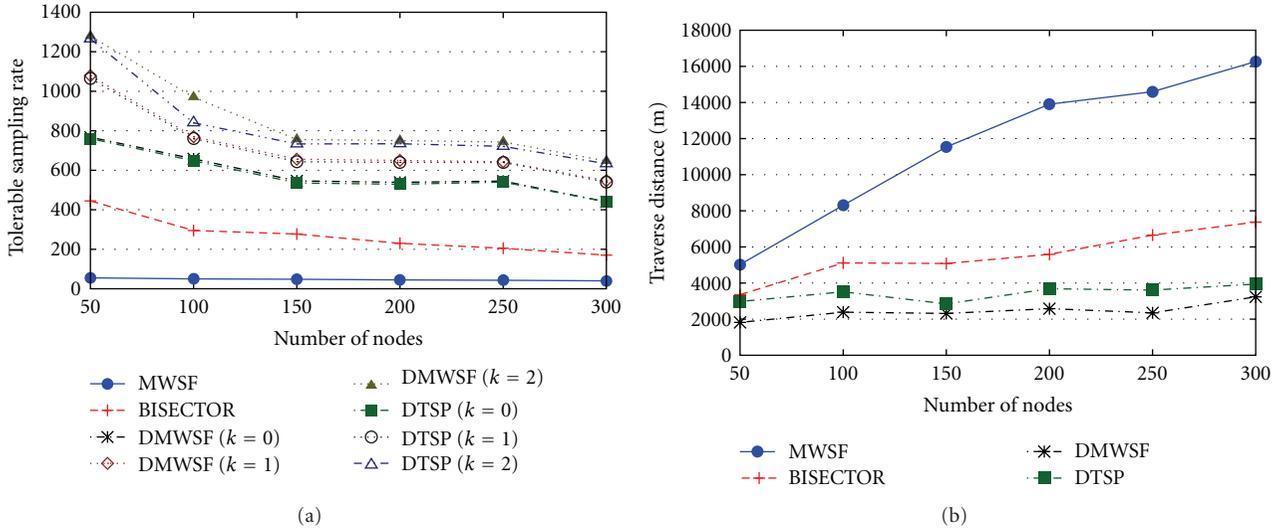


FIGURE 5: (a) Maximum tolerable sampling rate (bytes per unit of time) for each data-gathering approach. (b) MS traverse distance for each data-gathering approach.

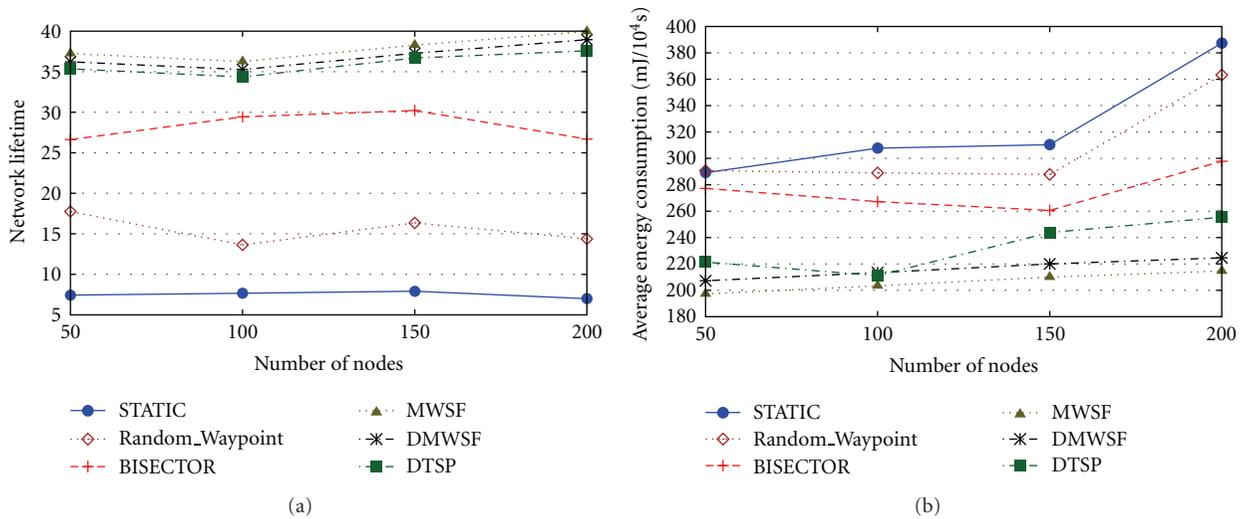


FIGURE 6: (a) Network lifetime (in units of time) for each data-gathering approach. (b) Average energy consumption for each data-gathering approach.

previous subsection, we also include Random_Waypoint and STATIC. In Random_Waypoint, the MS randomly moves straight to a randomly selected destination in the ROI where it collects data from nearby sensor nodes. In STATIC, the data are collected by a static sink via multihop forwarding.

Figure 6(a) compares the network lifetime (in units of time) against the number of nodes for various data-gathering algorithms. Here, the lifetime is defined as the time it takes for a sensor node to exhaust its energy. As Figure 6(a) indicates, STATIC has the shortest network lifetime because of the hot spot problem. Note that Random_Waypoint uses the MS moving straight to a randomly selected destination, but some sensor nodes near the MS also need to relay packets for the other nodes; thus, the hot spot problem still occurs, though with less severity than in STATIC. Thus, the network

lifetime of Random_Waypoint is better than that of STATIC, but worse than others. In a heterogeneous WSN, BISECTOR works the same way as in a WSN in which each node has the same sampling rate (i.e., the MS moves from one turning point to another to collect data from sensor nodes via multihop forwarding). Though the hot spot problem still occurs, the presence of multiple turning points in BISECTOR results in better network lifetime than in Random_Waypoint and STATIC.

Note that MWSF has the longest network lifetime among all data-gathering approaches. This is not surprising since the MS will move to each node to collect data. Thus, there is no need for data forwarding for MWSF. However, as indicated in the previous subsection, MWSF has the lowest maximum tolerable sampling rate, which means that MWSF

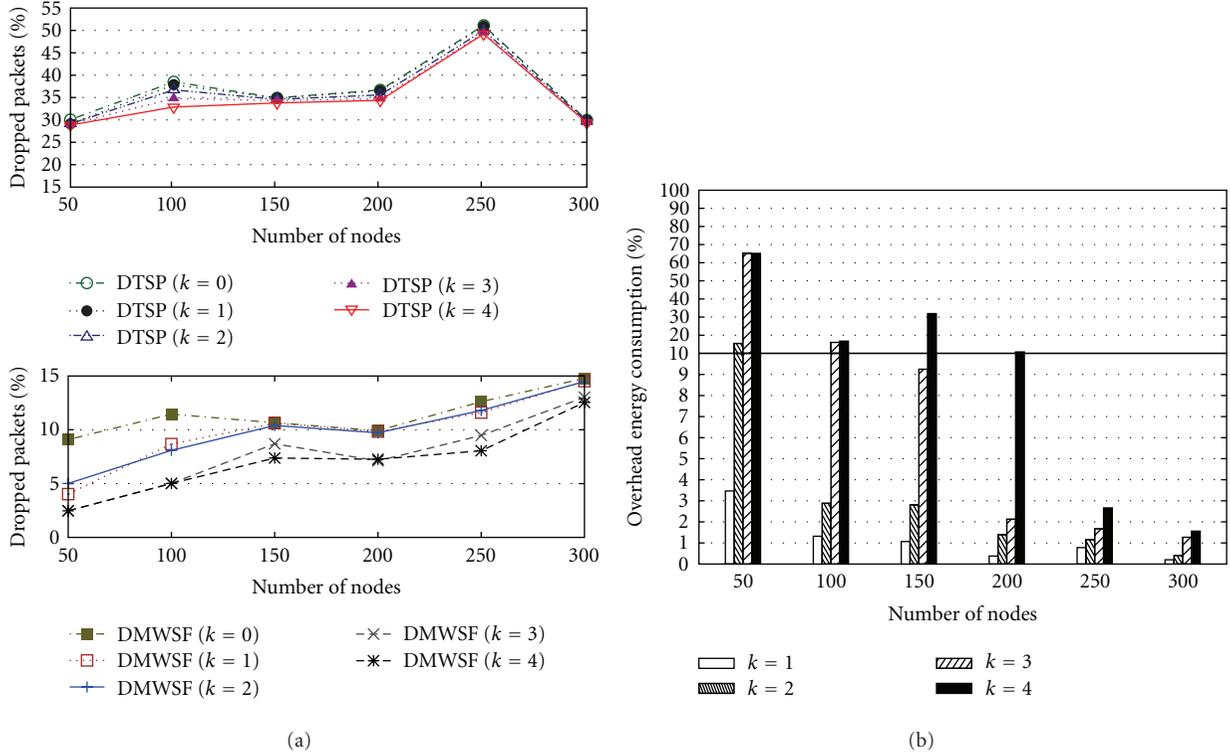


FIGURE 7: (a) Percentage of dropped packets, defined as dropped packets/total packets $\times 100\%$, for different k values. (b) Percentage of overhead energy consumption, defined as energy consumption of *Allotment Mechanism*/total energy consumption $\times 100\%$, for different k values.

has the worst buffer overflow problem. On the other hand, in DMWSF and DTSP, the MS only needs to traverse *DHs* and collect the data from *DMs* via one-hop forwarding. Thus, the energy consumption of DMWSF and DTSP should be close to that of MWSF; that is, DMWSF and DTSP have slightly shorter network lifetimes than does MWSF, as seen in Figure 6(b). However, as discussed in the previous subsection, both DMWSF and DTSP have much better maximum tolerable sampling rates than does MWSF, which makes them promising schemes for data gathering. Figure 6(b) compares the average energy consumption against the number of nodes for the various data-gathering algorithms. It is not surprising that STATIC has the worst energy consumption since it requires so much multihop forwarding. The energy consumption of Random_Waypoint is also high for the same reason. As in Figure 6(b) above, MWSF has the best energy efficiency, closely followed by DMWSF and DTSP. Again, considering the buffer overflow problem, DMWSF and DTSP are promising schemes for data gathering.

5.6. Performance of the Allotment Mechanism with Different k Values. We also conducted simulations to illustrate the impact of k values on the *Allotment Mechanism*. Here, each sensor node is randomly assigned a sampling rate between 0 and 25 bytes per unit of time, and the duration is 10000 units of time. We compared the performance of DMWSF and DTSP with $k = 0-4$. Figure 7(a) shows the percentage of dropped packets due to buffer overflow against the total

number of nodes. As indicated, the percentage of dropped packets decreases as the k value increases. However, as illustrated in Figure 7(b), the percentage of overhead energy consumption indicates that the buffer overflow problem is alleviated at the cost of increased energy consumption. In our experiments, the *Allotment Mechanism* with $k \geq 3$ can no longer effectively alleviate the buffer overflow problem but only consumes more energy.

6. Conclusions and Future Work

Typical WSNs use static sinks to collect data from all sensor nodes via multihop forwarding, which can easily result in the hot spot problem since nodes close to the sink tend to consume more energy in relaying data from other nodes. This can exhaust the close nodes, leaving the sinks isolated from the rest of network and the remaining nodes underutilized.

An MS can prevent the hot spot problem, but it takes time to move around the ROI to collect data. A poorly designed traverse path may result in the buffer overflow problem since the MS cannot arrive at nodes in time to collect the data buffered in their memory, necessitating the dropping of some information.

Our proposed approach addresses the hot spot problem using an MS to collect data, but the MS only traverses the rendezvous points, achieved by TDD, where every node is within a one-hop communication range of a rendezvous

point (i.e., the dominating set). Reducing the number of points to traverse reduces the time needed to traverse them, thus, alleviating the buffer overflow problem. The traverse path is determined by DMWSF or DTSP, in which the traverse cost consists two factors: the buffer overflow time and interdistance. The weighting between these two factors is controlled by the α value. Furthermore, we proposed the *Allotment Mechanism* that allows the nodes with higher sampling rates to temporarily buffer their data in the memory of their one-hop neighbors with lower sampling rates, thus, extending the buffer overflow deadline which further alleviates the buffer overflow problem.

The effectiveness of proposed approach was verified via the GloMoSim network simulator. Simulation results show that our approach incurs fewer buffer overflows than other data-gathering schemes such as BISECTOR and MWSF. Moreover, the simulation results of α value test suggest that $\alpha = 0.4\text{--}0.5$ has the least buffer overflow problem, which means that both buffer overflow time and interdistance need to be considered when planning a traverse path for an MS. In addition, the buffer overflow problem can be alleviated with a larger k at the cost of increased energy consumption. However, our simulation results show that the *Allotment Mechanism* with $k \geq 3$ can no longer effectively alleviate the buffer overflow problem but only consumes more energy.

Finally, in future work we plan to design a more efficient *Allotment Mechanism* for large-scale wide sensor network environments, for example, using a topology control supported in the IEEE 802.15.4 standard [40]. We will also consider the possibility of using adaptive k values. For example, information loss may result if a DH is too far from the initial position of the MS. We will study whether this problem can be alleviated by using k correlated with the distance from the DH to the initial position of the MS. In addition, we will investigate more recent and efficient simulation platforms, such as the OMNeT++ simulator [41] which gives a more realistic behavior in WSNs. With more realistic propagation models and different MAC protocols, we may verify the effectiveness of our proposed approach under unreliable communications conditions.

Acknowledgment

This research was supported by National Science Council (NSC), Taiwan, ROC, under Grant NSC 99-2221-E-194-021. The authors gratefully acknowledge this support.

References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [2] A. 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*, pp. 68–77, ACM, September 2002.
- [3] J. R. Polastre, *Design and implementation of wireless sensor networks for habitat monitoring*, M.S. thesis, University of California at Berkeley, 2003.
- [4] A. Chehri, P. Fortier, and P.-M. Tardif, "Security monitoring using wireless sensor networks," in *Proceedings of the 5th Annual Conference on Communication Networks and Services Research*, pp. 13–17, IEEE Computer Society, Washington, DC, USA, 2007.
- [5] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," in *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems*, vol. 37, pp. 96–107, October 2002.
- [6] W. Du, L. Fang, and N. Peng, "LAD: localization anomaly detection for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 66, no. 7, pp. 874–886, 2006.
- [7] H. W. Tsai, C. P. Chu, and T. S. Chen, "Mobile object tracking in wireless sensor networks," *Computer Communications*, vol. 30, no. 8, pp. 1811–1825, 2007.
- [8] B. Sun, L. Osborne, Y. Xiao, and S. Guizani, "Intrusion detection techniques in mobile ad hoc and wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 5, pp. 56–63, 2007.
- [9] T. Miyazaki, R. Kawano, Y. Endo, and D. Shitara, "A sensor network for surveillance of disaster-hit region," in *Proceedings of the 4th International Symposium on Wireless and Pervasive Computing*, pp. 1–6, February 2009.
- [10] T. Park, D. Kim, S. Jang, S. E. Yoo, and Y. Lee, "Energy efficient and seamless data collection with mobile sinks in massive sensor networks," in *Proceedings of the 23rd IEEE International Parallel and Distributed Processing Symposium*, pp. 1–8, May 2009.
- [11] Y. S. Chen, S. Y. Ann, and Y. W. Lin, "VE-mobicast: a variant-egg-based mobicast routing protocol for sensor networks," *Wireless Networks*, vol. 14, no. 2, pp. 199–218, 2008.
- [12] M. Halkidi, V. Kalogeraki, D. Gunopulos, D. Papadopoulos, D. Zeinalipour-Yazti, and M. Vlachos, "Efficient online state tracking using sensor networks," in *Proceedings of the 7th International Conference on Mobile Data Management*, IEEE Computer Society, Washington, DC, USA, 2006.
- [13] C. Weng, M. Li, and X. Lu, "Data aggregation with multiple spanning trees in wireless sensor networks," in *Proceedings of the International Conference on Embedded Software and Systems*, pp. 355–362, July 2008.
- [14] T. L. Sheu and W. C. Liu, "An adaptive data collection scheme for mobile sinks in a grid-based wireless sensor network," in *Proceedings of the 3rd International Conference on Communications and Networking in China*, pp. 382–386, August 2008.
- [15] R. Yu, X. Wang, and S. Das, "Efficient data gathering using mobile elements in partially connected sensor networks," in *Proceedings of the Chinese Control and Decision Conference*, pp. 5337–5342, July 2008.
- [16] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 30–41, 2003.
- [17] M. Ma and Y. Yang, "SenCar: an energy-efficient data gathering mechanism for large-scale multihop sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 10, pp. 1476–1488, 2007.
- [18] G. Xing, T. Wang, W. Jia, and M. Li, "Rendezvous design algorithms for wireless sensor networks with a mobile base station," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing 2008*, pp. 231–239, May 2008.

- [19] D. Mandala, X. Du, F. Dai, and C. You, "Load balance and energy efficient data gathering in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 8, no. 5, pp. 645–659, 2008.
- [20] D. Cokuslu, K. Erciyes, and O. Dagdeviren, "A dominating set based clustering algorithm for mobile ad hoc networks," in *Proceedings of International Conference on Computational Science*, pp. 571–578, 2006.
- [21] J. Wu and H. Li, "A dominating-set-based routing scheme in ad hoc wireless networks," *Telecommunication Systems*, vol. 3, pp. 63–84, 1999.
- [22] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla, "GloMoSim: a scalable network simulation environment," Tech. Rep. 990027, UCLA Computer Science Department, 1999.
- [23] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, IEEE Computer Society, Washington, DC, USA, 2000.
- [24] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach," in *Proceedings of the 23rd Conference of the IEEE Communications Society*, pp. 629–640, March 2004.
- [25] Y. Gu, D. Bozdag, E. Ekici, F. Ozguner, and C.-G. Lee, "Partitioning based mobile element scheduling in wireless sensor networks," in *Proceedings of the Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pp. 386–395, 2005.
- [26] R. Wohlers, N. Trigoni, R. Zhang, and S. A. Ellwood, "TwinRoute: energy-efficient data collection in fixed sensor networks with mobile sinks," in *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware*, pp. 192–201, 2009.
- [27] G. S. Chhabra and D. Sharma, "Cluster-tree based data gathering in wireless sensor network," *International Journal of Soft Computing and Engineering*, vol. 1, pp. 27–32, 2011.
- [28] K. D. Samuel, S. M. Krishnan, K. Y. Reddy, and K. Suganthi, "Improving energy efficiency in wireless sensor network using mobile sink," in *Advances in Networks and Communications*, N. Meghanathan, B. K. Kaushik, and D. Nagamalai, Eds., vol. 132 of *Communications in Computer and Information Science*, pp. 63–69, Springer, Berlin, Germany, 2011.
- [29] Z. Chen, S. Liu, and J. Huang, "Multi-tier grid routing to mobile sink in large scale wireless sensor networks," *Journal of Networks*, vol. 6, pp. 765–773, 2011.
- [30] L. He, Z. Chen, and J.-D. Xu, "Optimizing data collection path in sensor networks with mobile elements," *International Journal of Automation and Computing*, vol. 8, pp. 69–77, 2011.
- [31] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting mobility for energy efficient data collection in wireless sensor networks," *Mobile Networks and Applications*, vol. 11, no. 3, pp. 327–339, 2006.
- [32] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines," in *Proceedings of the 25th IEEE International Real-Time Systems Symposium, Washington*, pp. 296–305, IEEE Computer Society, Washington, DC, USA, 2004.
- [33] J. Rao and S. Biswas, "Data harvesting in sensor networks using mobile sinks," *IEEE Wireless Communications*, vol. 15, no. 6, pp. 63–70, 2008.
- [34] E. M. Saad, M. H. Awadalla, and R. R. Darwish, "A data gathering algorithm for a mobile sink in large-scale sensor networks," in *Proceedings of the 4th International Conference on Wireless and Mobile Communications*, pp. 207–213, August 2008.
- [35] F. Dai and J. Wu, "Distributed dominant pruning in Ad Hoc networks," in *Proceedings of the 2003 International Conference on Communications*, pp. 353–357, May 2003.
- [36] B. Han, H. Fu, L. Lin, and W. Jia, "Efficient construction of connected dominating set in wireless ad hoc networks," in *Proceedings of the 2004 IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 570–572, October 2004.
- [37] Y. Bi, J. Niu, L. Sun, W. Huangfu, and Y. Sun, "Moving schemes for mobile sinks in wireless sensor networks," in *Proceedings of the 27th IEEE International Performance Computing and Communications Conference*, pp. 101–108, April 2007.
- [38] Concorde TSP Solver. <http://www.tsp.gatech.edu/concorde.html>.
- [39] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Systems Technical Journal*, vol. 36, pp. 1389–1401, 1957.
- [40] C. Buratti, M. Martalò, R. Verdone, and G. Ferrari, *Sensor Networks with IEEE 802.15.4 Systems*, Springer, 2011.
- [41] OMNeT++ Network Simulation Framework. <http://www.omnetpp.org/>.

Research Article

The Complexity of the Minimum Sensor Cover Problem with Unit-Disk Sensing Regions over a Connected Monitored Region

Ren-Song Ko

Department of Computer Science and Information Engineering, National Chung Cheng University, 168 University Road, Min-Hsiung Chia-Yi 621, Taiwan

Correspondence should be addressed to Ren-Song Ko, korenson@cs.ccu.edu.tw

Received 17 June 2011; Revised 12 August 2011; Accepted 16 August 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Ren-Song Ko. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper considers the complexity of the Minimum Unit-Disk Cover (MUDC) problem. This problem has applications in extending the sensor network lifetime by selecting minimum number of nodes to cover each location in a geometric connected region of interest and putting the remaining nodes in power saving mode. MUDC is a restricted version of the well-studied Minimum Set Cover (MSC) problem where the sensing region of each node is a unit-disk and the monitored region is geometric connected, a well-adopted network model in many works of the literature. We first present the formal proof of its NP-completeness. Then we illustrate several related optimum problems under various coverage constraints and show their hardness results as a corollary. Furthermore, we propose an efficient algorithm for reducing MUDC to MSC which has many well-known algorithms for approximated solutions. Finally, we present a decentralized scalable algorithm with a guaranteed performance and a constant approximation factor algorithm if the maximum node density is fixed.

1. Introduction

The research in wireless ad hoc networks has rapidly grown in recent years due to their applications in civil and military domains. Combined with recent developments in micro-electro-mechanical systems and low-cost mass production, various small and low-power devices that integrate sensors with limited on-board processing and wireless communication capabilities begin to emerge. Hence, wireless networks with large numbers of sensors become possible and open up potential of many new applications, such as environment monitoring and surveillance [1].

With the available technology, the sensors are usually battery powered. Due to size and cost constraints, the energy available at each sensor is limited. Therefore, one of the important design considerations in sensor networks is to minimize energy consumption and prolong network lifetime. There is a significant amount of the literature addressing the issue of efficient energy management in generic wireless ad hoc networks from various perspectives, such as medium access control [2, 3], routing [4, 5], broadcasting [6, 7], multicasting [8, 9], and topology control [10, 11]. Of course,

similar approaches have been also considered in wireless sensor networks [12–15].

An alternative approach commonly adopted in sensor networks is based on scheduling sensor activity so that some nodes may enter the power saving mode while the remaining active nodes can still provide continuous service [14, 16]. For instance, if all the sensor nodes simultaneously operate in active mode, an excessive amount of energy is wasted and the data collected is highly correlated and redundant. In addition, multiple packet collisions may occur when all the sensors in a certain region try to transmit as a result of a triggering event. Several research results [3, 16] illustrate that a mode of operation alternating active and inactive battery states has a significant reduced energy consumption.

However, such scheduling schemes may face new constraints about sensing coverage introduced by their distributed sensing applications [17]. For example, surveillance applications may require each location of monitored regions to be covered by at least one sensor, while many stronger environmental monitoring, such as military applications, require multiple sensors for fault-tolerant purpose. Besides, triangulation positioning-based tracking applications [18, 19]

may require at least three sensors at any locations. Data sampling applications may require a given percentage of monitored regions to be covered.

Therefore, this paper considers the scheduling approach that extends the network lifetime by minimizing the number of active nodes while maintaining coverage constraints. As mentioned earlier, the advantage of this approach is that less packet collisions may occur since less spatially close sensors try to transmit highly correlated and redundant information as a result of a triggering event. Hence, the lifetime of each sensor cover may be extended.

We model a sensor network as a 2D geometric connected region monitored by a set of deployed sensor nodes with unit-disk sensing regions, a realistic assumption that is well adopted in many network models. The coverage constraint is that each location of the 2D region is covered by at least one active node. Note that optimum sensor cover problems may be solved by partitioning monitored regions into disjoint sectors [20, 21]. Here a sector is a maximum region covered by the same set of nodes. Hence the minimum sensor cover problem is transformed to the Minimum Set Cover (MSC) problem which is NP-complete [22] and has been studied extensively in the literature [23–26]. However, with the additional unit-disk sensing region and geometric connected monitored region restrictions, the problem considered here is only a restricted version of MSC, and, to the best of our knowledge, its complexity is still unknown. (That is, it is not trivial to transform each instance of MSC to an instance in the minimum sensor cover problem with unit-disk sensing regions over a connected monitored region in polynomial time.) Thus, we will answer this fundamental question and present the formal proof of its NP-completeness. Furthermore, we illustrate several related optimum problems under different coverage constraints and show their hardness results as a corollary.

Next, we propose the arc sampling algorithm which may effectively and efficiently reduce MUDC to MSC. Consequently, many well-known algorithms can be applied to find approximated solutions. In addition, we present a decentralized scalable algorithm with a guaranteed performance, and a constant approximation factor algorithm if the maximum node density is fixed. Finally, we illustrate simulation results to evaluate the proposed algorithms.

2. Related Work

2.1. Coverage Problems. Meguerdichian et al. [17] defined the coverage problems from several different application domains including deterministic, statistical, worst, and best cases. They also presented optimum polynomial time algorithms to evaluate paths that are the best and least monitored in the sensor network. The work in [27] further defined the exposure problem as measure of how well an object can be observed by the sensor network while it moves along an arbitrary path with an arbitrary velocity. A localized exposure-based coverage algorithm was proposed in [28] for finding the minimal exposure path between two points.

Furthermore, Gui and Mohapatra [29] considered the object tracking applications in which networks operate between

surveillance state and tracking state. During surveillance state, they devised a set of metrics for quality of surveillance for detecting moving objects and quantify the trade-off between power conservation and quality. They also proposed an algorithm for each node to determine when to wake up or sleep during the tracking stage.

Tian and Georganas [30] developed a coverage-preserving scheduling scheme to reduce energy consumption by turning off some redundant nodes based on some eligibility rules. Carbutar et al. [31] proposed distributed algorithms for detecting and eliminating redundancy in a sensor network while preserving the network's coverage via Voronoi diagrams, even in cases of sensor failures or insertion of new sensors.

Huang and Tseng [32] considered the k -coverage problem to determine whether every point in the monitored region is covered by at least k nodes. They reduced this problem to the perimeter-coverage problem which determines the coverage degree of the perimeter of each node's sensing region and presented polynomial-time algorithms in the number of nodes.

In addition to coverage, connectivity also needs to be assured to make sensor networks successfully. It has been shown in [33] that if the communication range of sensors is at least twice as large as their sensing range, then full coverage of a convex region implies connectivity. Wang et al. [34] presented a Coverage Configuration Protocol (CCP) that allows the network to self-configure dynamically to achieve guaranteed degrees of coverage and connectivity.

2.2. Minimum Sensor Cover Problems. In [21], Funke et al. proposed the greedy sector cover algorithm which selects a node that covers the maximum number of uncovered sectors at each iteration step. That is, the problem is reduced to MSC and solved by the greedy algorithm. They proved that the well-known approximation factor $O(\log m)$ remains tight in this restricted version. Here m is the maximum number of sectors covered by a single node. To obtain better approximation factors, they also presented a grid placement algorithm and a distributed dominating cover algorithm. These two algorithms have constant approximation factors, but cannot guarantee the full coverage.

Gupta et al. [35] designed an $O(\log |N|)$ centralized approximation algorithms with the connectivity constraint. Here $|N|$ is the number of sensor nodes. In their definition of the sensor cover problem, the sensing region can take any convex shape. They also mentioned that such a problem is NP-hard as the less general problem of covering discrete points using line segments is known to be NP-hard [36]. On the other hand, the sensing region considered in this paper is restricted to a unit-disk, which is well adopted in many network models. We will prove such a problem remains NP-complete even with the unit-disk restriction. They also proposed a distributed algorithm based on node priorities, but did not provide any guarantee on the solution size.

2.3. Related Optimum Problems. Fowler et al. [37] proved the NP-completeness of the Box Cover problem which aims at

finding the minimum number of identical rectangles to cover a set of given points. Similarly, Megiddo and Supowit [38] considered the Circle Covering problem which is equivalent to the Geometric Disc Covering problem, that is, to find the minimum number of identical disks to cover a set of given points. There are two fundamental differences between MUDC and these two problems. In these two problems, the covered object is a set of discrete points but not a connected region. Hence, in the proofs of the NP-completeness, we have less flexibility in the connected case than in the discrete cases, since we need to ensure the monitored region is connected while constructing a problem instance. Furthermore, the two problems have the flexibility to determine the “good” locations of covering objects (rectangles or disks), which could be anywhere on the plane. On the other hand, in MUDC, the locations of disks are pre-deployed.

Marathe et al. [39] considered several basic optimization problems for unit-disk graphs with hierarchical structures. They presented a general technique to prove the hardness results of several problems. The hardness of these problems, including Box Cover and Circle Covering, was proved via satisfiability problems. The reduction strategy was to use some geometric structures to represent variables. Each clause is represented by a special structure that “glues” the corresponding structures of the variables in the clause.

There are several polynomial approximation algorithms [40–42] for the Geometric Disc Covering problem. Franceschetti et al. pointed out in [43] that the number of possible disk positions can be bounded if any disk that covers at least two points has two of these points on its border. Hence, by performing a search on a subset of the possible disk positions, the running time of these algorithms becomes polynomial and the solution sizes are guaranteed. They also gave a detailed comparison of these algorithms in [44].

3. Preliminaries

In this section, we define the Minimum Unit-Disk Cover (MUDC) problem that aims at finding the least number of nodes with unit-disk sensing regions to fully cover a designated connected region. We prove this problem is intractable; that is, it belongs to the NP-complete class.

Definition 1. Consider a two-dimensional Euclidean metric space \mathbb{E} , the unit-disk sensing region of a given node $n \in \mathbb{E}$ is defined as $\text{disk}(n) = \{x \in \mathbb{E} \mid d(x, n) \leq 1\}$. (In the context of discussing MUDC, we represent a node by its geometric location without any confusion.) Here $d(x, n)$ is the distance in Euclidean metric between x and n . Furthermore, the unit-disk sensing region of a set U of nodes is defined as $\text{disk}(U) = \bigcup_{n \in U} \text{disk}(n)$.

Definition 2. A two-dimensional finite region A is said to be *unit-disk covered* by a set U of nodes in a two-dimensional Euclidean metric space if $A \subseteq \text{disk}(U)$. Furthermore, U is called a *unit-disk cover* (UDC) of A .

The objective is to find the minimum unit-disk cover (MUDC) of A . Note that the optimum problems discussed in

this paper could be solved by their associated decision problems in polynomial time. Therefore, we discuss the decision version of MUDC instead, and it can be formally stated in the following.

Problem 3 (MUDC). Given a set N of nodes in a two-dimensional Euclidean metric space \mathbb{E} , a two-dimensional geometric connected finite region $A \subset \mathbb{E}$, and a positive integer K , determine whether there is a subset $U \subseteq N$ with $|U| \leq K$ such that $A \subseteq \text{disk}(U)$. Here $|U|$ is the cardinality of U .

For simplicity’s sake, the geometry of an MUDC problem, that is, the region A and the set N of nodes, is denoted as (A, N) .

Thus, we will prove the following theorem.

Theorem 4. *MUDC is NP-complete.*

The NP-completeness of MUDC will be proved by reduction from the Planar 3-SAT problem, which is known to be NP-complete [45].

Problem 5 (Planar 3-SAT, P3SAT). Given a set of variables $V = \{v_1, v_2, \dots, v_{\eta'}\}$ and a set of clauses $C = \{c_1, c_2, \dots, c_{\eta}\}$ over V such that each $c \in C$ has $2 \leq |c| \leq 3$ (denoted as a boolean formula B) determine whether there is an assignment for the variables so that all clauses are satisfied. (In Lichtenstein’s NP-completeness proof of P3SAT [45], an instance of 3SAT is transformed to an instance of P3SAT with $|c|$ being 2 or 3. Thus, the restriction, $2 \leq |c| \leq 3$, does not change the complexity of the problem. This restriction is required to prove Lemma 17.) Furthermore, the bipartite graph $G_B = \{V \cup C, E\}$ (in this NP-completeness proof of MUDC, G_B will be used to construct an equivalent MUDC problem for B) is planar, where $E = \{(v_i, c_j) \mid v_i \in c_j \text{ or } \bar{v}_i \in c_j\}$. (We remove the edges $\{(v_i, v_{i+1}) \mid 1 \leq i < m'\} \cup \{(v_{m'}, v_1)\}$ without any changes in the difficulty of the problem [46].)

That is, let B be a boolean formula in P3SAT with η clauses and η' variables. We wish to construct an equivalent MUDC problem with the geometry $\text{MUDC}(B) = (A_B, N_B)$ where A_B and N_B are the region and the set of nodes transformed from G_B , respectively.

Inspired by Lichtenstein’s NP-completeness proof of the Geometric Connected Dominating Set problem [45], $\text{MUDC}(B) = (A_B, N_B)$ is constructed via structures. Each structure S is a geometry containing a polygon A and a set of nodes N and denoted as $S = (A, N)$. Variables, clauses, and edges of G_B are represented by various structures. Hence, $\text{MUDC}(B)$ is constructed from G_B by replacing variables, clauses, and edges with their corresponding structures.

Each structure $S = (A, N)$ is constructed in such a way that N can be partitioned into two disjoint subsets of equal size, denoted as N^+ and N^- , and the MUDC of A , except the ones representing clauses, is either N^+ or N^- . Thus, a variable is assigned *true* corresponding to that N^+ is the MUDC of A ; *false* corresponds to N^- . For convenience throughout this paper, we assign each node a polarity. The node n has positive polarity if $n \in N^+$ or negative polarity if $n \in N^-$.

The property of structures mentioned above can be formally defined in the following.

Definition 6. One calls a structure $S = (A, N)$ *well aligned* if $A \subseteq \text{disk}(N^+)$ and $A \subseteq \text{disk}(N^-)$.

Definition 7. For a structure $S = (A, N)$ and $N' \subseteq N$, we call S *partially well behaved on N'* , if the following preconditions hold:

- (i) $|N'^+| = |N'^-| = |N'|/2$
- (ii) if $U \subseteq N$ is a UDC of A , $|U \cap N'| \geq |N'|/2$
- (iii) if $U \subseteq N$ is an MUDC of A , $U \cap N' = N'^+$ or $U \cap N' = N'^-$. (For a given set of nodes N , we denote the set of nodes with the same polarity as N with superscripts $+$ or $-$ throughout this paper. i.e., $N^+ = \{n \in N \mid n \text{ has positive polarity}\}$ and $N^- = \{n \in N \mid n \text{ has negative polarity}\}$.)

Furthermore, we call S *well behaved* if S is partially well behaved on N .

Note that, from the above definition, if $S = (A, N)$ is well behaved and $U \subseteq N$ is an MUDC of A , then $|U| = |N|/2$ and U only contains the nodes with the same polarity.

The NP-completeness proof of MUDC will proceed as follows.

- (1) Describe structures representing variables, edges, and clauses. These structures have the properties defined in Definitions 6 and 7.
- (2) Describe how the above structures may be connected together to represent G_B while preserving the properties defined in Definitions 6 and 7. Here the resulting composite structure is $\text{MUDC}(B) = (A_B, N_B)$.
- (3) We claim that B is satisfiable if and only if A_B can be covered by half the nodes of N_B . In the proof of the claim, the properties defined in Definitions 6 and 7 will be used in the forward direction and the backward direction respectively.

4. NP-Completeness Proof of MUDC

We first prove that MUDC belongs to the NP class. This could be done since a nondeterministic algorithm needs only guess a set of nodes, U , and verify whether $A \subseteq \text{disk}(U)$. Besides, as stated in [32], this verification could be done in $O(|U|^2 \log |U|)$.

We continue the proof by reduction from the Planar 3-SAT problem. Let B be a boolean formula in P3SAT with η clauses and η' variables. We wish to construct an equivalent MUDC problem with the geometry $\text{MUDC}(B) = (A_B, N_B)$ transformed from the bipartite graph G_B .

4.1. Structures. We encode each variable by the structure, denoted as $S_v = (A_v, N_v)$, shown in Figure 1. A_v represents the shaded region which is a $d_v \times 1$ rectangle. N_v represents the set of the $2(d_v + 1)$ nodes which are positioned accordingly and used to cover A_v . Each node has a either positive or

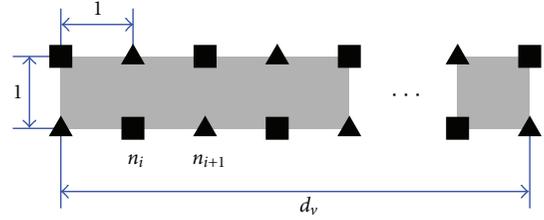


FIGURE 1: The structure representing a variable.

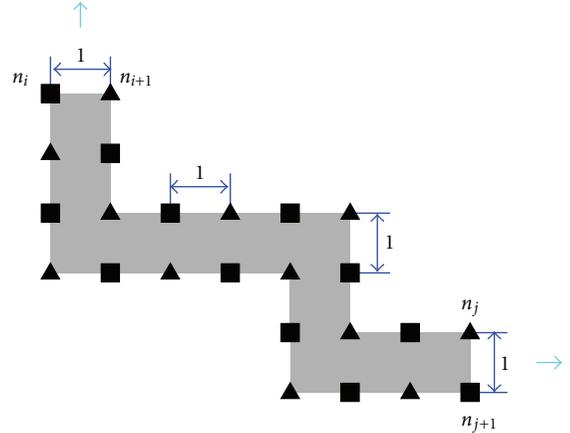


FIGURE 2: The structure representing an edge.

negative polarity and is represented by a square or triangle, respectively, in the figure. The d_v may be long enough to prevent unwanted interactions between nearby edge structures.

Next, we may encode an edge by a strip-like structure, denoted as $S_e = (A_e, N_e)$, which may extend horizontally and vertically. Figure 2 illustrates an example. The shaded region, denoted as A_e , is composed of rectangles. N_e represents the set of the positive and negative polar nodes which are positioned accordingly and used to cover A_e .

It is not hard to prove the structures S_v and S_e satisfy the following lemma.

Lemma 8. *The structures $S_v = (A_v, N_v)$ shown in Figure 1 and $S_e = (A_e, N_e)$ shown in Figure 2 are well aligned and well behaved.*

Proof. The lemma may be proved by induction. For the sake of brevity, the complete proof is given in Appendix A. \square

Each clause c may be represented by a structure, called an n -way connector and denoted as $S_c = (A_c, N_c(\mathcal{P}, H))$. Here $n = |c|$ and could be the value of 2 and 3. Figure 3 illustrates the possible realization of n -way connectors. A_c represents the shaded polygon that will be covered by the set of the nodes, N_c . The geometries of A_c and relative positions of nodes of N_c are shown in Figures 3 and 4 and Tables 1 and 2.

Furthermore, N_c is divided into n disjoint partitions $P_i \subseteq N_c$ with $1 \leq i \leq n$ and we denote $\mathcal{P} = \{P_1, P_2, \dots, P_n\}$. As shown in Figure 3, each node is labeled as an alphabet

TABLE 2: The geometry of the 3-way connector shown in Figure 3(b).

N_c		A_c	
Node	Position	Vertex	Position
n_1	(0, 0)	v_1	$(\frac{1}{20}, 0)$
n_2	(1, 0)	v_2	(1, 0)
n_3	(1, 1)	v_3	(1, 1)
n_4	(1, 2)	v_4	$(\frac{1}{2}, 1)$
n_5	(1, 3)	v_5	(1, 2)
n_6	(0, 3)	v_6	(1, 3)
n_7	(-1, 3)	v_7	(0, 3)
n_8	(-1, 2)	v_8	$(-\frac{2}{5}, 2\frac{4}{5})$
n_9	(0, 2)	v_9	(-1, 3)
n_{10}	$(0, 1\frac{1}{10})$	v_{10}	(-1, 2)
		v_{11}	$(-\frac{7}{20}, 2\frac{1}{4})$
		v_{12}	$(-\frac{1}{5}, 1\frac{3}{4})$
		v_{13}	$(\frac{1}{20}, 2\frac{1}{10})$
		v_{14}	$(\frac{1}{5}, 1\frac{1}{2})$

Lemma 9. Each n -way connector, $S_c = (A_c, N_c \langle \mathcal{P}, H \rangle)$, of Figure 3 has the following properties.

- (i) For all $P_i \in \mathcal{P}$, S_c is partially well behaved on P_i . (For an MUDC of A_c , the active nodes of each partition have the same polarity. Thus, a variable of c can be assigned true or false based on the polarity of the active nodes in its corresponding partition. Here, in the context of discussing a given UDC, we call a node active if it is in the UDC.)
- (ii) If $U \subseteq N_c$ is a UDC of A_c , then $H \cap U \neq \emptyset$. (At least one header node must be active for covering A_c .)
- (iii) $A_c \subseteq \text{disk}(\bigcup_{1 \leq i \leq n} P_i^{p_i})$, if $H \cap \bigcup_{1 \leq i \leq n} P_i^{p_i} \neq \emptyset$. Here $P_i^{p_i} \subset P_i$ contains either all positive polar nodes ($p_i = +$) or all negative polar nodes ($p_i = -$). That is, $P_i^{p_i} = P_i^+$ or P_i^- . (In other words, if the active nodes of each partition have the same polarity and one of them is a header node, then A_c is covered.)

Proof. The idea is to examine each possible case of partition P_i and ensure A_c will not be covered if, for each P_i , less than $|P_i|/2$ nodes are active or if exactly $|P_i|/2$ nodes but not having the same polarity are active. Furthermore, we need to examine whether A_c will be covered if none of header nodes is active. The complete proof is given in Appendix B. \square

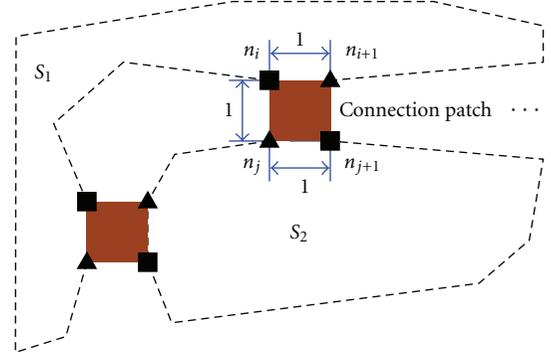


FIGURE 5: Connection patches for connecting two structures.

Note that, from Lemma 9(ii), at least one header node must be active for covering A_c . Thus, together with Lemma 9(i), an MUDC of A_c will allow each variable of a clause to be assigned *true* or *false*, based on the polarity of active nodes in its corresponding partition, for the clause being satisfied.

4.2. Composite Structures. Next, we illustrate how structures may be connected together to form a complex structure. As shown in Figure 5, two structures, $S_1 = (A_1, N_1)$ and $S_2 = (A_2, N_2)$, are connected via several 1×1 squares called *connection patches*. Each connection patch has two nodes from each structure, for example, $n_i, n_{i+1} \in N_1$ and $n_j, n_{j+1} \in N_2$, located at its vertices. (This is formally defined as precondition 25(i). For convenience sake, we use precondition 25(i) for referring to precondition (i) of Definition 25, and will use this labeling throughout this paper.) Besides, the nodes on the same edge of each connection patch have opposite polarities. (This is formally defined as preconditions 25(ii) and 26(ii).) For the sake of brevity, the formal definitions are given in Appendix C. We call the set $N_p = \{n_i, n_{i+1}\}$ a *port* of S_1 , and a port is a *connected port* if there is a connection patch attaching to it. Besides, the nodes from different structures but on the same edge of a connection patch are each other's *connection counterpart*, for example, n_i and n_j . Obviously, it is easy to derive the following lemma.

Lemma 10. A connection patch can be unit-disk covered by the same polar nodes located at its vertices, for example, $\{n_i, n_{j+1}\}$ or $\{n_{i+1}, n_j\}$ in Figure 5.

In this NP-completeness proof, in order to preserve the partially well-behaved property, we require two structures to be in such a way, that is, *least interactively connected*, that

- (1) at least one node from each connected port is active, (this is formally defined as precondition 27(i));
- (2) nonconnected port nodes do not cover any point, except the vertices, of the connection patches (this is formally defined as precondition 27(ii) which states whether a connection patch can be fully covered only depends on its connected port);

- (3) nonconnected port nodes of one structure do not cover any region of the other structure, (this is formally defined as precondition 28(i));
- (4) the connected ports of one structure cannot cover any point, except their connection counterparts, of the other structure (this is formally defined as precondition 28(ii));

The formal definitions about the least interactive connection are also given in Appendix C.

A variable structure can use any two nearby nodes on the side of border as a port, for example, $\{n_i, n_{i+1}\}$ shown in Figure 1. An edge structure uses its endpoints as ports, indicated by the arrows in Figure 2. For an n -way connector, each partition P_i contains a port indicated by the arrows in Figure 3. The fact that it is possible to make the above structures least interactively connected via the ports described is proved in Appendix D.

We can define the composite structure in the following definition and derive several lemmas about the least interactive connection. For the sake of brevity, the complete proofs of these lemmas are given in Appendix E.

Definition 11. The structures $S_1 = (A_1, N_1)$ and $S_2 = (A_2, N_2)$ are least interactively connected via the connection patches $A_{cp,1}, A_{cp,2}, \dots, A_{cp,T}$. One calls $S = (A_1 \cup A_2 \cup \bigcup_{1 \leq t \leq T} A_{cp,t}, N_1 \cup N_2)$ the *composite structure* of S_1 and S_2 . Furthermore, one denotes $S = S_1 + S_2$.

Lemma 12. Suppose $(A, N) = (A_1, N_1) + (A_2, N_2)$ and the cardinality of MUDCs of A_1 and A_2 is l_1 and l_2 , respectively. If $U \subseteq (N_1 \cup N_2)$ is a UDC of A and $|U| = l_1 + l_2$, then $U \cap N_1$ and $U \cap N_2$ are MUDCs of A_1 and A_2 , respectively.

Lemma 13 (Connection Lemma). Consider $S_1 = (A_1, N_1)$, $S_2 = (A_2, N_2)$, and $S = S_1 + S_2$ via one connection patch at the ports $N_{p_1} \subseteq N_1$ and $N_{p_2} \subseteq N_2$. (We note that this lemma may be extended to more than one connection patches with possible minor modification on Definition 28(ii); however we do not use this property and therefore ignore it.) Furthermore, for any $N'_1 \subseteq N_1$ with $N_{p_1} \subseteq N'_1$ and $N'_2 \subseteq N_2$ with $N_{p_2} \subseteq N'_2$, S will be partially well behaved on $N'_1 \cup N'_2$ if the following connection preconditions hold.

- (i) S_1 and S_2 are partially well behaved on N'_1 and N'_2 , respectively.
- (ii) There exist an MUDC of A_1 , $U'_1 \subseteq N_1$, and an MUDC of A_2 , $U'_2 \subseteq N_2$, such that $U'_1 \cap N'_1$ and $U'_2 \cap N'_2$ have the same polarity. (Since $U'_1 \cap N'_1$ and $U'_2 \cap N'_2$ have the same polarity, by Lemma 10, the connection patch is automatically covered without the help of the nodes not in U'_1 and U'_2 . Hence, $U'_1 \cup U'_2$ is an MUDC of $A_1 \cup A_2 \cup A_{cp}$.)

After describing the properties of the least interactive connection, we describe how these structures may be connected to encode G_B .

Figure 6 illustrates how edge structures are connected to a variable structure with connection patches. The edges may go up or down from the variable structure.

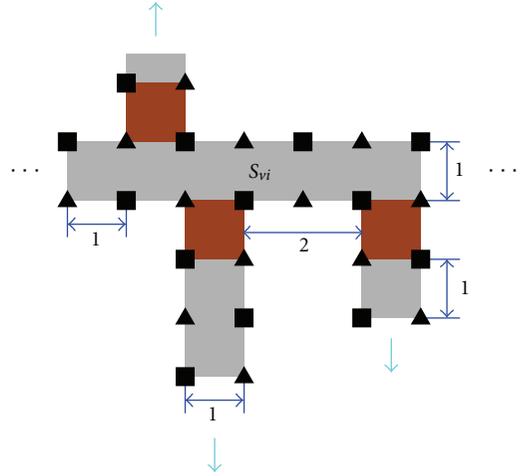


FIGURE 6: The structure represents a variable v_i with two edges going down and one edge going up. Note that the distance of two nearby ports on the same side must be at least 2 to ensure the least interactive connection.

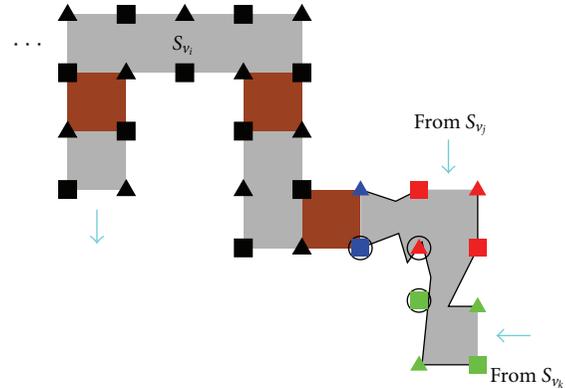


FIGURE 7: The variable structure S_{v_i} is connected to the positive partition of the 3-way connector, which represents the clause containing two positive literals, v_i and v_k , and one negative literal, v_j . Note that the 3-way connector has two positive partitions and one negative partition. The header nodes are indicated by squares or triangles surrounded by circles.

With n -way connectors described above, we can transform an edge and its ends in G_B by connecting a variable structure to a partition of an n -way connector via an edge structure. For example, suppose a clause is $(v_i + v_j + v_k)$, then we need a 3-way connector with two positive partitions and one negative partition. As shown in Figure 7, we connect the edge from S_{v_i} to one positive partition, the edge from S_{v_j} to the negative partition, and the edge from S_{v_k} to the other positive partition.

Since all structures are connected via 1×1 squares, the positions of structures must match, that is, all structures can be placed in a 2D space such that their ports are on a 2D grid with resolution 1×1 . Obviously, such a placement is possible for variable and edge structures. Furthermore, as indicated in

Tables 1 and 2, the relative positions of the ports of each n -way connector are integers, so it is also possible to make such a placement for n -way connectors.

Note that, in addition to positions, polarities of ports must also match. (Precondition 26(ii) needs to be satisfied.) However, for example, it is possible that the polarities may not match when an edge structure is connected to a connector as shown in Figure 8(a).

Therefore, referring to Figure 9, we introduce a structure called a *polarity inverter* and denoted as $S_p = (A_p, N_p)$ to invert the polarity. S_p contains a main structure, 9×1 rectangles, and two buffers, 1×1 squares. (The purpose of the buffers is to ensure S_e satisfies the requirement that nonconnected port nodes do not cover any point, except the vertices, of the connection patches, that is, precondition 27(ii).) Similar to edge structures, a polarity inverter use its endpoint pairs, that is, $\{n_i, n_{i+1}\}$ and $\{n_j, n_{j+1}\}$ in Figure 9, as ports.

The distance between two nearby column-pairs of the main structure is $9/10$. Thus, the polarities are inverted compared with a normal edge of the same length. With the polarities inverted, the polarity requirements for vertices of a connection patch can be satisfied as shown in Figure 8(b).

Of course, the structure $S_p = (A_p, N_p)$ also satisfies the following lemma. The proof is similar to variable structures and is given in Appendix F.

Lemma 14. *The structure $S_p = (A_p, N_p)$ shown in Figure 9 is well aligned and well behaved.*

Definition 15. For a given variable structure S_v , an n -way connector S_c is called an *associated connector* of S_v if S_c is connected to S_v via an edge structure. Furthermore, the partition P_i of S_c where S_v is connected to is called an *associated partition* of S_v . Similarly, the edges, polarity inverters, and connection patches used to connect S_v and S_c are called *associated connection edges*, *associated polarity inverters*, and *associated connection patches* of S_v , respectively.

For the i th variable, v_i , let

NV_i be the set of nodes in the corresponding variable structures S_{v_i} ,

NE_i be the set of nodes in all the associated edge structures of S_{v_i} ,

NI_i be the set of nodes in all the associated polarity inverters of S_{v_i} ,

NC_i be the set of nodes in all the associated n -way connectors of S_{v_i} ,

NP_i be the set of nodes in all the associated partitions of S_{v_i} ,

AV_i be the shaded region of S_{v_i} ,

AE_i be the union of the shaded region from all the associated edge structures of S_{v_i} ,

AI_i be the union of the shaded region from all the associated polarity inverters of S_{v_i} ,

AC_i be the union of the shaded region from all the associated n -way connectors of S_{v_i} ,

ACP_i be the union of the shaded region from all the associated connection patches of S_{v_i} .

Definition 16. Let $N_i = NV_i \cup NE_i \cup NI_i \cup NC_i$, $A_i = AV_i \cup AE_i \cup AI_i \cup AC_i \cup ACP_i$, and $T_i = (A_i, N_i)$. We call the composite structure T_i the *territory* of the i th variable. Furthermore, let $N'_i = NV_i \cup NE_i \cup NI_i \cup NP_i$. The nodes in N'_i are the *pieces* of the i th variable. Note that $NP_i \cap NP_j = \emptyset$ if $i \neq j$, and thus $N'_i \cap N'_j = \emptyset$ if $i \neq j$.

Note that T_i represents the i th variable and all clauses it belongs to as shown in Figure 10. It is not difficult to layout each structure on the plane and make variable structures and connectors far enough to prevent unwanted interactions between nearby structures, that is, structures are least interactively connected. Consequently, we have the following lemma which states that, for a given variable, its territory is partially well behaved on the set of its pieces.

Lemma 17. *If all the associated structures of the variable structure S_{v_i} are least interactively connected, the territory T_i is partially well-behaved on N'_i .*

Proof. Since structures are least interactively connected, this lemma can be proved by Lemmas 8, 9(i), 14, and Connection Lemma. The fact that $2 \leq |c| \leq 3$ for each clause c is the key to ensure precondition (ii) of Connection Lemma is satisfied for Connection Lemma being applicable. The complete proof is given in Appendix G.

After introducing the structures and their properties, we can define an equivalent MUDC problem with the geometry, $MUDC(B)$, for a given boolean formula, B , in P3SAT by replacing the variables, clauses, and edges of the bipartite graph G_B with their corresponding structures. Denote that

NV is the set of nodes in all the variable structures,

NE is the set of nodes in all the edge structures,

NI is the set of nodes in all the polarity inverters,

NC is the set of nodes in all the n -way connectors,

AV is the union of the shaded region from all the variable structures,

AE is the union of the shaded region from all the edge structures,

AI is the union of the shaded region from all the polarity inverters,

AC is the union of the shaded region from all the n -way connectors,

ACP is the union of the required connection patches.

Let $MUDC(B) = (A_B, N_B)$ with $A_B = AV \cup AE \cup AI \cup AC \cup ACP$ and $N_B = NV \cup NE \cup NI \cup NC$, and $K = |N_B|/2$. Hence, we have the following claim. Note that it is not difficult to prove that the construction from B to $MUDC(B)$ can be done in polynomial time. \square

Claim B is satisfiable if and only if $MUDC(B)$ has a UDC with cardinality K .

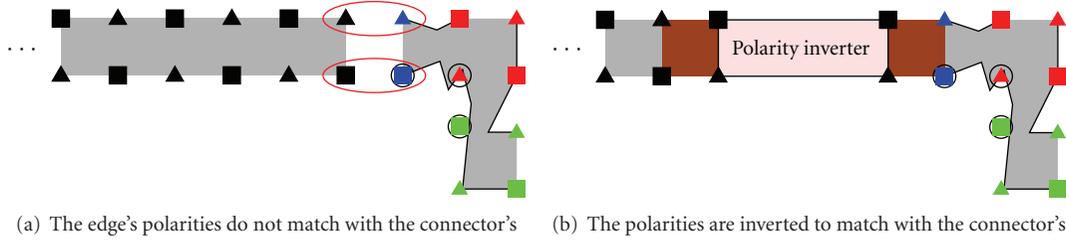


FIGURE 8: A polarity inverter structure may be needed to invert the polarity of the edge to connect the edge and 3-way connector.

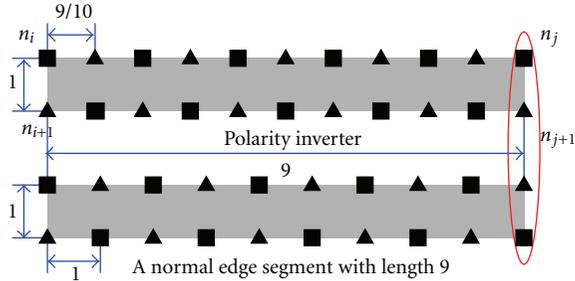


FIGURE 9: A structure for inverting the polarity of the edge. After distance 11, the polarities are inverted compared with a normal edge.

Note that the key to the backward direction of the proof is Lemma 17; that is, for a given variable, its territory is partially well behaved on the set of its pieces. Lemma 17 will be used to derive, that if $MUDC(B)$ has a UDC with cardinality K , then, for each variable, half of its pieces with same polarity needs to be active to unit-disk cover its territory. Therefore, each variable could be assigned *true* or *false* based on the polarity of its active pieces.

4.3. Proof of the Claim

Proof. \Rightarrow For each N'_i , choose the nodes with the polarity which is the same as the assignment of the i th variable in a given satisfying instance of B . Obviously, only $|N|/2 = K$ nodes are picked. By Lemmas 8, 14, and 10, AV , AE , AI , and ACP are covered. Furthermore, since B is satisfiable, at least one header node of each n -way connector is active. By Lemma 9(iii), AC is covered. Thus $MUDC(B)$ has a UDC with cardinality K .

\Leftarrow Let $MUDC(B)$ have a UDC $U \subseteq N$ with $|U| = K$. We will show that this set must look right.

From Lemmas 8, 9(i), and 14, we know that the cardinality of an MUDC of a variable structure, edge structure, n -way connector, or polarity inverter is half the number of the nodes in the structure. Since all structures in $MUDC(B)$ are least interactively connected and $|U| = |N|/2$, it is not difficult to derive that, for the i th variable structure, $U \cap N_i$ is an MUDC of T_i by removing nonassociated edge structures of T_i one by one and Lemma 12.

From Lemma 17 and $N'_i \subseteq N_i$, $U \cap N'_i$ only contains the nodes with the same polarity. Thus, the i th variable could be

assigned *true* or *false* based on the polarity of the nodes in $U \cap N'_i$. Finally, since at least one header node of each n -way connector must be active from Lemma 9(ii), the corresponding clause will be *true*. \square

5. Extensions of MUDC

MUDC can be easily extended to the following two more general cover problems, which require each location to be unit-disk covered by predefined number of nodes. These problems regard the quality of various services of sensor network applications such as surveillance, object tracking, and fault tolerance.

Problem 18 (Minimum Unit-Disk k -Cover, MUDKC). Given a geometry (A, N) and two positive integers k and K , determine whether there is a subset $U \subseteq N$ with $|U| \leq K$ such that for all $x \in A$, $|\{u \in U \mid x \in \text{disk}(u)\}| \geq k$; that is, x is unit-disk covered by at least k nodes in U .

Problem 19 (Minimum Unit-Disk Multicover, MUDM). Given a geometry (A, N) , a quality of surveillance function $q : A \rightarrow \mathbb{Z}^+$, and a positive integer K , determine whether there is a subset $U \subseteq N$ with $|U| \leq K$ such that for all $x \in A$, $|\{u \in U \mid x \in \text{disk}(u)\}| \geq q(x)$; that is, x is unit-disk covered by at least $q(x)$ nodes in U .

We may also consider connectivity and have the following problem.

Problem 20 (Minimum Connected Unit-Disk Cover, MCUDC). Given a geometry (A, N) , a positive number $R_c \in \mathbb{R}^+$, and a positive integer K , determine whether there is a subset $U \subseteq N$ with $|U| \leq K$ such that $A \subseteq \text{disk}(U)$ and the graph $G_c = \{U, E_c\}$ is connected. Here $E_c = \{(n, n') \mid d(n, n') \leq R_c\}$.

Furthermore, under many environmental data sampling applications, instead of full coverage, a predefined percentage of coverage is required for achieving energy efficiency and preciseness of sampling. The objective of the following problem is to find as few nodes as possible to achieve the coverage requirements.

Problem 21 (Minimum Unit-Disk Partial Cover, MUDPC). Given a geometry (A, N) , a positive number r with $0 \leq r \leq 1$, and a positive integer K , determine whether there is a subset

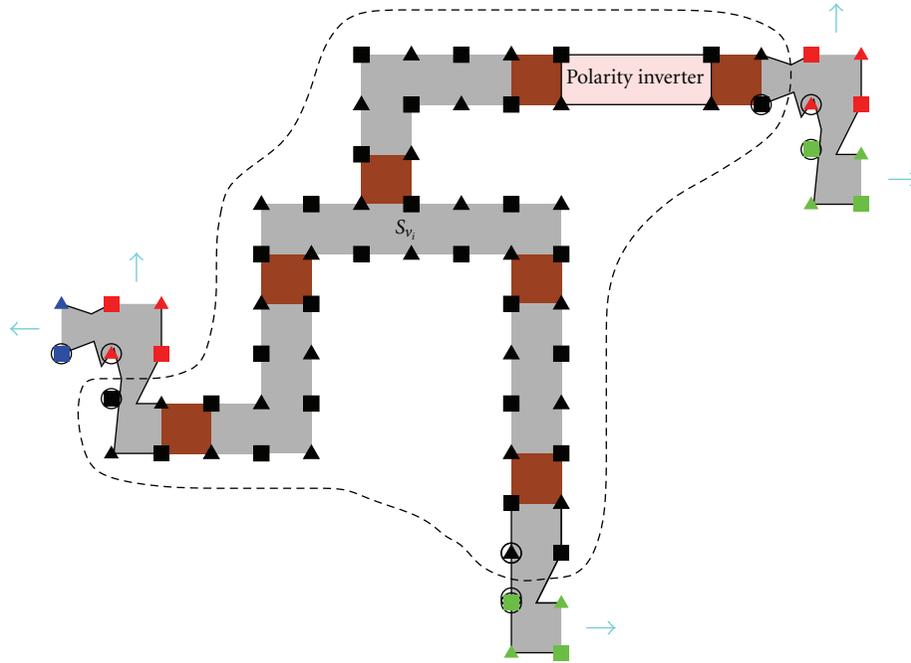


FIGURE 10: The territory, T_i , of the variable v_i includes variable structure S_{v_i} and all associated edge structures, patches, inverters, and connectors. The nodes enclosed in dashed line are pieces of v_i .

$U \subseteq N$ with $|U| \leq K$ such that $(\text{area}(\text{disk}(U)))/\text{area}(A) \geq r$. Here the function $\text{area}(\cdot)$ gives the area of a given region.

By the NP-completeness of MUDC, it is easy to derive the complexity of the above problems.

Corollary 22. *MUDKC, MUDM, MCUDC, and MUDPC are NP-complete.*

Proof. Note that every instance of MUDC can be viewed as an instance of MUDKC, MUDM, or MUDPC simply by letting $k = 1$, $q(x) = 1$ for all $x \in A$, or $r = 1$, respectively. Thus MUDC is just a restricted version of these problems, and their NP-completeness follows by trivial transformations from MUDC.

For the geometry, MUDC(B), described in the NP-completeness proof of MUDC, it is obvious that, for any UDC of A_B , the distance between an active node and its closest active node is less than 2. Thus, it is not difficult to prove that if $R_c \geq 2$, G_c is connected. That is, MUDC is just a restricted version of MCUDC with $R_c \geq 2$, and the NP-completeness of MCUDC follows by a trivial transformation from MUDC. \square

6. Arc Sampling Algorithm for Reducing MUDC to MSC

As stated earlier, MUDC may be solved by partitioning the region A into disjoint sectors [20]. Consequently, MUDC is reduced to MSC and many well-known algorithms can be applied, for example, the greedy algorithm is the best approximation algorithm and the approximation factor is well known.

To identify necessary sectors is a key factor to whether the solutions found by the algorithms for the transformed MSC are valid, that is, the solutions are disk covers of the original MUDC. A naive approach for partitioning is to sample A at uniform spacings in a grid pattern; then the sampling points covered by the same set of nodes would be grouped into one sector. With enough resolution, all necessary sectors can be successfully identified at the expense of computation time.

However, to determine a good resolution may be difficult. For example, Figure 11 shows that inappropriately increasing resolution may not necessarily find a valid solution, and Figure 13(a) illustrates that the ratio of successfully finding a valid solution decreases as the node density decreases. Therefore, we propose an arc sampling approach which is inspired by the theorem of the paper [32], that is, A is covered if and only if the perimeter of each node's sensing region is covered. (Several special cases including boundary are also discussed in [32].)

Consider the node n with its neighbors, that is, the nodes with distance not greater than 2 from n . As illustrated in Figure 12, n 's perimeter is divided into disjoint arcs by its neighbors' perimeters and the boundary of A . It is obvious that all points of each disjoint arc in A are covered by the same set of nodes. Thus, we can simply choose a point such as the midpoint from each arc in A , for example, χ_2 from arc $\widehat{\alpha_2\alpha_3}$, as a sampling point. Note that if n 's perimeter cannot be divided, n 's perimeter (and thus A) is not covered [32], as indicated in Lines 7~9 of Algorithm 1. From the earlier mentioned theorem of the paper [32], it is easy to derive that A is covered if and only if all these sampling points are covered. Thus, the solutions of the MSC transformed by this arc sampling approach are always valid.

The arc sampling algorithm is shown in Algorithm 1. Here $P(n) = \{x \mid d(n, x) = 1\}$ is the perimeter of n . The outer loop between Lines 2 and 15 will run $|N|$ times. The average time complexity for a node to find all its perimeter intersections with neighbors, that is, Lines 4~6, is $O(4\pi d)$. Here $d = |N|/\text{area}(A)$ is the density of nodes. Note that each node has average $4\pi d$ neighbors and thus $8\pi d$ disjoint arcs. Hence, the sorting in Line 10 could be implemented in $O(8\pi d \cdot \log 8\pi d) = O(d \log d)$ time. In addition, the time complexity of finding the midpoints, that is, Lines 11~14, is $O(8\pi d)$. Thus the overall time complexity of Algorithm 1 is $O(|N|d \log d)$ or $O(\text{area}(A) \cdot d^2 \log d)$. On the other hand, it is not difficult to derive that the time complexity of finding all sampling points is $O(\text{area}(A)/a)$ for the grid sampling approach. Here a is the area of each grid. Together with Figure 13(a), we may conclude that the arc sampling approach will perform more efficiently than the grid sampling approach, particularly with low density of nodes.

We conducted an experiment to compare the grid sampling and the arc sampling approaches. In the experiment, there are 240 nodes deployed uniformly in a square region ranging from 30×30 to 75×75 . The radius of the sensing range is 10. The sampling interval of the grid sampling approach is 0.1. After transforming to MSC, the greedy algorithm is used to find the approximated solution. Each result is the average of 100 random deployments.

Figure 13(b) shows the effectiveness of the arc sampling approach. The solutions from both approaches have almost same sizes, that is, same number of nodes. However, Figure 13(a) shows that not every solution obtained from the grid sampling approach is valid. Figure 13(c) illustrates that the arc sampling approach requires less computation time to reduce MUDC to MSC than the grid sampling approach except the densest deployment. Thus, the arc sampling approach is effective and efficient for reducing MUDC to MSC.

7. Decentralized Polynomial Approximation Algorithms

Algorithm 1 and the greedy algorithm may not be suitable for all practical sensor network applications, since it is a centralized algorithm at the cost of potentially excessive communication across the whole network and communication accounts for the majority of energy consumption. The communication power consumption increases with number of nodes and internode distances, so it is not well scalable. Unless the nodes involving in the communication and computation have enough resources, the algorithm may not complete successfully.

Thus, we present a decentralized algorithm in which nodes only require local information by using the *divide and conquer* technique described in [42] and derive its approximation factor. Furthermore, if the maximum node density is fixed, we may design a constant approximation factor algorithm by using the similar technique. (Note that MUDC remains NP-complete even with fixed maximum node density. It could be easily proved from the fact that the density of

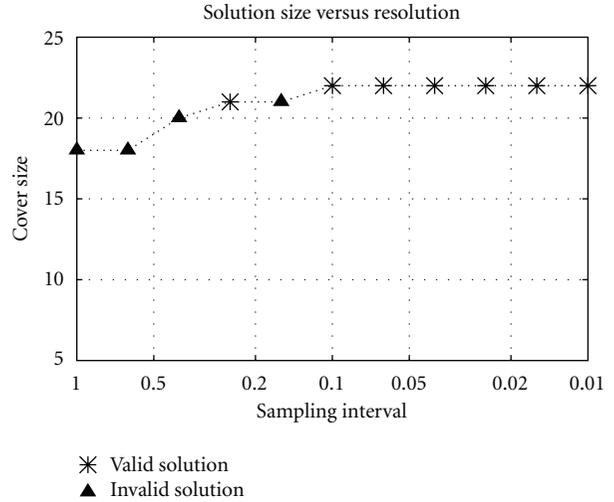


FIGURE 11: The solution found at different sampling intervals. Increasing resolution may not necessarily find a valid solution.

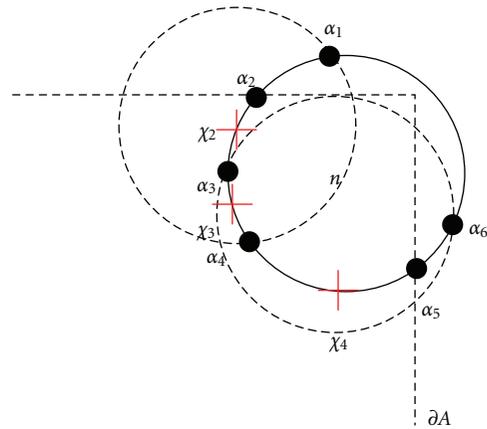
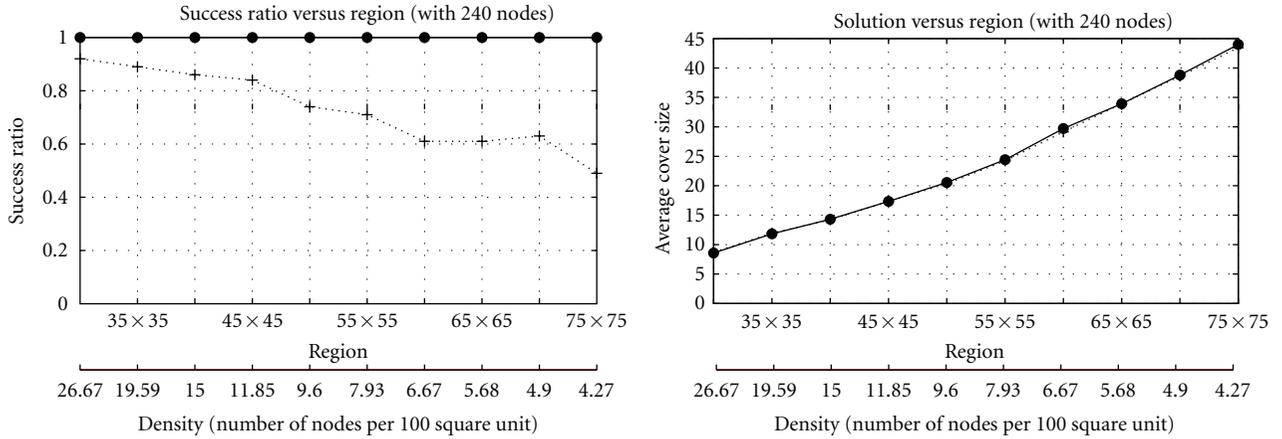


FIGURE 12: Each arc in A , that is, $\widehat{\alpha_2\alpha_3}$, $\widehat{\alpha_3\alpha_4}$, or $\widehat{\alpha_4\alpha_5}$, is covered by the same set of nodes. Hence, we can simply use a point, for example, the midpoint marked as a cross, from each arc as a sampling point. Here ∂A represents the boundary of A .

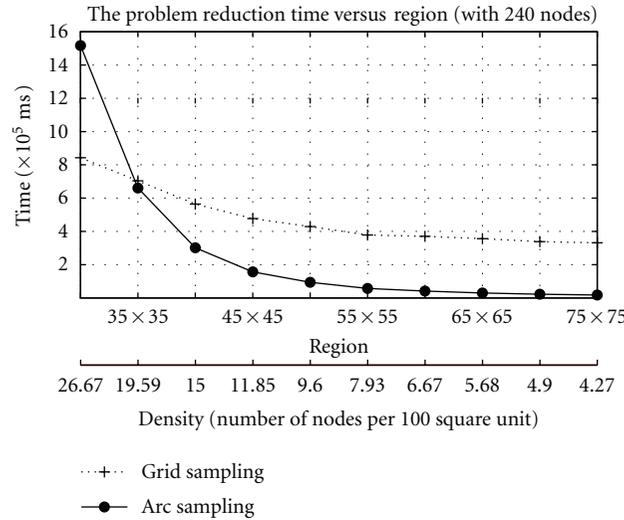
MUDC(B) in the NP-completeness proof of MUDC is bounded by a constant.)

7.1. *Decentralized Greedy Approximation Algorithm.* The proposed algorithm for the instance (A, N) proceeds as follows.

- (1) Use Algorithm 1 to determine sampling points.
- (2) Divide A into vertical strips with width being the diameter of the sensing region, that is, 2. Each strip is left closed and right open. Number strips from left to right. There are total I strips.
- (3) Divide each strip into cells with length being the diameter of the sensing region. Each cell is bottom closed and top open. Number cells from bottom to top. Denote the j th cell of the i th strip as $C_{i,j}$. There are total J cells for each strip.



(a) The ratio of successfully finding a solution with different node density: the greedy algorithm with the arc sampling approach always generates a valid solution (b) The arc sampling approach is as effective as the grid sampling approach



(c) The arc sampling approach requires less computation time to reduce MUDC to MSC than the grid sampling approach except the densest deployment

FIGURE 13: The experiment results show that the arc sampling approach is effective and efficient for reducing MUDC to MSC.

(4) Apply the greedy algorithm to each cell, that is, select a node that covers the maximum number of uncovered sampling points in the cell. Denote the solution of $C_{i,j}$ as $SOL_{C_{i,j}}$.

(5) Output the solution $SOL_A = \bigcup_{\substack{1 \leq i \leq I \\ 1 \leq j \leq J}} SOL_{C_{i,j}}$.

Figure 14(a) illustrates that A is divided into 2×2 cells. Note that each sampling point is located in exactly one cell. This algorithm requires that geometric information of A and cells are known a priori to each node and each node's location can be determined after deployment.

For each cell $C_{i,j}$, we define its *repository* $p_{i,j} = \{x \mid \exists y \in C_{i,j}, d(x, y) \leq 1\}$. As illustrated in Figure 14(b), $p_{i,j}$ is the region containing all nodes that may cover the sampling points in $C_{i,j}$. Hence, in Step 4, the greedy algorithm is applied to the nodes in each cell's repository and can be implemented in

$O(n_{rp}^2 \log n_{rp})$, where n_{rp} is the maximum number of nodes in a repository. Furthermore, Figure 14(b) also illustrates that a node does not need to communicate with others further than $(2 + 2\sqrt{2})$ times of the sensing radius. Thus, this approach is more scalable than the centralized greedy algorithm.

Theorem 23. *The above algorithm has an approximation factor $4O(\log m)$. (Though $4O(\log m)$ can be written as $O(\log m)$ by definition, we explicitly write it out to emphasize the approximation factor of the decentralized algorithm is four times the approximation factor of the centralized algorithm.) Here m is the maximum number of sampling points covered by a single node.*

Proof. The theorem is the result of *the shifting lemma* in [42]. The proof proceeds as follows.

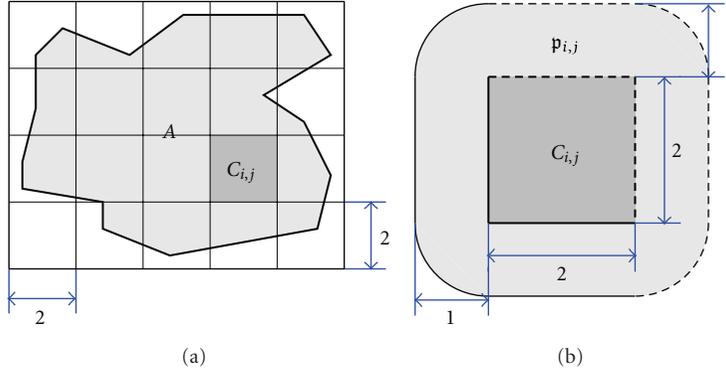


FIGURE 14: Divide and conquer: (a) The monitored region A (enclosed in heavy line) is divided into 2×2 cells. (b) $C_{i,j}$ (enclosed in heavy line) and its repository $p_{i,j}$ (enclosed in thin line)—the solid lines and dashed lines represent closed and open boundaries. Note that the distance between any two points in $p_{i,j}$ does not exceed $(2 + 2\sqrt{2})$.

For $A' \subseteq A$, denote that $\text{OPT}_{A'}$ is the optimum solution to cover the sampling points in A' . That is, $\text{OPT}_{C_{i,j}}$ is the optimum solution for $C_{i,j}$, $\text{OPT}_{\cup_j C_{i,j}}$ is the optimum solution for the i th strip, and so forth. Thus, from Step 4, $|\text{SOL}_{C_{i,j}}| \leq O(\log m_{i,j}) \cdot |\text{OPT}_{C_{i,j}}|$. Here $m_{i,j}$ is the maximum number of sampling points in $C_{i,j}$ covered by a single node.

Consider the i th strip, and define the following disjoint subsets of $\text{OPT}_{\cup_j C_{i,j}}$:

$\text{OPT}^{(j)}$ be the set of nodes that only cover the sampling points in $C_{i,j}$,

$\text{OPT}^{(j,j+1)}$ be the set of nodes that cover both the sampling points in $C_{i,j}$ and $C_{i,j+1}$.

Note that since the length of cells is the diameter of the sensing region, the union of the above disjoint subsets is $\text{OPT}_{\cup_j C_{i,j}}$. Hence, $|\text{OPT}_{\cup_j C_{i,j}}| = \sum_{1 \leq j \leq J} |\text{OPT}^{(j)}| + \sum_{1 \leq j \leq J-1} |\text{OPT}^{(j,j+1)}|$. Besides, it is obvious that $\text{OPT}^{(j-1,j)} \cup \text{OPT}^{(j)} \cup \text{OPT}^{(j,j+1)}$ covers all sampling points in $C_{i,j}$. (Here $\text{OPT}^{(0,1)} = \emptyset$ and $\text{OPT}^{(J,J+1)} = \emptyset$.) Thus, $|\text{OPT}_{C_{i,j}}| \leq |\text{OPT}^{(j-1,j)} \cup \text{OPT}^{(j)} \cup \text{OPT}^{(j,j+1)}| = |\text{OPT}^{(j-1,j)}| + |\text{OPT}^{(j)}| + |\text{OPT}^{(j,j+1)}|$. Therefore, it can easily be derived that

$$\begin{aligned} \sum_{1 \leq j \leq J} |\text{OPT}_{C_{i,j}}| &\leq |\text{OPT}_{\cup_j C_{i,j}}| \\ &+ \sum_{1 \leq j \leq J-1} |\text{OPT}^{(j,j+1)}| \\ &\leq 2 \cdot |\text{OPT}_{\cup_j C_{i,j}}|. \end{aligned} \quad (1)$$

□

Similarly, it can easily be derived that

$$\sum_{1 \leq i \leq I} |\text{OPT}_{\cup_j C_{i,j}}| \leq 2 \cdot |\text{OPT}_A|, \quad (2)$$

and then

$$\sum_{\substack{1 \leq i \leq I \\ 1 \leq j \leq J}} |\text{OPT}_{C_{i,j}}| \leq 4 \cdot |\text{OPT}_A|. \quad (3)$$

Consequently,

$$\begin{aligned} |\text{SOL}_A| &\leq \sum_{\substack{1 \leq i \leq I \\ 1 \leq j \leq J}} |\text{SOL}_{C_{i,j}}| \\ &\leq \sum_{\substack{1 \leq i \leq I \\ 1 \leq j \leq J}} O(\log m_{i,j}) |\text{OPT}_{C_{i,j}}| \\ &\leq O(\log m) \cdot \sum_{\substack{1 \leq i \leq I \\ 1 \leq j \leq J}} |\text{OPT}_{C_{i,j}}| \\ &\leq 4O(\log m) \cdot |\text{OPT}_A|. \end{aligned} \quad (4)$$

Note that, obviously, $\max_{i,j} \{m_{i,j}\} \leq m$.

7.2. Constant Approximation Factor Algorithm with Fixed Maximum Density. When the maximum node density, denoted as d , is fixed, the similar divide and conquer technique can be used to derive a constant approximation factor algorithm.

The algorithm is almost the same as the previous one except Step 4, which will be modified as follows:

(4) Apply an exhaustive search for optimum solution to each cell. Denote the solution of $C_{i,j}$ as $\text{SOL}_{C_{i,j}}$.

Theorem 24. *The above algorithm has a constant approximation factor 4.*

Proof. Note that the number of nodes in each cell's repository is at most $\lceil \text{area}(p_{i,j}) \cdot d \rceil = \lceil (12 + \pi)d \rceil$. (Refer to Figure 14(b); the area of each repository is $(12 + \pi)$.) Thus, the time complexity of the exhaustive search is at most $2^{\lceil (12 + \pi)d \rceil}$ for each cell. Since d is fixed, an optimum solution for each cell can be found with a constant time complexity. Since $|\text{SOL}_{C_{i,j}}| = |\text{OPT}_{C_{i,j}}|$, from (3), we have

$$|\text{SOL}_A| \leq \sum_{\substack{1 \leq i \leq I \\ 1 \leq j \leq J}} |\text{SOL}_{C_{i,j}}| \leq 4 \cdot |\text{OPT}_A|. \quad (5)$$

□

7.3. Performance Evaluation. We conducted various simulations to evaluate the proposed algorithms. In Figure 15,

```

Input:  $(A, N)$ 
Output: a set  $\Xi$  of sampling points
(1)  $\Xi \leftarrow \emptyset$ 
(2) for all  $n_i \in N$  do
(3)    $\Gamma \leftarrow P(n_i) \cap \partial A$ 
(4)   for all  $n_j \in N$  such that  $d(n_i, n_j) \leq 2$  do
(5)      $\Gamma \leftarrow \Gamma \cup (P(n_i) \cap P(n_j))$ 
(6)   end for
(7)   if  $\Gamma = \emptyset$  then
(8)     exit/*  $A$  is not covered by  $N$ . */
(9)   end if
(10)  List  $L$   $\leftarrow$  the points in  $\Gamma$  sorted by their azimuth
      angles on the polar coordinate system
      with reference to  $n_i$ 
(11)  for all  $\alpha_i \in L$  such that  $1 \leq i < |L|$  do
(12)     $\Xi \leftarrow \Xi \cup \{\chi \mid \chi \text{ is the midpoint of } \widehat{\alpha_i \alpha_{i+1}} \text{ and } \chi \in A\}$ 
(13)  end for
(14)   $\Xi \leftarrow \Xi \cup \{\chi \mid \chi \text{ is the midpoint of } \widehat{\alpha_{|L|} \alpha_1} \text{ and } \chi \in A\}$ 
(15)end for

```

ALGORITHM 1: Arc sampling algorithm.

nodes are deployed uniformly within a 30×30 square region. Figures 15(a) and 15(b) show the solution size and execution time for various sensing ranges in which there are 25 nodes. The optimum solution OPT is found by exhaustive searching. GRD denotes the solution by using Algorithm 1 and the greedy algorithm. deGRD and deOPT represent the algorithms described in Sections 7.1 and 7.2 respectively. The cover size decreases as the sensing radius increases, since each node can cover a larger region. GRD and deOPT have similar performance in terms of cover size and execution time. Furthermore, deGRD generates the largest cover size, on average 48% more than OPT, 22.7% more than GRD, or 21% more than deOPT, but requires the least execution time, on average 0.01% of OPT, 22.7% of GRD, or 17.8% of deOPT.

Figures 15(c) and 15(d) indicate the solution size and execution time for various number of nodes in which the sensing radius is fixed at 10. The cover size does not change significantly as the number of nodes increases, since the sensing region of each node does not change. GRD and deOPT have similar cover size, but deOPT requires more execution time than GRD. Similarly, deGRD generates the largest cover size, on average 32% more than OPT, 17% more than GRD, or 14% more than deOPT, in the least time, on average 0.017% of OPT, 16.7% of GRD, or 9.5% of deOPT.

We also considered the scenario in which nodes are deployed in a Gaussian distribution with the peak located at the center of A and the variance 15, and the results are illustrated in Figure 16. Here A is a 30×30 square region. In Figures 16(a) and 16(b), the number of nodes is 25 and the sensing radius varies between 8.5 and 12. deGRD generates the largest cover size, on average 50% more than OPT, 18.7% more than GRD, or 26% more than deOPT, but requires the least execution time, on average 0.019% of OPT, 21.7% of GRD, or 10% of deOPT.

Furthermore, in Figures 16(c) and 16(d), the sensing radius is fixed at 10 and the number of nodes varies between 12 and 30. Similarly, deGRD generates the largest cover size, on average 40% more than OPT, 13.1% more than GRD, or 20.6% more than deOPT, in the least time, on average 0.03% of OPT, 20.8% of GRD, or 8.2% of deOPT. For most of cases, deOPT has smaller cover sizes than GRD in this scenario.

8. Conclusion

In this paper, we consider the complexity of MUDC, the Minimum Unit-Disk Cover problem. This problem has applications in extending the sensor network lifetime by selecting minimum number of nodes to fully cover a geometric connected region of interest and putting the remaining nodes in power saving mode. MUDC is a restricted version of MSC where the sensing region of each node is a unit-disk and the monitored region is geometric connected, a well-adopted network model in many works of the literature.

To prove the hardness of MUDC, we construct various structures to represent variables and edges of a given P3SAT instance's bipartite graph G_B . With the well-aligned and partially well-behaved properties of these structures, we illustrate that the structures can be unit-disk covered with half of nodes. Furthermore, we introduce the n -way connectors to represent clauses, which can be unit-disk covered with half of its nodes if and only if the corresponding clauses have *true* assignments. Finally, we discuss how complex structures can be constructed by connecting simpler structures while still preserving these properties, that is, via the least interactive connection. Thus, we prove that P3SAT can be directly reduced to MUDC in polynomial time, and obtain the NP-completeness proof of MUDC.

We also discuss several optimum problems with various coverage constraints introduced by different sensing applications. These problems are extensions of MUDC, and their NP-completeness proofs are presented as a corollary.

We propose the arc sampling algorithm which may effectively and efficiently reduce MUDC to MSC, and many well-known algorithms can be applied to find approximated solutions. We also propose a decentralized algorithm with a guaranteed performance. The algorithm requires only local communication, that is, a node does not need to communicate with others further than $(2+2\sqrt{2})$ times of the sensing radius. Thus, this approach is scalable. Furthermore, we present an algorithm with a constant approximation factor 4 if the maximum node density is fixed. Finally, we provide simulation results to evaluate the proposed algorithms and the optimum algorithm in uniform and Gaussian deployment networks. The results show that deOPT may have smaller cover size than GRD at the cost of more execution time. In addition, deGRD generates the largest cover size in the least time.

Appendices

A. Proof of Lemma 8

In this appendix, we present the proof for Lemma 8.

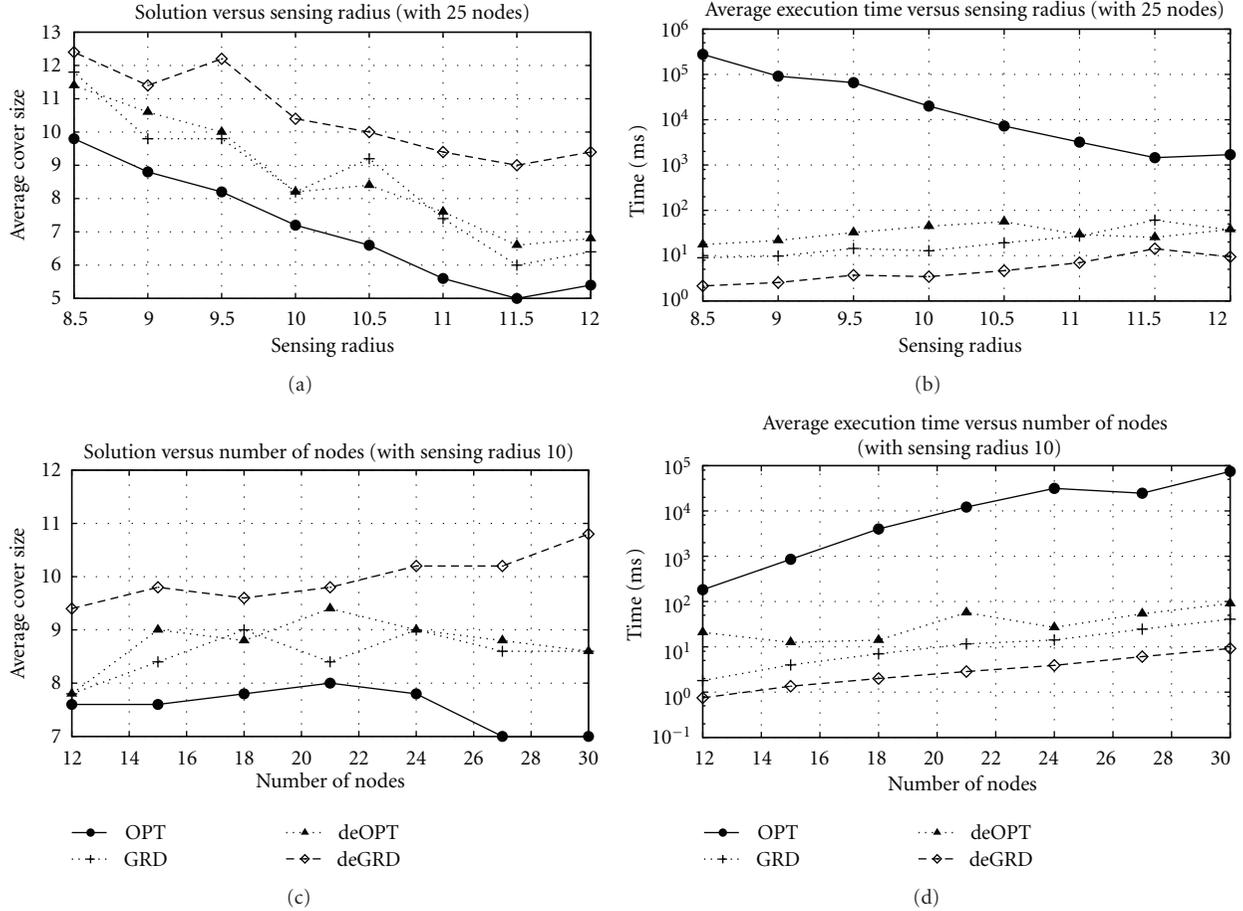


FIGURE 15: Performance evaluation: the nodes are deployed uniformly within a 30×30 square region.

A.1. Variable Structures. Obviously, the structure shown in Figure 1 has the same number of positive and negative polar nodes and, thus, precondition 7(i) is satisfied.

We call the pair of opposite polar nodes in the i th column the *column-pair* i . From Figure 17, it is not difficult to prove that each column-pair must have at least one active node to fully cover A_v . Since there are $d_v + 1$ columns, $|U| \geq d_v + 1 = |N_v|/2$ for a UDC U of A_c ; that is, the precondition 7(ii) is satisfied. Besides, if we can pick exactly one node from each column-pair and the resulting set, say U' , can-unit-disk cover A_v , then U' is an MUDC since $|U'| = d_v + 1$.

It is easy to prove that, if the picked nodes from each column-pair do not have the same polarity, A_v cannot be unit-disk covered by these $d_v + 1$ picked nodes. Suppose that the picked nodes of the i th column and the $(i+1)$ th column have opposite polarities. From the Figure 18, A_v cannot be covered.

Thus, the only possibility to cover A_v with $d_v + 1$ nodes is to pick the nodes with the same polarity from each column-pair, which can be easily proved by induction. Figures 19(a) and 19(b) show the base cases of the induction and Figure 19(c) illustrates the induction step. Therefore, precondition 7(iii) is satisfied and S is well behaved. Note that the induction step works for both positive and negative cases and also proves that S is well aligned.

A.2. Edge Structures. As illustrated in Figure 20, the structure $S_e = (A_e, N_e)$ is basically a composite structure from numbers of variable structures connected via connection patches. By Lemma 29 described in Appendix D, the variable structures are least interactively connectable at any two nearby nodes on the side of border. Thus, this lemma can be proved by induction on the composing variable structures with Lemma 10 and Connection Lemma.

B. Proof of Lemma 9

In this appendix, we complete the proof of Lemma 9.

B.1. 2-Way Connectors. Figures 3(a) and 4(a) illustrate the labels of nodes and vertices of A_c for the 2-way connector. Table 1 lists the positions of nodes and vertices relative to n_1 . Note that $\mathcal{P} = \{P_1, P_2\}$, $P_1 = \{n_4, n_5, n_6, n_7\}$, $P_2 = \{n_1, n_2, n_3, n_8\}$, and the header nodes $h_1 = n_7$ and $h_2 = n_8$. Obviously, $|P_1^+| = |P_1^-|$ for $i = 1$ and 2, and, thus, precondition 7(i) is satisfied.

As shown in Figure 21(d), only two active nodes n_4 and n_7 from P_1 cannot unit-disk cover A_c even all nodes in P_2 are active, which implies that only one node, n_4 or n_7 , from P_1 being active cannot unit-disk cover A_c . Similarly from

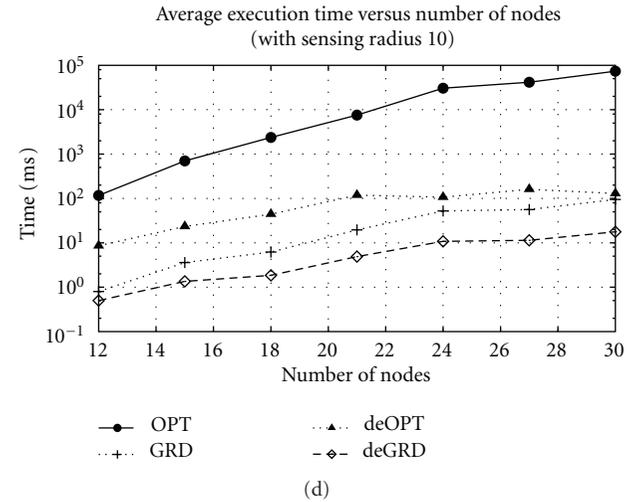
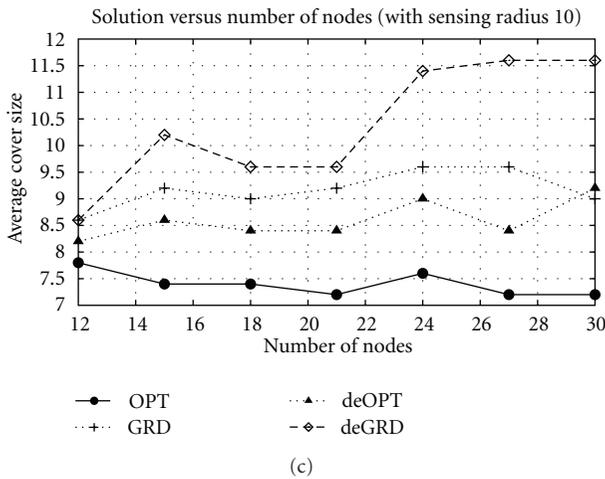
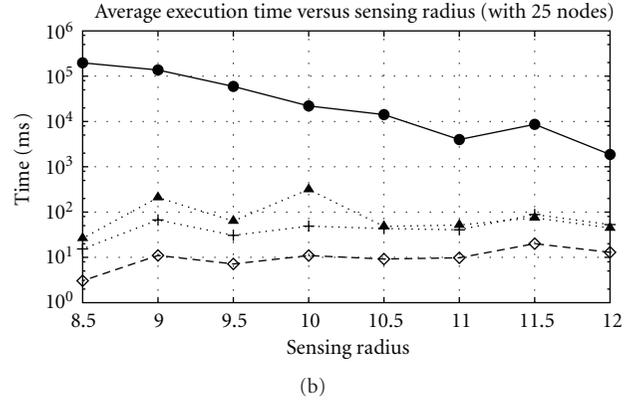
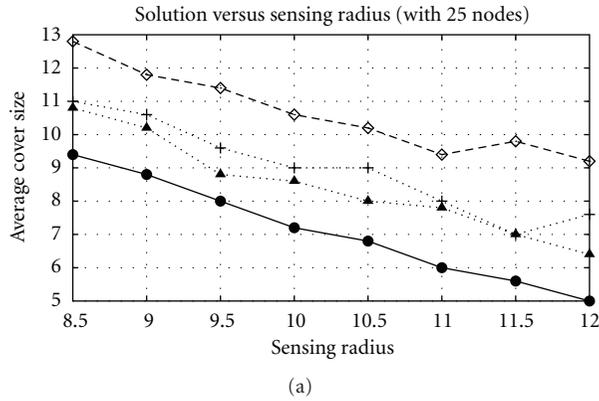


FIGURE 16: Performance evaluation: the nodes are deployed in a Gaussian distribution with the peak located at the center of A and the variance 15. Here A is a 30×30 square region.

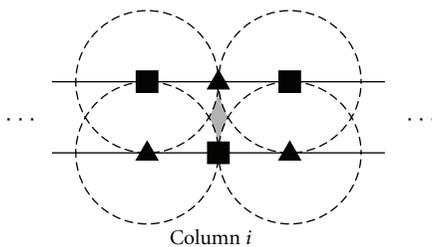


FIGURE 17: The shaded region cannot be covered while none of nodes from the column-pair i is active.

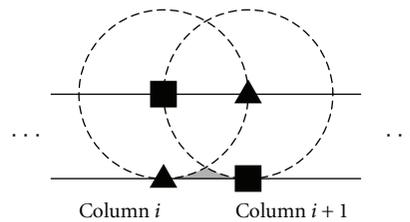


FIGURE 18: The shaded region cannot be covered by two nearby opposite polar nodes.

Figure 21(a), only one node, n_5 or n_6 , from P_1 being active cannot unit-disk cover A_c . Thus, if $U \subseteq N_c$ and $|U \cap P_1| < |P_1|/2 = 2$, U is not a UDC of A_c . That is, if U is a UDC of A_c , $|U \cap P_1| \geq |P_1|/2$.

Furthermore, Figure 21 lists all possible cases in which only two opposite polar nodes from P_1 being active cannot unit-disk cover A_c even all nodes in P_2 are active.

As shown in Figure 22(d), only two active nodes n_1 and n_2 from P_2 cannot unit-disk cover A_c even all nodes in P_1 are active.

active, which implies that only one node, n_1 or n_2 , from P_2 being active cannot unit-disk cover A_c . Similarly from Figure 22(a), only one node, n_3 or n_8 , from P_2 being active cannot unit-disk cover A_c . Thus, if $U \subseteq N_c$ and $|U \cap P_2| < |P_2|/2 = 2$, U is not a UDC of A_c . That is, if U is a UDC of A_c , $|U \cap P_2| \geq |P_2|/2$. Thus, together with the result from P_1 , precondition 7(ii) is satisfied.

Furthermore, Figure 22 lists all possible cases in which only two opposite polar nodes from P_2 being active cannot unit-disk cover A_c even all nodes in P_1 are active.

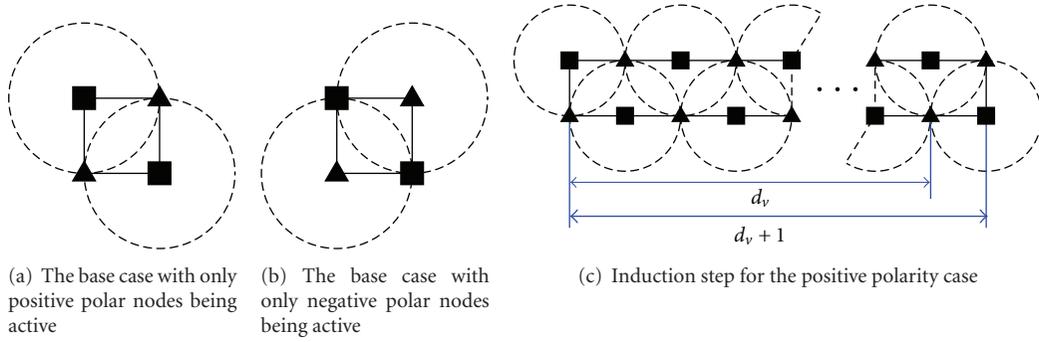


FIGURE 19: Induction proof for that a d_v long variable structure can be unit-disk covered by $d_v + 1$ nodes with the same polarity.

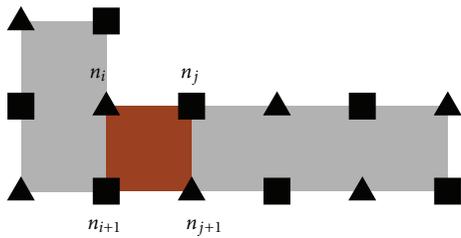


FIGURE 20: An edge is a composite structure of variable structures. Note that a variable structure can be rotated by 90° and still satisfies Lemmas 8 and 29. Here (n_i, n_{i+1}) and (n_j, n_{j+1}) are the connected ports of the connection patch.

Figure 23 illustrates possible cases in which A_c can be unit-disk covered with two same polar nodes from P_1 and two same polar nodes from P_2 being active. Together with Figures 21 and 22, precondition 7(iii) is satisfied. With satisfaction of preconditions 7(i), 7(ii), and 7(iii) for P_1 and P_2 , property (i) is satisfied. Furthermore, in Figure 23, at least one of the header nodes must be active, so property (iii) is also satisfied.

Figure 24 shows that if no header node is active, A_c cannot be unit-disk covered. That is, property (ii) is satisfied. Therefore, we prove that Lemma 9 holds for the 2-way connector shown in Figure 3(a).

B.2. 3-Way Connectors. Figures 3(b) and 4(b) illustrate the labels of nodes and vertices of A_c for the 3-way connector. Table 2 lists the positions of nodes and vertices relative to n_1 . Note that $\mathcal{P} = \{P_1, P_2, P_3\}$, $P_1 = \{n_7, n_8\}$, $P_2 = \{n_4, n_5, n_6, n_9\}$, $P_3 = \{n_1, n_2, n_3, n_{10}\}$, and the header nodes $h_1 = n_8$, $h_2 = n_9$, and $h_3 = n_{10}$. Obviously, $|P_i^+| = |P_i^-|$ for $i = 1, 2$, and 3, and, thus, precondition 7(i) is satisfied.

Figure 25 shows that A_c cannot be fully unit-disk covered without any node in P_1 being active, even if all nodes in P_2 and P_3 are active. Thus, if U is a UDC of A_c , $|U \cap P_1| \geq |P_1|/2 = 1$.

As shown in Figure 26(d), only two active nodes n_4 and n_9 from P_2 cannot unit-disk cover A_c even if all nodes in P_1 and P_3 are active, which implies that only one node, n_4 or n_9 , from P_2 being active cannot unit-disk cover A_c . Similarly from Figure 26(a), only one node, n_5 or n_6 , from P_2 being

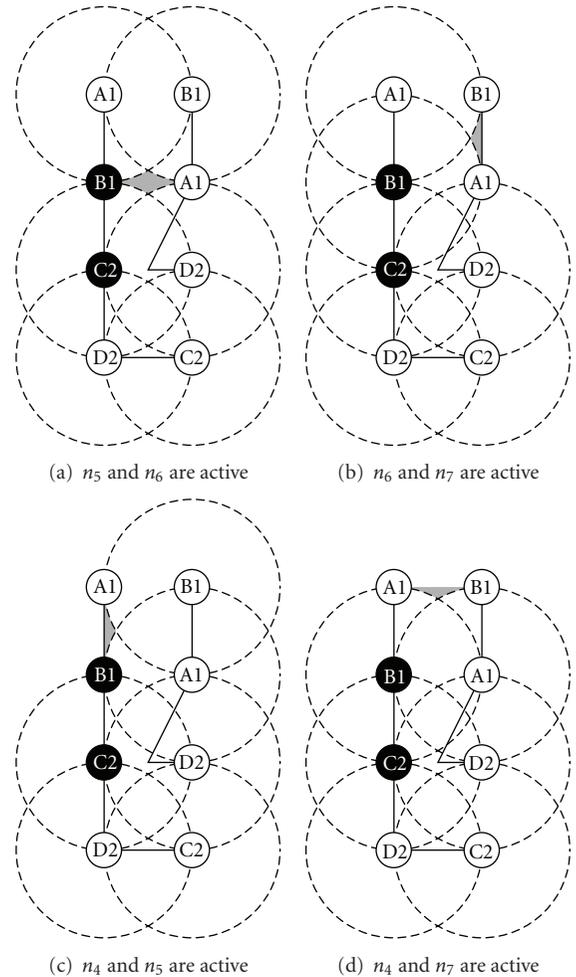


FIGURE 21: A_c cannot be fully unit-disk covered by two opposite polar nodes from P_1 , even all the nodes in P_2 are active. The shaded regions indicate the uncovered regions.

active cannot unit-disk cover A_c . Thus, if $U \subseteq N_c$ and $|U \cap P_2| < |P_2|/2 = 2$, U is not a UDC of A_c . That is, if U is a UDC of A_c , $|U \cap P_2| \geq |P_2|/2$.

Furthermore, Figure 26 lists all possible cases in which only two opposite polar nodes from P_2 being active cannot unit-disk cover A_c even all nodes in P_1 and P_3 are active.

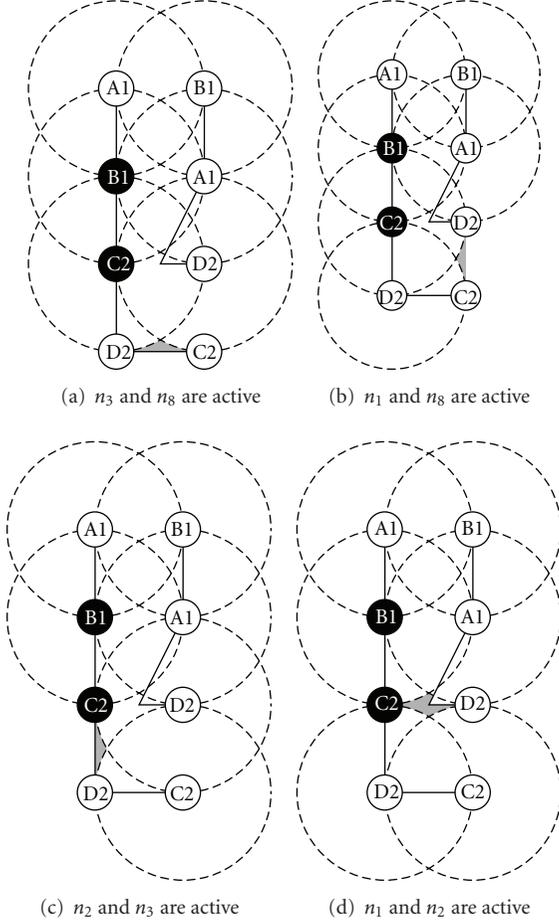


FIGURE 22: A_c cannot be fully unit-disk covered by two opposite polar nodes from P_2 , even if all the nodes in P_1 are active. The shaded regions indicate the uncovered regions.

As shown in Figure 27(d), only two active nodes n_1 and n_2 from P_3 cannot unit-disk cover A_c even all nodes in P_1 and P_2 are active, which implies that only one node, n_1 or n_2 , from P_3 being active cannot unit-disk cover A_c . Similarly from Figure 27(a), only one node, n_3 or n_{10} , from P_3 being active cannot unit-disk cover A_c . Thus, if $U \subseteq N_c$ and $|U \cap P_3| < |P_3|/2 = 2$, U is not a UDC of A_c . That is, if U is a UDC of A_c , $|U \cap P_3| \geq |P_3|/2$. Thus, together with the results from P_1 and P_2 , precondition 7(ii) is satisfied.

Furthermore, Figure 27 lists all possible cases in which only two opposite polar nodes from P_3 being active cannot unit-disk cover A_c even all nodes in P_1 and P_2 are active.

Figure 28 illustrates possible cases in which A_c can be unit-disk covered with one node from P_1 , two same polar nodes from P_2 , and two same polar nodes from P_3 being active. Together with Figures 26 and 27, precondition 7(iii) is satisfied. With satisfaction of preconditions 7(i), 7(ii), and 7(iii) for each port, property (i) is satisfied. Furthermore, in Figure 28, one of the header nodes must be active, so property (iii) is also satisfied.

Figure 29 shows that if no header node is active, A_c cannot be unit-disk covered. That is, property (ii) is satisfied.

Therefore, we prove that Lemma 9 holds for the 3-way connector shown in Figure 3(b).

C. Formal Definitions about Structure Connection

In this appendix, we define how structures may be connected together to form a complex structure.

Definition 25. The structure, $S = (A, N)$, is *connectable*, if, refer to Figure 5, there exists a pair of nodes, n_i and n_{i+1} , such that

- (i) $d(n_i, n_{i+1}) = 1$, (hence, n_i and n_{i+1} can be the vertices of a connection patch defined in Definition 26.)
- (ii) n_i and n_{i+1} have opposite polarities.

Furthermore, we call S connectable at n_i and n_{i+1} and the set $N_p = \{n_i, n_{i+1}\}$ a *port* of S .

Definition 26. Consider two connectable structure, $S_1 = (A_1, N_1)$ and $S_2 = (A_2, N_2)$. Suppose S_1 and S_2 are connected together via $T \times 1$ squares, $A_{cp,1}, A_{cp,2}, \dots, A_{cp,T}$, called *connection patches*. Each $A_{cp,t}$, $1 \leq t \leq T$, is attached to ports $N_{p_1,t}$ of S_1 and $N_{p_2,t}$ of S_2 . These ports are positioned at vertices of each connection patch as shown in Figure 5 and called *connected ports*. S_1 and S_2 are *well connected* if the following hold:

- (i) $A_1 \cap A_2 = \emptyset$ and $N_1 \cap N_2 = \emptyset$
- (ii) for $1 \leq t \leq T$, if $n \in N_{p_1,t}$ and $n' \in N_{p_2,t}$ on the same edge of $A_{cp,t}$, n and n' have opposite polarities, for example, (n_i, n_j) and (n_{i+1}, n_{j+1}) in Figure 5. Furthermore, n and n' are each other's *connection counterpart*.

In this NP-completeness proof, we would like to show that two structures are connected in such a way that the nodes, except connected ports, of one structure cannot cover any point of the other structure for preserving the partially well-behaved property. Thus, we have the following definitions.

Definition 27. The structure, $S = (A, N)$, is *least interactively connectable*, if there exists a port, $N_p = \{n_i, n_{i+1}\}$, such that the following preconditions hold.

- (i) There exists a point $x \in A$ that is not at the location of n_i or n_{i+1} and can only be unit-disk covered by n_i or n_{i+1} . (This precondition requires that at least one node from each connected port needs to be active.)
- (ii) For all $n \in (N - N_p)$, $\text{disk}(n) \cap A_{cp} \subset N_p$. Here A_{cp} is the connection patch attached to N_p . (Nonconnected port nodes and the connection patch are so far that nonconnected port nodes cannot cover any point, except the vertices, of the connection patch. Thus, whether a connection patch can be fully covered only depends on its connected ports.)

Furthermore, we call N_p the *least interactively connectable port*.

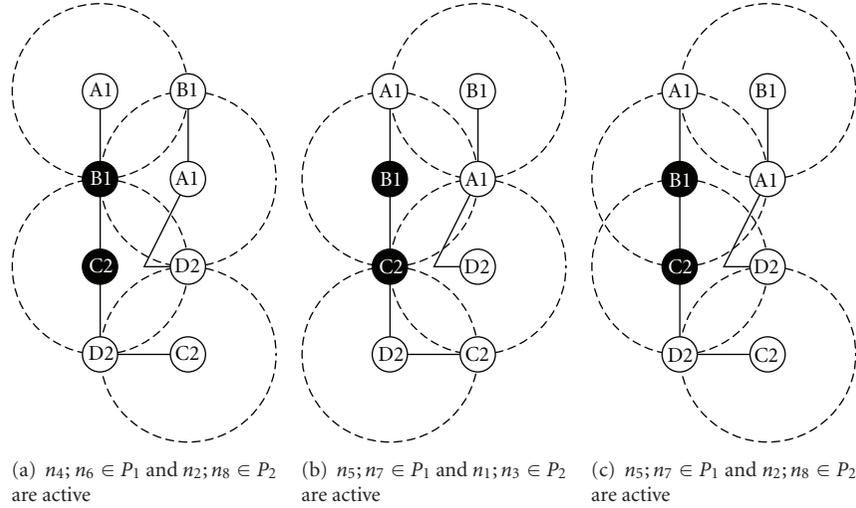


FIGURE 23: With at least one of the header nodes being active, A_c can be fully unit-disk covered by the nodes with the same polarity from each port.

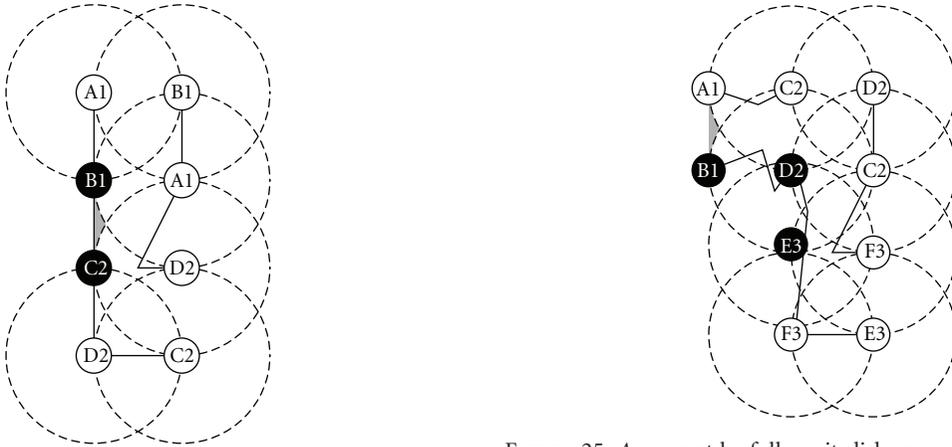


FIGURE 24: Without any header node being active, A_c cannot be fully unit-disk covered, even all the rest of nodes are active.

FIGURE 25: A_c cannot be fully unit-disk covered without any node in P_1 being active, even if all the nodes in P_2 and P_3 are active. The shaded region indicates the uncovered region.

Definition 28. The least interactively connectable structures $S_1 = (A_1, N_1)$ and $S_2 = (A_2, N_2)$ are well connected via the connection patches $A_{cp,1}, A_{cp,2}, \dots, A_{cp,T}$. For $1 \leq t \leq T$, $A_{cp,t}$ is attached to the least interactively connectable ports $N_{p_1,t}$ of S_1 and $N_{p_2,t}$ of S_2 . We call S_1 and S_2 *least interactively connected* if the following preconditions hold.

- (i) for all $n \in (N_1 - \bigcup_{1 \leq t \leq T} N_{p_1,t})$, for all $x \in A_2$, $d(n, x) > 1$ and for all $n \in (N_2 - \bigcup_{1 \leq t \leq T} N_{p_2,t})$, for all $x \in A_1$, $d(n, x) > 1$. (The distance between any nonconnected port nodes of one structure and any point of the other structure is greater than 1. Thus, the nodes, except connected ports, of one structure cannot cover any point of the other structure.)
- (ii) for $1 \leq t \leq T$, for all $n \in N_{p_1,t}$, $\text{disk}(n) \cap A_2 \subset N_{p_2,t}$ and for all $n \in N_{p_2,t}$, $\text{disk}(n) \cap A_1 \subset N_{p_1,t}$. (The connected ports of one structure cannot cover any point,

except their connection counterparts, of the other structure.)

D. The Least Interactive Connectability of Structures

Lemma 29. The structure $S_v = (A_v, N_v)$ shown in Figure 1 is least interactively connectable at any two nearby nodes on the side of border.

Proof. Suppose n_i and n_{i+1} are two nearby nodes as shown in Figure 1. It is not difficult to prove that $\{n_i, n_{i+1}\}$ is a port.

Furthermore, as shown in Figure 1, x , the midpoint of n_i and n_{i+1} , is the point satisfying precondition 27(i). Finally, it is obvious that for all $n \in (N_v - \{n_i, n_{i+1}\})$, $d(n, n_i) \geq 1$ and $d(n, n_{i+1}) \geq 1$. Hence, it is easy to derive

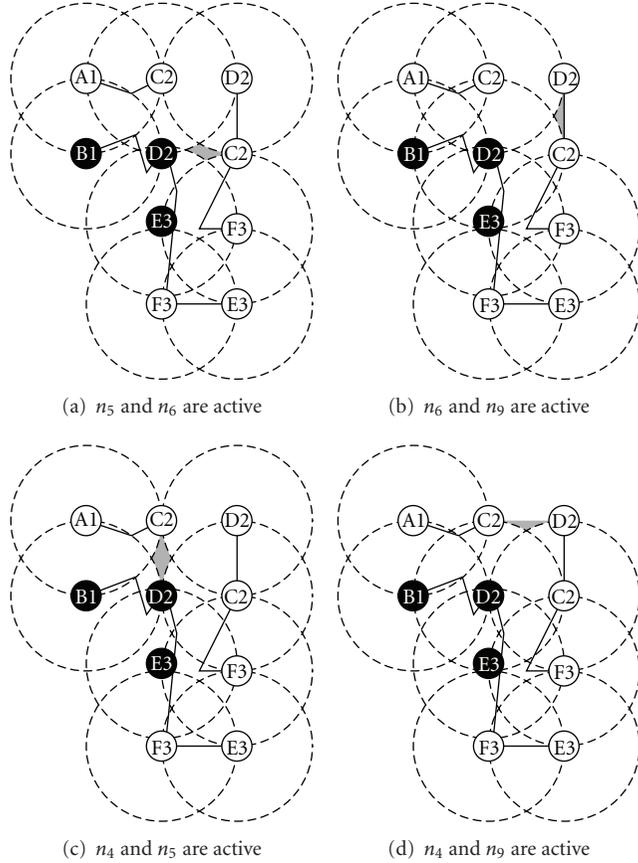


FIGURE 26: A_c cannot be fully unit-disk covered by two opposite polar nodes from P_2 , even if all the nodes in P_1 and P_3 are active. The shaded regions indicate the uncovered regions.

that precondition 27(ii) is satisfied. Therefore, S_v is least interactively connectable at n_i and n_{i+1} . \square

Lemma 30. *The structure $S_e = (A_e, N_e)$ shown in Figure 2 is least interactively connectable at the endpoint pairs indicated by the arrows.*

Proof. It is obvious that the endpoint pairs are also ports, and for each endpoint pair, for example, $\{n_i, n_{i+1}\}$, for all $n \in (N_e - \{n_i, n_{i+1}\})$, $d(n, n_i) \geq 1$ and $d(n, n_{i+1}) \geq 1$. Hence, it is easy to derive that precondition 27(ii) is satisfied at each endpoint pair. Besides, similar to variable structures, the midpoint of each endpoint pair can only be unit-disk covered by the endpoint pair. Thus, S_e is least interactively connectable at the endpoint pairs. \square

Lemma 31. *Each n -way connector of Figure 3 is least interactively connectable at the pairs of nodes indicated by the arrows.*

Proof. It is obvious that the pairs indicated by the arrows are also ports. Furthermore, for each port, it is obvious that the midpoint of the port is the point satisfying precondition 27(i). Moreover, without loss of generality, it is straightforward to prove from Table 2 that, for the port $\{n_7, n_8\}$ of the 3-way connector, for all $n \in (N_c - \{n_7, n_8\})$, $d(n, n_7) \geq 1$

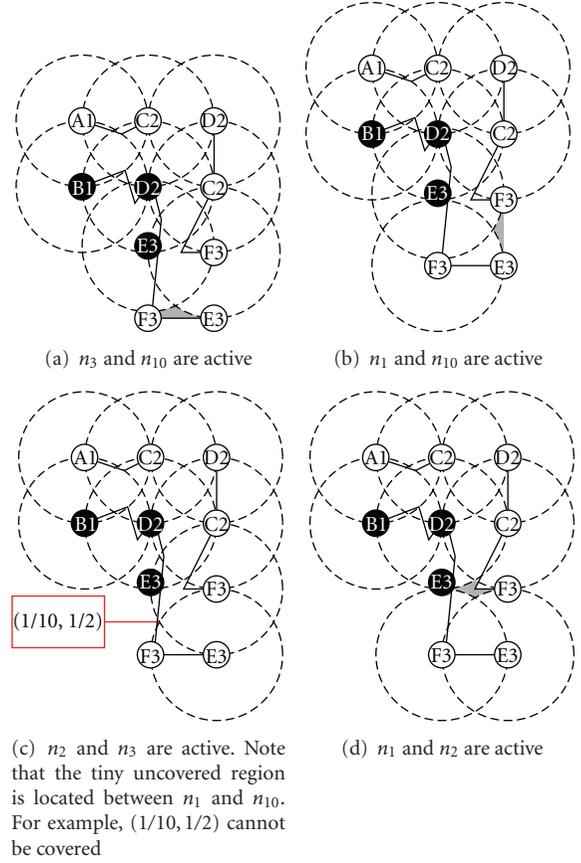


FIGURE 27: A_c cannot be fully unit-disk covered by two opposite polar nodes from P_2 , even if all the nodes in P_1 and P_3 are active. The shaded regions indicate the uncovered regions.

and $d(n, n_8) \geq 1$. Hence, it is easy to derive that precondition 27(ii) is satisfied at $\{n_7, n_8\}$. Similar argument may apply to other ports and the 2-way connector. \square

Lemma 32. *The structure $S_p = (A_p, N_p)$ shown in Figure 9 is least interactively connectable at its endpoint pairs.*

Proof. It is obvious that the endpoint pairs are also ports, and for each endpoint pair, for example, $\{n_i, n_{i+1}\}$ for all $n \in (N_e - \{n_i, n_{i+1}\})$, $d(n, n_i) \geq 1$ and $d(n, n_{i+1}) \geq 1$. Hence, it is easy to derive that precondition 27(ii) is satisfied at each endpoint pair. Besides, similar to variable structures, the midpoint of each endpoint pair can only be unit-disk covered by the endpoint pair. Thus, S_e is least interactively connectable at the endpoint pairs. \square

E. Proof of Lemmas about the Least Interactive Connection

E.1. Proof of Lemma 12. First, consider the following lemma.

Lemma 33. *Suppose $S = (A_1, N_1) + (A_2, N_2)$. If $U \subseteq (N_1 \cup N_2)$ is a UDC of $A_1 \cup A_2$, then $U \cap N_1$ and $U \cap N_2$ are UDCs of A_1 and A_2 , respectively.*

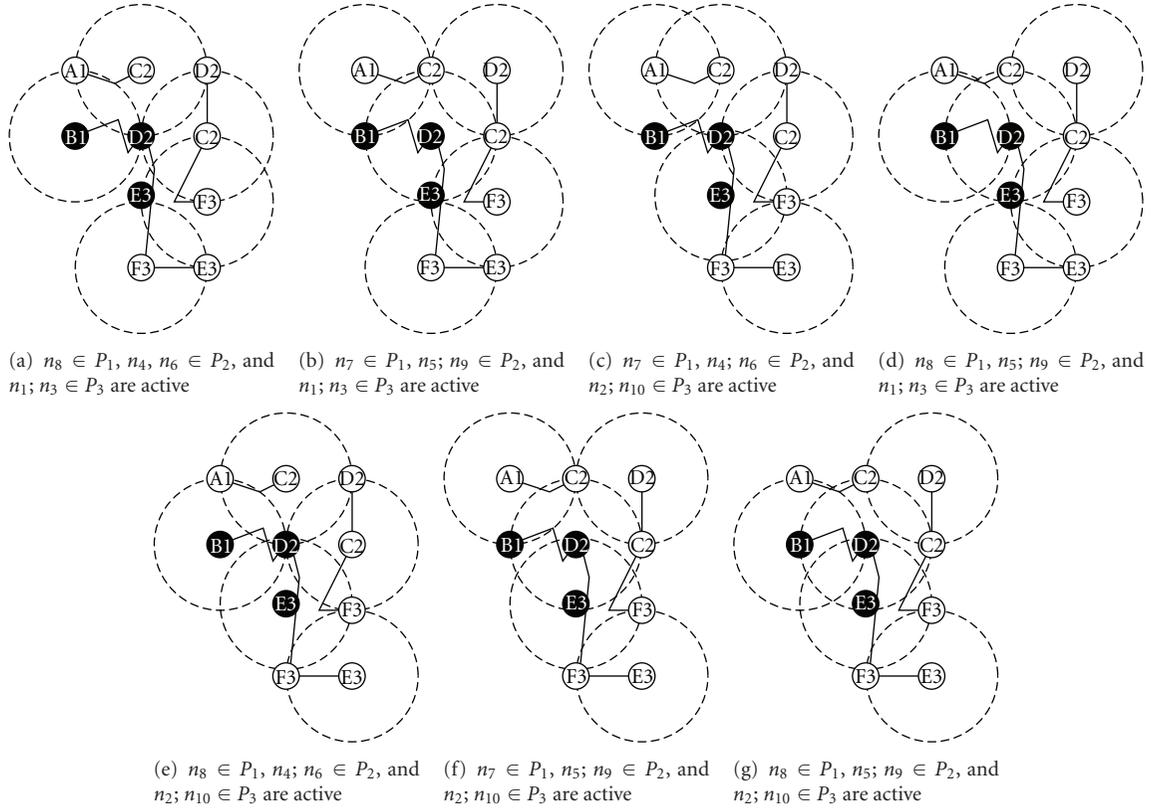


FIGURE 28: With at least one of the header nodes being active, A_c can be fully unit-disk covered by the nodes with the same polarity from each port

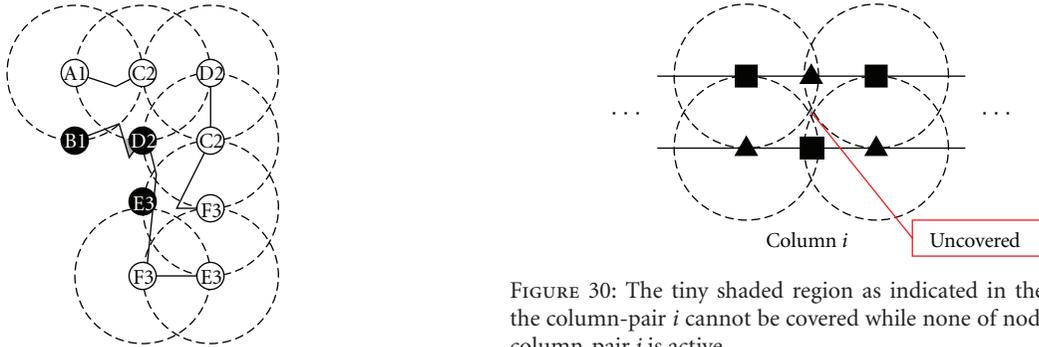


FIGURE 29: Without any header node being active, A_c cannot be fully unit-disk covered, even if all the rest of nodes are active.

Proof. Suppose $U \cap N_1$ is not a UDC of A_1 , then there exists a point $x^\dagger \in A_1$ such that x^\dagger cannot be covered by any node in $U \cap N_1$ but some node, say n , in $U \cap N_2$.

From Definition 28(i), we know the distance between any node in N_2 , except connected ports, and any point in A_1 is greater than 1. Thus, none of nodes in N_2 , except connected ports, can cover any point in A_1 . Hence, n must belong to some connected port. Without loss of generality, let n be n_j of Figure 5. From Definition 28(ii), the only point in A_1 which can be covered by n_j is at the location of n_i , and it implies that x^\dagger can only be at the location of n_i .

FIGURE 30: The tiny shaded region as indicated in the middle of the column-pair i cannot be covered while none of nodes from the column-pair i is active.

Let $x \in A_1$ be the point that is not at the location of n_i or n_{i+1} and can only be covered by n_i or n_{i+1} , as stated in Definition 27(i). From the above argument, x cannot be covered by any node of N_2 . Thus, at least one node from $\{n_i, n_{i+1}\}$ must be active in U . Hence, there is a contradiction since either n_i or n_{i+1} can cover x^\dagger , that is, the location of n_i . Therefore, $U \cap N_1$ is a UDC of A_1 and similar argument can also derive that $U \cap N_2$ is a UDC of A_2 . \square

With the help of Lemma 33, we can prove Lemma 12.

If $U \subseteq (N_1 \cup N_2)$ is a UDC of A , U is a UDC of $A_1 \cup A_2$. By Lemma 33, $U \cap N_1$ and $U \cap N_2$ are UDCs of A_1 and A_2 , respectively, which implies that $|U \cap N_1| \geq l_1$ and $|U \cap N_2| \geq l_2$.

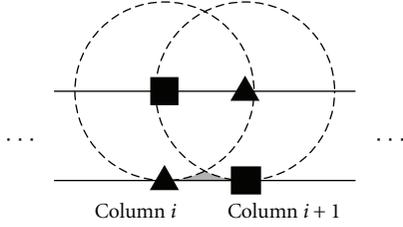
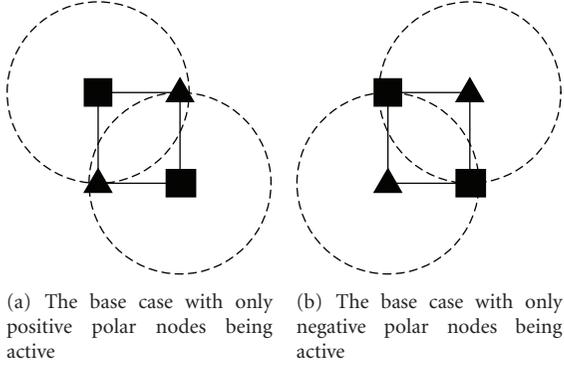
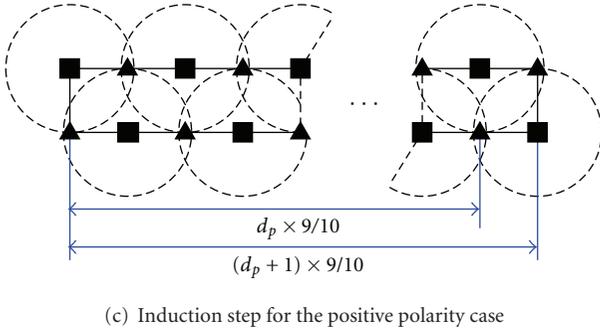


FIGURE 31: The shaded region cannot be covered by two nearby opposite polar nodes.



(a) The base case with only positive polar nodes being active (b) The base case with only negative polar nodes being active



(c) Induction step for the positive polarity case

FIGURE 32: Induction proof for that the $d_p \times (9/10)$ long main structure can be unit-disk covered by $d_p + 1$ nodes with the same polarity. Note that for the main structure of a polarity inverter, $d_p = 10$.

From Definition 26(i), $N_1 \cap N_2 = \emptyset$ and, hence, $|U| = |U \cap N_1| + |U \cap N_2| = l_1 + l_2$. Thus, $|U \cap N_1| = l_1$ and $|U \cap N_2| = l_2$.

E.2. Proof of Connection Lemma. Before proving Connection Lemma, consider the following lemma.

Lemma 34. *If $S = (A_1, N_1) + (A_2, N_2)$ via one connection patch, then the following propositions will hold. (We note that this lemma may be extended to more than one connection patches with possible minor modification on Definition 28(ii); however we do not use this property to prove Connection Lemma and therefore ignore it.)*

- (a) *If $U \subseteq (N_1 \cup N_2)$ is an MUDC of $A_1 \cup A_2$, then $U \cap N_1$ and $U \cap N_2$ are MUDCs of A_1 and A_2 , respectively.*
- (b) *If $U_1 \subseteq N_1$ and $U_2 \subseteq N_2$ are MUDCs of A_1 and A_2 , respectively, then $U_1 \cup U_2$ is an MUDC of $A_1 \cup A_2$.*

Proof. Let $U \subseteq (N_1 \cup N_2)$ be an MUDC of $A_1 \cup A_2$. By Lemma 33, $U \cap N_1$ and $U \cap N_2$ are UDCs of A_1 and A_2 respectively.

Suppose $U \cap N_1$ is not an MUDC of A_1 . Then there exists a UDC $U' \subseteq N_1$ of A_1 and $|U'| < |U \cap N_1|$. Since U' is a UDC of A_1 and $U \cap N_2$ is a UDC of A_2 , $U' \cup (U \cap N_2)$ is a UDC of $A_1 \cup A_2$. Since $N_1 \cap N_2 = \emptyset$ by Definition 26(i), $|U' \cup (U \cap N_2)| = |U'| + |U \cap N_2| < |U \cap N_1| + |U \cap N_2| = |(U \cap N_1) \cup (U \cap N_2)| = |U|$, which contradicts the assumption that U is an MUDC of $A_1 \cup A_2$. Similar argument can apply to $U \cap N_2$. Hence, we have proved proposition (a) of this lemma.

Now assume $U_1 \subseteq N_1$ and $U_2 \subseteq N_2$ are MUDCs of A_1 and A_2 , respectively, and, obviously, $U_1 \cup U_2$ is a UDC of $A_1 \cup A_2$. Suppose there exists an MUDC U'' of $A_1 \cup A_2$ and $|U''| < |U_1 \cup U_2|$. By proposition (a), $U'' \cap N_1$ and $U'' \cap N_2$ are MUDC of A_1 and A_2 , respectively. Again, since $N_1 \cap N_2 = \emptyset$, $|U''| = |U'' \cap N_1| + |U'' \cap N_2| < |U_1 \cup U_2| = |U_1| + |U_2|$. Hence, $|U'' \cap N_1| < |U_1|$ or $|U'' \cap N_2| < |U_2|$, which contradicts the assumption that U_1 and U_2 are MUDCs of A_1 and A_2 , respectively. Hence, $U_1 \cup U_2$ is an MUDC of $A_1 \cup A_2$. \square

With the help of Lemmas 33 and 34, we can prove Lemma 13.

Let A_{cp} be the connection patch. The key point to the proof is precondition (ii). That is, A_{cp} can be automatically covered by the active nodes of ports without the help of nodes not in U'_1 and U'_2 , if $U'_1 \cap N'_1$ and $U'_2 \cap N'_2$ have the same polarity. Thus, the MUDC of $A_1 \cup A_2 \cup A_{cp}$ can only be $U'_1 \cup U'_2$.

Note that $N'_1 \cap N'_2 = \emptyset$ by Definition 26(i). Thus, precondition 7(i) is satisfied since S_1 and S_2 are partially well behaved on N'_1 and N'_2 , respectively.

Moreover, let $\tilde{U} \subseteq (N_1 \cup N_2)$ be a UDC of $A_1 \cup A_2 \cup A_{cp}$. By Lemma 33, it is easy to derive that $\tilde{U} \cap N_1$ and $\tilde{U} \cap N_2$ are UDCs of A_1 and A_2 , respectively. Then by precondition (i) and Definition 7(ii), $|\tilde{U} \cap N'_1| \geq |N'_1|/2$ and $|\tilde{U} \cap N'_2| \geq |N'_2|/2$. Since $N'_1 \cap N'_2 = \emptyset$, $|\tilde{U} \cap (N'_1 \cup N'_2)| = |\tilde{U} \cap N'_1| + |\tilde{U} \cap N'_2| \geq (|N'_1| + |N'_2|)/2 = |N'_1 \cup N'_2|/2$. That is, precondition 7(ii) is satisfied.

Now, suppose $U' \subseteq (N_1 \cup N_2)$ is an MUDC of $A_1 \cup A_2$ and $U \subseteq N_1 \cup N_2$ is an MUDC of $A_1 \cup A_2 \cup A_{cp}$. Obviously, $|U| \geq |U'|$. Since \tilde{U} is a UDC of $A_1 \cup A_2 \cup A_{cp}$, $|\tilde{U}| \geq |U| \geq |U'|$. Consequently, by showing there exists an MUDC U' of $A_1 \cup A_2$ such that U' is a UDC of $A_1 \cup A_2 \cup A_{cp}$, we can prove that $|U| = |U'|$, which implies that U is also an MUDC of $A_1 \cup A_2$.

Note that precondition (ii) states that there exists an MUDC of A_1 , $U'_1 \subseteq N_1$, and an MUDC of A_2 , $U'_2 \subseteq N_2$, such that $U'_1 \cap N'_1$ and $U'_2 \cap N'_2$ have the same polarity. Let $U' = U'_1 \cup U'_2$ be such a candidate. Hence, by Lemma 34(b), U' is an MUDC of $A_1 \cup A_2$.

Since $U'_1 \cap N'_1$ and $U'_2 \cap N'_2$ have the same polarity, $U' \cap (N'_1 \cup N'_2) = (U'_1 \cap N'_1) \cup (U'_2 \cap N'_2) = N'^+_1 \cup N'^+_2$ or $N'^-_1 \cup N'^-_2$. Referring to Figure 5, by Definitions 25(ii) and 26(ii), either the pair (n_i, n_{i+1}) or (n_{i+1}, n_i) is active. Obviously, either pair can unit-disk cover A_{cp} by Lemma 10.

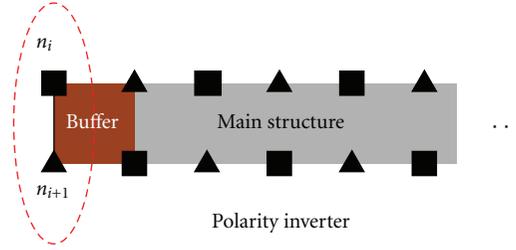


FIGURE 33: Connection Lemma is applied to the main structure and the left end $S_l = (A_l, N_l)$ (enclosed in dashed line). A_l is the line segment $\overline{n_i n_{i+1}}$ and $N_l = \{n_i, n_{i+1}\}$.

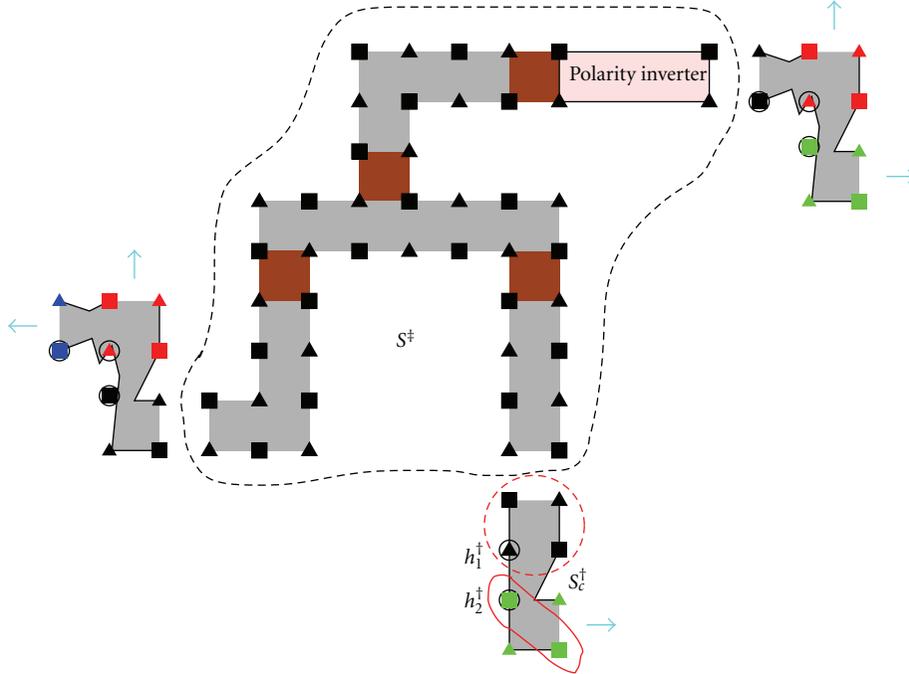


FIGURE 34: The structure enclosed in dashed line is S^{\ddagger} . The set of nodes enclosed in dashed circle is P_1^{\ddagger} , and the set of nodes enclosed in solid line is P_2^{\ddagger} . By proving $S^{\ddagger} + S_c^{\ddagger}$ is partially well-behaved on $N^{\ddagger} \cup P_1^{\ddagger}$, this lemma will be proved.

Therefore, U' is a UDC of $A_1 \cup A_2 \cup A_{cp}$, which implies that, for any MUDC U of $A_1 \cup A_2 \cup A_{cp}$, $|U| = |U'|$ and U is also an MUDC of $A_1 \cup A_2$.

Now we need to prove that, for any MUDC U of $A_1 \cup A_2 \cup A_{cp}$, $U \cap (N_1' \cup N_2')$ only contains the nodes with the same polarity. Since U is also an MUDC of $A_1 \cup A_2$, by Lemma 34(a), $U \cap N_1$, denoted as U_1 , is an MUDC of A_1 and $U \cap N_2$, denoted as U_2 , is also an MUDC of A_2 . By Definition 7(iii), $U_1 \cap N_1'$ only contains the nodes with the same polarity and so does $U_2 \cap N_2'$. Thus, we need to prove that $U \cap (N_1' \cup N_2')$ cannot be $N_1'^+ \cup N_2'^-$ or $N_1'^- \cup N_2'^+$. Referring to Figure 5, without loss of generality, let $U \cap (N_1' \cup N_2') = N_1'^+ \cup N_2'^-$ with $n_i \in N_1'^+$ and $n_j \in N_2'^-$ being active. Note that it is obvious that $A_{cp} \not\subseteq \text{disk}(\{n_i, n_j\})$. (In other words, A_{cp} cannot be covered by n_i and n_j .) That is, $\text{disk}(\{n_i, n_j\}) \cap A_{cp} \subsetneq A_{cp}$.

Denote $\overline{N_1} = N_1 - \{n_i, n_{i+1}\}$. Obviously, $N_1'^+ \subset (\overline{N_1} \cup \{n_i\})$, and, hence, $\text{disk}(N_1'^+) \cap A_{cp} \subseteq \text{disk}(\overline{N_1} \cup \{n_i\}) \cap A_{cp}$. From Definition 27(ii), for all $n \in \overline{N_1}$, $\text{disk}(n) \cap A_{cp} \subset \{n_i, n_{i+1}\}$. Thus, $\text{disk}(\overline{N_1}) \cap A_{cp} \subset \{n_i, n_{i+1}\} \subset \text{disk}(n_i) \cap A_{cp}$,

which implies $\text{disk}(\overline{N_1} \cup \{n_i\}) \cap A_{cp} = \text{disk}(n_i) \cap A_{cp}$. Hence, $\text{disk}(N_1'^+) \cap A_{cp} \subseteq \text{disk}(n_i) \cap A_{cp}$. Similarly, $\text{disk}(N_2'^-) \cap A_{cp} \subseteq \text{disk}(n_j) \cap A_{cp}$.

Therefore, $\text{disk}(N_1'^+ \cup N_2'^-) \cap A_{cp} = (\text{disk}(N_1'^+) \cap A_{cp}) \cup (\text{disk}(N_2'^-) \cap A_{cp}) \subseteq (\text{disk}(n_i) \cap A_{cp}) \cup (\text{disk}(n_j) \cap A_{cp}) = \text{disk}(\{n_i, n_j\}) \cap A_{cp} \subsetneq A_{cp}$. That is, $A_{cp} \not\subseteq \text{disk}(N_1'^+ \cup N_2'^-)$. Thus, simply nodes from $N_1'^+$ and $N_2'^-$ cannot unit-disk cover A_{cp} , and at least one more node not in U is needed, for example, $n_{i+1} \in N_1'^-$ or $n_{j+1} \in N_2'^+$. Hence, U cannot be an MUDC of $A_1 \cup A_2 \cup A_{cp}$.

Therefore, we can conclude that if $U \subseteq (N_1 \cup N_2)$ is an MUDC of $A_1 \cup A_2 \cup A_{cp}$, $U \cap (N_1' \cup N_2')$ contains only the nodes with the same polarity. That is, precondition 7(iii) is satisfied and, thus, S is partially well behaved on $N_1' \cup N_2'$.

F. Proof of Lemma 14

Similar to the proof of variable structures in Lemma 8, the well-aligned and well-behaved properties of the main

structure can be proved via Figures 30, 31, and 32. Consequently, with the help of Lemma 10 and Connection Lemma, the well-aligned and well-behaved properties of S_p can be proved. Note that, in this case, Connection Lemma is applied to the main structure and both “end”, $S_l = (A_l, N_l)$ and $S_r = (A_r, N_r)$. Referring to Figure 33, A_l is the line segment $\overline{n_i n_{i+1}}$ and $N_l = \{n_i, n_{i+1}\}$. Besides, the left buffer is the connection patch for connecting S_l and the main structure. Similar idea can apply to the right end, that is, S_r .

G. Proof of Lemma 17

For the i th variable v_i , consider the composite structure $S^\ddagger = (A^\ddagger, N^\ddagger)$ with $A^\ddagger = AV_i \cup AE_i \cup AI_i \cup ACP_i^*$ and $N^\ddagger = NV_i \cup NE_i \cup NI_i$. Here ACP_i^* is the union of the shaded region from all the associated connection patches, except the ones attached to the associated connectors. An example of S^\ddagger is illustrated in Figure 34. In other words, the composite structure of S^\ddagger and all associated connectors is the territory T_i . It is obvious that S^\ddagger is well behaved by Lemmas 8, 14, and Connection Lemma. After that, we connect each associated n -way connector to S^\ddagger iteratively. At each iteration, we can prove the partially well behaved property of the composite structure. When all associated connectors are connected, this lemma is proved.

Without loss of generality, consider an MUDC of A^\ddagger , $N^{\ddagger+}$. Since $2 \leq |c| \leq 3$ for each clause c , the clauses are represented by 2-way or 3-way connectors. Note that an P3SAT instance has a graph structure, G_B , in which there is at most one edge between two nodes. That is, the variables in a clause are different. Thus, for each associated connector of T_i , there is another partition, that is, not enclosed in dashed line in Figure 10, which is not an associated partition. Without loss of generality, suppose $S_c^\ddagger = (A_c^\ddagger, N_c^\ddagger \langle \mathcal{P}^\ddagger, H^\ddagger \rangle)$ is an associated 2-way connector of S_{v_i} . Here $\mathcal{P}^\ddagger = \{P_1^\ddagger, P_2^\ddagger\}$ and $H^\ddagger = \{h_1^\ddagger, h_2^\ddagger\}$. Referring to Figure 34, let P_1^\ddagger be the associated partition and $P_2^{\ddagger P} = \{n \in P_2^\ddagger \mid n \text{ has the polarity of } h_2^\ddagger\}$. We want to prove $S^\ddagger + S_c^\ddagger$ is partially well behaved on $N^\ddagger \cup P_1^\ddagger$.

From Lemma 9(i), Lemma 9(iii), and Definition 7, it is easy to derive that $P_1^{\ddagger+} \cup P_2^{\ddagger P}$ is an MUDC of A_c^\ddagger . Thus, $N^{\ddagger+}$ and $P_1^{\ddagger+} \cup P_2^{\ddagger P}$ are the MUDCs of A^\ddagger and A_c^\ddagger , respectively, and precondition (ii) of Connection Lemma is satisfied. (In this case, $A_1 = A^\ddagger$, $A_2 = A_c^\ddagger$, $N'_1 = N^\ddagger$, $N'_2 = P_1^{\ddagger+}$, $U'_1 = N^{\ddagger+}$, and $U'_2 = P_1^{\ddagger+} \cup P_2^{\ddagger P}$.) Besides, as mentioned earlier, the variables in a clause are different. That is, S^\ddagger and S_c^\ddagger are connected via one connection patch. Thus, Connection Lemma is applicable. By Lemma 9(i) and Connection Lemma, $S^\ddagger + S_c^\ddagger$ is partially well behaved on $N^\ddagger \cup P_1^\ddagger$. This procedure can be iteratively applied to the rest of connectors and, hence, the lemma is proved.

Acknowledgment

This research was supported by National Science Council (NSC), Taiwan, under Grant NSC 99-2221-E-194-021. The author gratefully acknowledges this 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] ANSI/IEEE, Standard 802.11, “Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) specifications,” 1999.
- [3] J. A. Stine and G. D. Veciana, “Improving energy efficiency of centrally controlled wireless data networks,” *Wireless Networks*, vol. 8, no. 6, pp. 681–700, 2002.
- [4] A. Srinivas and E. Modiano, “Minimum energy disjoint path routing in wireless Ad-hoc networks,” in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pp. 122–133, ACM Press, San Diego, Calif, USA, 2003.
- [5] A. Sankar and Z. Liu, “Maximum lifetime routing in wireless Ad-hoc networks,” in *Proceedings of the IEEE Infocom*, Hong Kong, China, March 2004.
- [6] S. Singh, C. S. Raghavendra, and J. Stepanek, “Power-aware broadcasting in mobile Ad Hoc networks,” in *Proceedings of the 10th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, Osaka, Japan, September 1999.
- [7] J. Wu, B. Wu, and I. Stojmenovic, “Power-aware broadcasting and activity scheduling in Ad Hoc wireless networks using connected dominating sets,” *Wireless Communications and Mobile Computing*, vol. 3, no. 4, pp. 425–438, 2003.
- [8] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, “Multicasting in energy-limited Ad-hoc wireless networks,” in *Proceedings of the IEEE Military Communications Conference*, pp. 723–729, October 1998.
- [9] W. Liang, “Approximate minimum-energy multicasting in wireless Ad Hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 4, pp. 377–387, 2006.
- [10] L. Hu, “Topology control for multihop packet radio networks,” *IEEE Transactions on Communications*, vol. 41, no. 10, pp. 1474–1481, 1993.
- [11] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, “Distributed topology control for power efficient operation in multihop wireless Ad Hoc networks,” in *Proceedings of the IEEE Infocom*, pp. 1388–1397, Anchorage, AK, USA, April 2001.
- [12] R. C. Shah and J. M. Rabaey, “Energy aware routing for low energy Ad Hoc sensor networks,” in *Proceedings of the IEEE Wireless Communication and Networking Conference*, Orlando, Fla, USA, March 2002.
- [13] J.-H. Chang and L. Tassiulas, “Maximum lifetime routing in wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, 2004.
- [14] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, “Optimizing sensor networks in the energy-latency-density design space,” *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70–80, 2002.
- [15] H. Nama and N. Mandayam, “Sensor networks over information fields: optimal energy and node distributions,” in *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 1842–1847, March 2005.
- [16] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, “Energy-aware wireless microsensor networks,” *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.
- [17] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, “Coverage problems in wireless Ad-hoc sensor networks,” in *Proceedings of the IEEE Infocom*, pp. 1380–1387, Anchorage, Alaska, USA, April 2001.

- [18] 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*, pp. 1734–1743, San Francisco, Calif, USA, April 2003.
- [19] Y.-C. Tseng, S.-P. Kuo, H.-W. Lee, and C.-F. Huang, "Location tracking in a wireless sensor network by mobile agents and its data fusion strategies," in *Proceedings of the 2nd International Symposium on Information Processing in Sensor Networks*, pp. 625–641, Palo Alto, Calif, USA, April 2003.
- [20] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications*, vol. 2, pp. 472–476, Helsinki, Finland, June 2001.
- [21] S. Funke, A. Kesselman, F. Kuhn, Z. Lotker, and M. Segal, "Improved approximation algorithms for connected sensor cover," *Wireless Networks*, vol. 13, no. 2, pp. 153–164, 2007.
- [22] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, chapter A3, W.H. Freeman and Company, 1979.
- [23] D. S. Johnson, "Approximation algorithms for combinatorial problems," *Journal of Computer and System Sciences*, vol. 9, no. 3, pp. 256–278, 1974.
- [24] L. Lovász, "On the ratio of optimal integral and fractional covers," *Discrete Mathematics*, vol. 13, no. 4, pp. 383–390, 1975.
- [25] V. Chvátal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.
- [26] U. Feige, "A threshold of $\ln 2$ for approximating set cover," *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, 1998.
- [27] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in wireless Ad-Hoc sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 139–150, ACM Press, Rome, Italy, 2001.
- [28] S. Meguerdichian, S. Slijepcevic, V. Karayan, and M. Potkonjak, "Localized algorithms in wireless Ad-Hoc networks: location discovery and sensor exposure," in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 106–116, ACM Press, Long Beach, CA, USA, 2001.
- [29] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pp. 129–143, Philadelphia, PA, USA, October 2004.
- [30] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 32–41, ACM Press, Atlanta, Ga, USA, 2002.
- [31] B. Carburnar, A. Grama, J. Vitek, and O. Carburnar, "Redundancy and coverage detection in sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 1, pp. 94–128, 2006.
- [32] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," in *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, pp. 115–121, ACM Press, San Diego, Calif, USA, 2003.
- [33] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *The Wireless Ad Hoc and Sensor Networks*, vol. 1, pp. 89–124, 2005.
- [34] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp. 28–39, ACM Press, Los Angeles, Calif, USA, 2003.
- [35] H. Gupta, Z. Zhou, S. R. Das, and Q. Gu, "Connected sensor cover: self-organization of sensor networks for efficient query execution," *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, pp. 55–67, 2006.
- [36] A. V. S. Kumar, S. Arya, and H. Ramesh, "Hardness of set cover with intersection 1," in *Proceedings of the 27th International Colloquium on Automata, Languages and Programming*, pp. 624–635, Springer, London, UK, 2000.
- [37] R. J. Fowler, M. Paterson, and S. L. Tanimoto, "Optimal packing and covering in the plane are NP complete," *Information Processing Letters*, vol. 12, no. 3, pp. 133–137, 1981.
- [38] N. Megiddo and K. J. Supowit, "On the complexity of some common geometric location problems," *SIAM Journal on Computing*, vol. 13, no. 1, pp. 182–196, 1984.
- [39] M. V. Marathe, V. Radhakrishnan, I. Harry, B. Hunt, and S. S. Ravi, "Hierarchically specified unit disk graphs," *Theoretical Computer Science*, vol. 174, no. 1-2, pp. 23–65, 1997.
- [40] H. Brönnimann and M. T. Goodrich, "Almost optimal set covers in finite VC-dimension," *Discrete & Computational Geometry*, vol. 14, no. 4, pp. 463–479, 1995.
- [41] T. F. Gonzalez, "Covering a set of points in multidimensional space," *Information Processing Letters*, vol. 40, no. 4, pp. 181–188, 1991.
- [42] D. S. Hochbaum and W. Maass, "Approximation schemes for covering and packing problems in image processing and VLSI," *Journal of the ACM*, vol. 32, no. 1, pp. 130–136, 1985.
- [43] M. Franceschetti, M. Cook, and J. Bruck, "A geometric theorem for network design," *IEEE Transactions on Computers*, vol. 53, no. 4, pp. 483–489, 2004.
- [44] M. Franceschetti, M. Cook, and J. Bruck, "A Geometric Theorem for Approximate Disk Covering Algorithms," California Institute of Technology, Technical Report ETR035, 2001 <http://www.paradise.caltech.edu/papers/etr035.pdf>.
- [45] D. Lichtenstein, "Planar formulae and their uses," *SIAM Journal on Computing*, vol. 11, no. 2, pp. 329–343, 1982.
- [46] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, chapter A9, W.H. Freeman and Company, 1979.

Research Article

Robust Interval-Based Localization Algorithms for Mobile Sensor Networks

Farah Mourad,¹ Hichem Snoussi,¹ Michel Kieffer,^{2,3} and Cédric Richard⁴

¹*Institut Charles Delaunay (ICD)—LM2S (STMR UMR CNRS 6279), Université de Technologie de Troyes, 12 rue Marie Curie, 10010 Troyes, France*

²*L2S—CNRS—SUPELEC, Université Paris-Sud, 91192 Gif-sur-Yvette, France*

³*LTCI—CNRS—Telecom-ParisTech, 75013 Paris, France*

⁴*Laboratoire FIZEAU (UMR CNRS 6525), Université de Nice Sophia-Antipolis, Parc de Valrose, 06108 Nice, France*

Correspondence should be addressed to Farah Mourad, farah.mourad@utt.fr

Received 16 June 2011; Accepted 27 August 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Farah Mourad et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper considers the localization problem in mobile sensor networks. Such a problem is a challenging task, especially when measurements exchanged between sensors may contain outliers, that is, data not matching the observation model. This paper proposes two algorithms robust to outliers. These algorithms perform a set-membership estimation, where only the maximal number of outliers is required to be known. Using these algorithms, estimates consist of sets of boxes whose union surely contains the correct location of the sensor, provided that the considered hypotheses are satisfied. This paper proposes as well a technique for evaluating the number of outliers to be robust to. In order to corroborate the efficiency of both algorithms, a comparison of their performances is performed in simulations using Matlab.

1. Introduction

Mobile Sensor Networks (MSNs) have recently emerged as a challenging research field. An MSN consists of a large number of low-cost smart sensors with limited computational capacities and energy resources [1]. Due to the lack of a fixed infrastructure in MSNs, the sensors are able to move in an uncontrolled manner. For this reason and since sensed data are related to the locations of the sensors in almost all MSN applications, many researchers have focused on the localization problem. A first solution for sensor localization is to equip all sensors with global positioning systems (GPSs) [2]. However, this solution is nonpractical in MSNs, since GPS are expensive, high energy consuming, and having great sizes. The alternative solution consists of equipping a few number of sensors with GPS receivers. These sensors, aware of their locations, are called *anchors*. The remaining sensors, called *nonanchor nodes* or simply *nodes*, have unknown locations, and hence they need to be localized.

Many anchor-based algorithms have been proposed for sensor localization. For instance, Doherty et al. [3] propose

a centralized technique for position estimation. Localization in [3] is formalized as a convex optimization problem, having connectivity measurements between sensors as constraints. In [4–6], different approaches requiring few anchors have been proposed. Local maps with relative positions are constructed using measured distances between nodes and their neighbors. A combination of these maps with known positions of anchors lead to absolute positions. Nevertheless, these techniques are not very robust because of errors accumulation while combining the maps. Authors in [7] propose a distributed static algorithm, where each node defines its position as the center of all observed anchors. In a different scenario [8], Galstyan et al. propose an online distributed technique, where nodes use their detection of a moving target to update their position estimates. Blatt and Hero [9] address the problem of source localization using sensors measurements. The problem is formulated as a convex problem that is solved using the aggregated projection onto convex sets (APOCSs) method. In [10–12], dynamic approaches, on the basis of sequential Monte-Carlo [13], are considered to estimate the positions of the nodes. The posterior distribution

of the unknown positions is estimated recursively with a set of *particles*. Terwilliger et al. [14] propose to cover all possible solutions with the smallest enclosing disk. Alternative dynamic algorithms for sensor localization, using interval analysis [15], have been proposed in [16, 17]. Position estimates are boxes covering the possible locations of the sensors.

Existing methods have mainly considered the localization problem with the hypothesis that all measurements are consistent with the considered measurement model. However, in practical situations, outliers, that is, data not matching the measurement model, are encountered. Previously mentioned estimation techniques are not very robust to such outliers. In [18], Jaulin et al. propose a set-membership estimator robust to outliers based on interval analysis. Savarese et al. [19] present a distributed robust algorithm for sensor localization. The method is separated into two phases: the start-up phase, where first estimates of node positions are computed using hop counts to anchors, and the refinement phase, where nodes communicate with their neighbors to update their positions using a least-squares triangulation technique. Nevertheless, a number of factors influence the convergence of the refinement phase, such as the accuracy of first estimates and the magnitude of ranging errors. In [20], Rabbat et al. introduce a robust localization algorithm of an isotropic energy source using kernel averaging techniques. The proposed estimator is more robust than the least-squares estimator under a variety of conditions. Leger and Kieffer [21] present a distributed version of the estimation algorithm [18], assuming that the maximal number of outliers is known. In particular, a static distributed algorithm is proposed for source localization using received signal strength (RSS) measurements. The proposed method adapts the set inversion via interval analysis (SIVIA) algorithm [15] to evaluate a solution set.

In this paper, we propose an original adaptive approach for sensor localization in the presence of outliers. Assuming only that the maximal number of outliers is given, the proposed approach uses connectivity measurements in addition to a mobility model to address the localization problem. The solution is then given using either SIVIA or an alternative combinatorial technique. Another contribution of the paper is that it proposes a technique for evaluating the maximal number of outliers to be robust to. Moreover, using a connectivity-based observation model, the paper compares the performances of both robust localization algorithms.

The rest of the paper is organized as follows. Section 2 introduces the localization problem. A description of the SIVIA algorithm and the combinatorial technique is then given in Section 3. Section 4 provides simulation results, whereas Section 5 concludes the paper.

2. Problem Statement

The proposed method is an anchor-based method, where each node exchanges information with anchors to localize itself. Consider a network of N_a anchors and N_x mobile nodes. All sensors are assumed to be in the same plane: their locations at time t are given by $\mathbf{a}_i(t) = (a_{i,1}(t), a_{i,2}(t))^T$,

$i = 1, \dots, N_a$, for anchors and $\mathbf{x}_j(t) = (x_{j,1}(t), x_{j,2}(t))^T$, $j = 1, \dots, N_x$, for nodes. In order to reduce the communication costs during the localization process, the proposed method assumes that each mobile node does not exchange information with other nodes. For this reason and without loss of generality, we focus on the localization of one generic mobile node $\mathbf{x}(t) = (x_1(t), x_2(t))^T$, and we thus drop the index j .

2.1. Observation Model. At time t , the mobile node receives signals from a set $J(t) \subseteq \{1, \dots, N_a\}$ of anchors with specific received signal strengths (RSSs) denoted by $\rho_i(t)$, $i \in J(t)$. These RSSs are assumed to follow the Okumura-Hata model [22]

$$\rho_i(t) = \rho_0 - 10n_p \log_{10} \frac{d_i(\mathbf{x}(t))}{d_0} + \varepsilon_i(t). \quad (1)$$

In (1), $\rho_i(t)$ is in dBm, ρ_0 is the power measured (in dBm) at a reference distance d_0 from the anchor $\mathbf{a}_i(t)$, n_p is the path-loss exponent, $d_i(\mathbf{x}(t)) = \|\mathbf{x}(t) - \mathbf{a}_i(t)\|$ is the Euclidian distance between the anchor $\mathbf{a}_i(t)$ and the considered node, and $\varepsilon_i(t)$ is the measurement noise, modeled as zero-mean Gaussian with variance σ^2 .

In practice, ρ_0 and n_p may vary from one anchor to the other, and σ^2 may be quite large. Given the RSS values, the proposed model may lead to inaccurate distance estimates. For this reason, only connectivity information are employed and (1) is only used to determine whether the node is in the vicinity of the i th anchor. Let ρ_r be some RSS threshold corresponding to a distance r , which is the sensing range of the sensors. Then, if $\rho_i(t) \geq \rho_r$, the distance $d_i(\mathbf{x}(t))$ from the anchor i to the node is deemed less than r . Anchors for which $\rho_i(t) \geq \rho_r$ are called *detected anchors*. Only detected anchors are then taken into account for the localization. The observation model is then given by

$$(x_1(t) - a_{i,1}(t))^2 + (x_2(t) - a_{i,2}(t))^2 \leq r^2, \quad i \in I(t), \quad (2)$$

where $I(t) \subseteq J(t)$ is the set of indices of detected anchors, that is, whose emitted signals have RSS at the node $\mathbf{x}(t)$ larger than ρ_r . The observation model is thus given by a set of disk equations centered on the detected anchors and having r as radius.

In real environments, measurements may not follow exactly the observation model. Indeed, due to the additive noise and the inaccuracy of the parameter values, a measured RSS $\rho_i(t)$ could be less than ρ_r while (2) is satisfied for real and vice versa. In the first case, the anchor is assumed to be out of the vicinity of the node, which is not true, and thus, a correct measurement is omitted, whereas in the second case, an outlier is obtained. The proposed approach takes such outliers into consideration. Using the connectivity-based model, it assumes that the maximal number of outliers is known and denoted by q . In other words, it considers that $|I(t)| - q$ measurements at minimum are correct at each time step.

2.2. Mobility Model. The proposed method takes also advantage of the mobility of the nodes to improve the estimation

accuracy. Any available information about the motion of the node could be used to define the mobility model. This paper proposes a very general mobility model, where only the maximal velocity of the node v_{\max} is assumed to be known. Then, the positions of the generic node at time steps $t - \Delta t$ and t satisfy

$$(x_1(t) - x_1(t - \Delta t))^2 + (x_2(t) - x_2(t - \Delta t))^2 \leq \Delta t^2 \cdot v_{\max}^2. \quad (3)$$

More generally, the mobility model could be reformulated as follows:

$$\mathbf{f}(\mathbf{x}(t - \Delta t), \mathbf{x}(t), \nu) = 0, \quad (4)$$

where ν is some parameter only known to belong to some known interval $[\nu]$ (here, $[\nu] = [0, v_{\max}]$).

2.3. Description of the Robust Set-Membership Localization. Estimating the location of the sensor at time t consists of finding the set $\mathbb{X}(t)$ of all locations consistent with the mobility model (4) and at least $|I(t)| - q$ observation constraints (2). In other words, these locations should be in the vicinity of at least $|I(t)| - q$ detected anchors. A set-membership estimator [23] robust to q outliers [24] at time t is then obtained, since any $|I(t)| - q$ measurements instead of $|I(t)|$ measurements are considered for the estimator.

Assume that $\mathbf{x}(t - \Delta t)$ belongs to some set $\mathbb{X}(t - \Delta t)$. According to this approach, and to compute $\mathbb{X}(t)$, a *predicted* set is first evaluated using the mobility model

$$\mathbb{X}^*(t) = \{\mathbf{x} \mid \exists \mathbf{x}' \in \mathbb{X}(t - \Delta t), \exists \nu \in [\nu], \mathbf{f}(\mathbf{x}', \mathbf{x}, \nu) \leq 0\}. \quad (5)$$

Measurements are then taken into account to correct $\mathbb{X}^*(t)$ as follows:

$$\mathbb{X}(t) = \bigcup_{C \in \mathbf{C}_{I(t)}^{|I(t)|-q}} \left(\bigcap_{\ell \in C} \mathbb{X}_\ell(t) \right), \quad (6)$$

where $\mathbf{C}_{I(t)}^{|I(t)|-q}$ is the set of all $(|I(t)| - q)$ -combinations of indices in $I(t)$, C is a set of indices belonging to $\mathbf{C}_{I(t)}^{|I(t)|-q}$, and

$$\mathbb{X}_\ell(t) = \{\mathbf{x} \in \mathbb{X}^*(t) \mid \|\mathbf{x} - \mathbf{a}_\ell(t)\| \leq r\}. \quad (7)$$

Then, $\bigcap_{\ell \in C} \mathbb{X}_\ell(t)$ denotes the set of locations of $\mathbb{X}^*(t)$ that satisfy the specific observation constraints denoted in C , whereas $\mathbb{X}(t)$ denotes the set of locations of $\mathbb{X}^*(t)$ that satisfy any $(|I(t)| - q)$ observation constraints. The number of combinations to be considered in (6) is $K(t) = |I(t)|! / (|I(t)| - q)!$, where $q!$ denotes the factorial of q .

An alternative definition of $\mathbb{X}(t)$ inspired by [24] is

$$\mathbb{X}(t) = \left\{ \mathbf{x} \in \mathbb{X}^*(t) \mid \sum_{i \in I(t)} \lambda(\mathbf{x}, \mathbf{a}_i(t)) \geq |I(t)| - q \right\}, \quad (8)$$

where

$$\lambda(\mathbf{x}, \mathbf{a}_i(t)) = \begin{cases} 1, & \text{if } \|\mathbf{x} - \mathbf{a}_i(t)\| \leq r, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

This definition does not involve any combinatorial. One may easily prove that (6) and (8) are equivalent. This technique would be used in the following to solve the localization problem.

3. Localization Algorithms

Solving the localization problem in a guaranteed way consists of finding the set of all node locations that satisfy the problem constraints while being robust to outliers. In this paper, interval analysis [15] is employed to achieve this goal. At each time step, the proposed method computes a set of nonoverlapping boxes, called *subpaving* [15], whose union covers the solution set $\mathbb{X}(t)$. Assume that $[\mathbb{X}](t)$ is the solution subpaving containing the actual position of the generic node at time t . One has

$$[\mathbb{X}](t) = \bigcup_{1 \leq j \leq n(t)} [\mathbf{x}_j](t), \quad (10)$$

where $[\mathbf{x}_j](t) = [x_{j,1}](t) \times [x_{j,2}](t)$ is a two-dimensional box and $n(t)$ is the number of boxes in $[\mathbb{X}](t)$. As shown in Section 2.3, finding $[\mathbb{X}](t)$ involves a prediction phase followed by a correction phase.

3.1. Prediction Phase. Assume that $[\mathbb{X}](t - \Delta t)$ is the subpaving obtained at time $t - \Delta t$. Computing the predicted set $[\mathbb{X}]^*(t)$ may be done by evaluating (5), where $\mathbb{X}(t - \Delta t)$ is replaced by $[\mathbb{X}](t - \Delta t)$, which is quite difficult. Nevertheless, for each box $[\mathbf{x}_j](t - \Delta t) = [x_{j,1}](t - \Delta t) \times [x_{j,2}](t - \Delta t) \in [\mathbb{X}](t - \Delta t)$, the corresponding box $[\mathbf{x}_j]^*(t) = [x_{j,1}]^*(t) \times [x_{j,2}]^*(t) \in [\mathbb{X}]^*(t)$ has to be compliant with the mobility model (3), leading to the following constraint:

$$\begin{aligned} & \left([x_{j,1}]^*(t) - [x_{j,1}](t - \Delta t) \right)^2 \\ & + \left([x_{j,2}]^*(t) - [x_{j,2}](t - \Delta t) \right)^2 \subseteq [0, \Delta t^2 \cdot v_{\max}^2]. \end{aligned} \quad (11)$$

Relaxing (11) yields the following expressions of $[x_{j,1}]^*(t)$ and $[x_{j,2}]^*(t)$:

$$\begin{aligned} [x_{j,i}]^*(t) &= [x_{j,i}](t - \Delta t) \\ &+ [-\Delta t \cdot v_{\max}, \Delta t \cdot v_{\max}], \quad i = 1, 2. \end{aligned} \quad (12)$$

Let $[\tilde{\mathbb{X}}]^*(t) = \bigcup_j [\mathbf{x}_j]^*(t)$ be the set of all boxes evaluated with (12). One may prove that $[\mathbb{X}]^*(t) \subset [\tilde{\mathbb{X}}]^*(t)$. The convex hull $[\mathbf{x}]^*(t) = [x_1]^*(t) \times [x_2]^*(t)$ [15] of $[\tilde{\mathbb{X}}]^*(t)$ is the smallest box containing $[\tilde{\mathbb{X}}]^*(t)$. Its components are defined as

$$[x_i]^*(t) = \left[\min_{j \leq n(t-1)} (\underline{\mathbf{x}}_{j,i}^*(t)), \max_{j \leq n(t-1)} (\bar{\mathbf{x}}_{j,i}^*(t)) \right], \quad i = 1, 2, \quad (13)$$

where $\underline{\mathbf{x}}_{j,1}^*(t)$ and $\bar{\mathbf{x}}_{j,1}^*(t)$ are the low and high endpoints of $[x_{j,1}]^*(t)$, respectively. $[\mathbf{x}]^*(t)$ is a rectangular area that

covers all possible locations that could be taken by the node at time t according to its mobility model. The convex hull is used in the correction phase instead of $[\tilde{\mathbf{X}}]^*(t)$ to reduce the computational complexity.

3.2. Correction Using the SIVIA Algorithm. The SIVIA algorithm [15] performs a succession of bisections and selections of boxes compliant with the localization constraints. Let $[\mathbf{x}]$ be a box, set initially to $[\mathbf{x}]^*(t)$. The following test function is evaluated:

$$\gamma([\mathbf{x}]) = \begin{cases} 1 & \text{if } \sum_{i \in I(t)} \lambda([\mathbf{x}], \mathbf{a}_i) \geq |I(t)| - q, \\ -1 & \text{if } \sum_{i \in I(t)} \tilde{\lambda}([\mathbf{x}], \mathbf{a}_i) > q, \\ 0 & \text{otherwise,} \end{cases} \quad (14)$$

where $\lambda([\mathbf{x}], \mathbf{a})$ is equal to 1 if $\sup(\|\mathbf{x} - \mathbf{a}\|) \leq r$ and 0 otherwise, while $\tilde{\lambda}([\mathbf{x}], \mathbf{a})$ is equal to 1 if $\inf(\|\mathbf{x} - \mathbf{a}\|) > r$ and 0 otherwise. Graphically, $\lambda([\mathbf{x}], \mathbf{a}) = 1$ means that the box $[\mathbf{x}]$ is entirely included in the connectivity disk centered on the anchor \mathbf{a} , whereas $\tilde{\lambda}([\mathbf{x}], \mathbf{a}) = 1$ means that the box $[\mathbf{x}]$ is entirely outside the connectivity disk centered on the anchor \mathbf{a} .

The box $[\mathbf{x}]$ is added to the solution $[\mathbb{X}](t)$ if $\gamma([\mathbf{x}]) = 1$, meaning that all $\mathbf{x} \in [\mathbf{x}]$ satisfy at least $|I(t)| - q$ observation constraints. Boxes inconsistent with more than q observation constraints ($\gamma([\mathbf{x}]) = -1$) are withdrawn, whereas others having a nonempty intersection with the solution set ($\gamma([\mathbf{x}]) = 0$) are bisected. The box $[\mathbf{x}]$ is bisected into two subboxes of equal area $[\mathbf{x}]_1$ and $[\mathbf{x}]_2$ along the dimension having the largest width. The subboxes are then tested, kept in the solution, withdrawn, or bisected until the maximal width of the resulting subboxes is less than a given threshold δ . An illustration of the proposed method is given in Figure 1. It shows four detected anchors, one of them being an outlier ($q = 1$). The exact solution of the problem is given in light gray, whereas the subpaving provided by SIVIA is given in both light and dark gray.

3.3. Correction Using the Combinatorial Technique. In this algorithm, both the combinatorial formulation (6) and the convex hull $[\mathbf{x}]^*(t)$ including all propagated boxes using (13) are considered. Based on (6), the proposed algorithm consists of contracting the initial domain $[\mathbf{x}]^*(t)$ with each combination $C \in \mathbf{C}_{I(t)}^{|I(t)|-q}$ of observations. For this reason, all observation equations indicated in C are iterated in the *forward-backward* contractor [15]. This contractor iterates all constraints without any prior order until no contraction is possible. The resulting region is the smallest box covering the intersection of $[\mathbf{x}]^*(t)$ with all the observation disks of C . In order to use each constraint of (2) in the forward-backward contractor, one should express $x_1(t)$ as a function of $x_2(t)$, and vice versa as follows:

$$\begin{aligned} a_{i,1}(t) - b_{i,1}(t) \leq x_1(t) \leq a_{i,1}(t) + b_{i,1}(t), \\ a_{i,2}(t) - b_{i,2}(t) \leq x_2(t) \leq a_{i,2}(t) + b_{i,2}(t), \end{aligned} \quad (15)$$

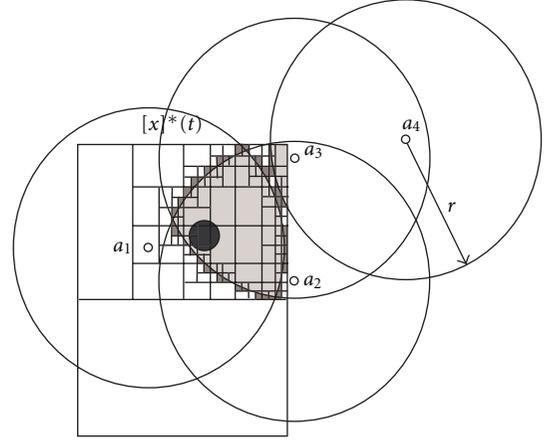


FIGURE 1: Robust localization with the SIVIA algorithm.

```

while  $[\mathbf{x}]_C$  is contracted do
  for  $i \in C$  do
     $\bar{b}_{i,1}(t) = \sup([\sqrt{r^2 - ([x_2]_C - a_{i,2}(t))^2}]$ ;
     $[x_1]_C = [x_1]_C \cap [a_{i,1}(t) - \bar{b}_{i,1}(t), a_{i,1}(t) + \bar{b}_{i,1}(t)]$ ;
     $\bar{b}_{i,2}(t) = \sup([\sqrt{r^2 - ([x_1]_C - a_{i,1}(t))^2}]$ ;
     $[x_2]_C = [x_2]_C \cap [a_{i,2}(t) - \bar{b}_{i,2}(t), a_{i,2}(t) + \bar{b}_{i,2}(t)]$ ;
  end
end

```

ALGORITHM 1: Computation of the contracted box using the forward-backward contractor.

for each detected anchor i where $b_{i,1}(t) = \sqrt{r^2 - (x_2(t) - a_{i,2}(t))^2}$ and $b_{i,2}(t) = \sqrt{r^2 - (x_1(t) - a_{i,1}(t))^2}$. Using intervals and having an initial box $[\mathbf{x}]$, these inequalities would lead to the contracted box $[\mathbf{x}']$ defined as follows:

$$\begin{aligned} [x'_1] &= [a_{i,1}(t) - \bar{b}_{i,1}(t), a_{i,1}(t) + \bar{b}_{i,1}(t)], \\ [x'_2] &= [a_{i,2}(t) - \bar{b}_{i,2}(t), a_{i,2}(t) + \bar{b}_{i,2}(t)], \end{aligned} \quad (16)$$

where $\bar{b}_{i,1}(t) = \sup([\sqrt{r^2 - ([x_2] - a_{i,2}(t))^2}]$ and $\bar{b}_{i,2}(t) = \sup([\sqrt{r^2 - ([x_1] - a_{i,1}(t))^2}]$. Then, considering the combination C of constraints, and starting with the predicted domain (initially $[\mathbf{x}]_C = [\mathbf{x}]^*(t)$), the contracted box $[\mathbf{x}]_C$ would be obtained by performing the following steps (Algorithm 1).

Each nonempty box $[\mathbf{x}]_C$, $C \in \mathbf{C}_{I(t)}^{|I(t)|-q}$, is added to the solution $[\mathbb{X}](t)$ at time t . The boxes in $[\mathbb{X}](t)$ may have nonempty intersections. Consequently, in order to obtain nonoverlapping boxes, one could apply the following procedure.

- (i) Consider an empty final set.
- (ii) Sort all boxes of $[\mathbb{X}](t)$ according to their decreasing areas.

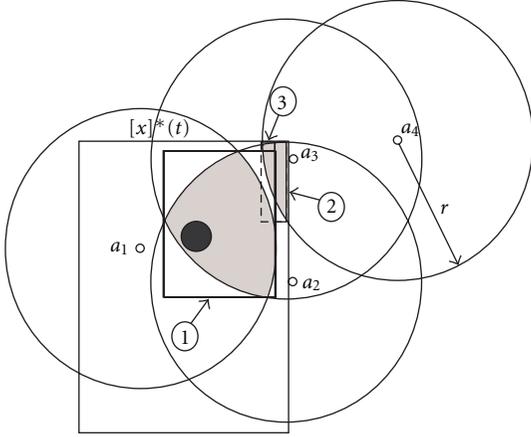


FIGURE 2: Robust localization with the combinatorial technique.

- (iii) Add the largest box to the final set.
- (iv) Select the following box in the sorted list.
- (v) Deprive it from all boxes already added to the final set.
- (vi) Add the result to the set.

Steps (iv) to (vi) are repeated until all sorted boxes are considered. Recall that depriving a box $[x]$ from a box $[y]$ yields a set of nonoverlapping boxes covering all the points \mathbf{x} of $[x]$ not included in $[y]$. An illustration of the proposed method is given in Figure 2. It shows four detected anchors, one of them yielding an erroneous observation. The first solution leads to two boxes, the box ① and the one in dashed line. Using the depriving technique, three nonoverlapping boxes ①, ②, and ③ (in bold line) are then selected covering the exact solution (in light gray).

3.4. Evaluation of the Number of Outliers to Tolerate. Considering the vector $\boldsymbol{\rho}$ of RSS measurements. Determining from $\boldsymbol{\rho}$ the maximal number of outliers q , which have to be tolerated, may be done by choosing q such that $\Pr(Q \leq q | \boldsymbol{\rho}) > 1 - \nu$, where $\Pr(Q \leq q | \boldsymbol{\rho})$ is the probability that q or less outliers have occurred knowing $\boldsymbol{\rho}$ and $\nu \in [0, 1]$ is some tuning parameter. One has

$$\begin{aligned} \Pr(Q \leq q | \boldsymbol{\rho}) &= \sum_{k=0}^q \Pr(Q = k | \boldsymbol{\rho}) \\ &= \sum_{k=0}^q \sum_{\mathbf{s} \in \{0,1\}^{N_a}, \sum_i s_i = k} \Pr(Q = k, \mathbf{s} | \boldsymbol{\rho}), \end{aligned} \quad (17)$$

where \mathbf{s} is some pattern indicating whether anchor $i = 1, \dots, N_a$ is providing an outlier ($s_i = 1$) or a reliable measurement ($s_i = 0$). Now,

$$\Pr(Q = k, \mathbf{s} | \boldsymbol{\rho}) = \Pr(Q = k | \mathbf{s}, \boldsymbol{\rho}) \cdot \Pr(\mathbf{s} | \boldsymbol{\rho}). \quad (18)$$

Let ρ_i and ρ_i^* be the noisy and noiseless RSS measurements provided by the i th anchor. The probability p_i that the i th

anchor provides an outlier is null ($p_i = 0$) if $\rho_i < \rho_r$, since in this case, the anchor i is not detected and will thus not provide any outlier. Otherwise, p_i is given as follows:

$$\begin{aligned} p_i &= \Pr(\rho_i^* < \rho_r | \rho_i \geq \rho_r) = \Pr(\rho_i - \rho_i^* > \rho_i - \rho_r) \\ &= \Pr(\varepsilon_i > \rho_i - \rho_r) \\ &= \frac{1}{2} \left(1 - \operatorname{erf} \left(\frac{\rho_i - \rho_r}{\sqrt{2\sigma^2}} \right) \right), \end{aligned} \quad (19)$$

where ε_i is the i th measurement noise. Then, assuming that all measurement noise samples are independent,

$$\Pr(\mathbf{s} | \boldsymbol{\rho}) = \prod_{i=1}^{N_a} (p_i s_i + (1 - p_i)(1 - s_i)). \quad (20)$$

Now, since

$$\Pr(Q = k | \mathbf{s}, \boldsymbol{\rho}) = \Pr(Q = k | \mathbf{s}) = \begin{cases} 1, & \text{if } \sum_i s_i = k, \\ 0, & \text{otherwise,} \end{cases} \quad (21)$$

one is able to evaluate $\Pr(Q \leq q | \boldsymbol{\rho})$ and to choose q .

4. Simulations

In this section, we compare the SIVIA-based method (SBL) to the combinatorial-based one (CBL). We consider a group trajectory model, where sensors are moving along similar trajectories over 100 s. We deploy 31 sensors in a 100 m \times 100 m area, 30 of them being anchors. Since nodes use only anchors information to localize themselves, we consider the localization of a single mobile node. The simulated trajectory of the node is with a maximal velocity of 2.3 m \cdot s⁻¹ and a localization step Δt of 1 s. RSS measurements are generated using the distances between the considered node and all anchors and model (1) with $\rho_0 = 100$ dBm, $d_0 = 1$ m, and $n_p = 4$. Moreover, r is set to 10 m and ρ_r to 60 dBm. In order to compare the SBL algorithm to the CBL algorithm, we take different values of the variance σ^2 of the measurement noise, leading to different numbers of outliers. In fact, for each value of σ^2 , q is evaluated using the results in Section 3.4, with $\nu = 0.1$.

Note that the initial position of the node is supposed to be known. One may also use the whole deployment area as initial domain. All simulations are performed on an Intel(R) Core(TM)2 CPU at 2.40 GHz and 1 GB RAM, using MATLAB 6.1.

We first set $\sigma = 3$ dBm, yielding either none or only one outlier per time step. Applying the results of Section 3.4 one obtains $\Pr(q = 0) = 0.4161$, $\Pr(q \leq 1) = 0.9109$, and $\Pr(q \leq 2) = 0.9958$. With $\nu = 0.1$, one gets $q = 1$. With $\nu = 0.01$, one would take $q = 2$, which leads to less accurate estimates (less measurements are taken into account) but still containing the solution set. Note that if too less outliers are tolerated, one may obtain empty solution sets or sets not containing the actual location of the node.

With $q = 1$, Figure 3 shows the subpavings obtained with both SBL and CBL methods. Note that the threshold

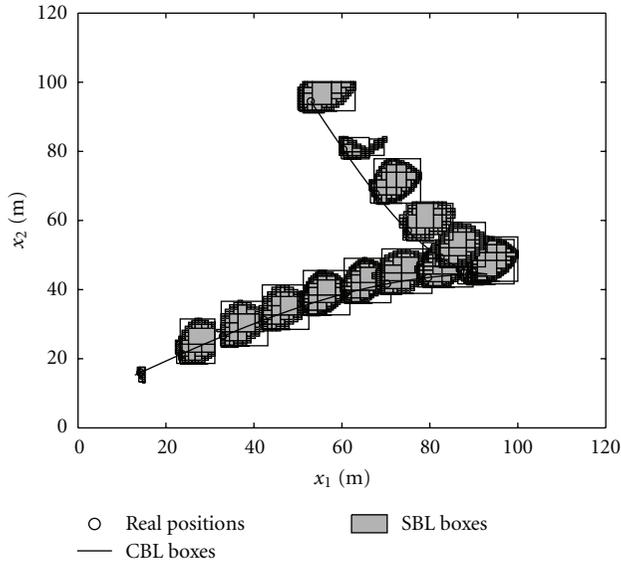


FIGURE 3: An illustration of the subpavings obtained with the SBL and the CBL algorithms.

δ of SIVIA is set to 1 m. The plot shows that both results cover the actual position of the node. The average ratio of the areas of subpavings obtained with SBL over those obtained with CBL is equal to 0.827. SBL leads to a more accurate estimate. However, the average time required for the localization process is equal to 0.471 s per time step with SBL and to 0.051 s with CBL. This difference is due to the limited number of combinations considered in CBL at each time step ($6 \leq |I| \leq 10$ with $q = 1$). The average number of boxes per subpaving is equal to 120 with SBL, whereas it is equal to 3 with CBL. Here, CBL is less memory consuming. Note that with a precision parameter higher than 1 m in SIVIA, SBL needs less computing time but provides larger subpavings.

In a second set of experiments, σ varies from 1 dBm to 15 dBm. Figure 4 shows the maximal number of outliers q , the total number of considered anchors $|I|$, the average computing time per step, and the ratio of the average subpaving areas obtained with SBL over CBL as a function of σ . The simulated data are generated ten times for each σ , and the results are thus average values over the set of simulations. The plot shows that CBL is faster than SBL when the standard deviation σ is less than 8 dBm. In these cases, q is less than 4, and the maximal number of considered anchors is less than 13. When the noise variance increases, the computation time of the CBL method becomes quite large compared to SBL. Choosing one algorithm or the other depends on the anchor density and on the proportion of outliers.

5. Conclusion

This paper proposes and compares two techniques for mobile sensor localization that are robust to any fixed number of erroneous measurements. Using interval analysis, the estimates are sets of nonoverlapping boxes containing the actual location. The SIVIA-based algorithm (SBL) bisects the

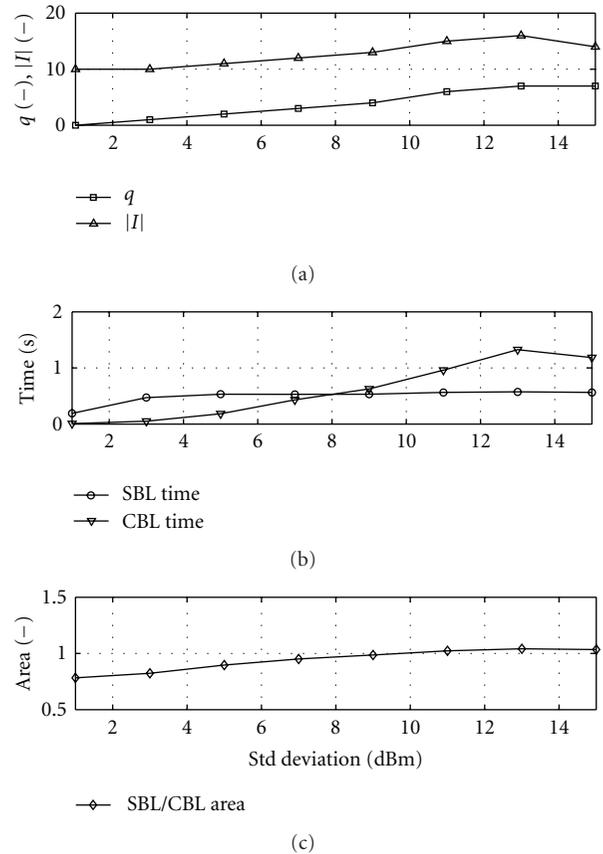


FIGURE 4: An illustration of q and $|I|$ at (a), the computing times are in (b) and the ratio of the subpavings areas (SBL over CBL) in (c).

search region leading to many boxes describing efficiently the solution set; the combinatorial method (CBL) leads to larger boxes including the solution as well. In terms of computing time, CBL is more efficient than SBL for a small number of outliers, whereas the complexity of SBL is almost constant whatever the number of tolerated outliers.

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] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*, Springer, New York, NY, USA, 1994.
- [3] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1655–1663, Anchorage, Alaska, USA, April 2001, <http://www.citeulike.org/group/7128/article/3014236>.
- [4] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-free positioning in mobile ad-hoc networks," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34 '01)*, pp. 3481–3490, Maui, Hawaii, January 2001.
- [5] D. Niculescu and B. Nath, "Ad-hoc positioning system (APS)," in *Proceedings of the IEEE Global Telecommunications*

- Conference (GLOBECOM '01), vol. 5, pp. 2926–2931, San Antonio, Tex, USA, 2001, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=965964>.
- [6] C. Savarese, J. M. Rabaey, and J. Beutel, “Locating in distributed ad-hoc wireless sensor networks,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, vol. 4, pp. 2037–2040, Salt Lake City, Utah, USA, May 2001, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=940391>.
- [7] D. Moore, J. Leonard, D. Rus, and S. Teller, “Robust distributed network localization with noisy range measurements,” in *Proceedings of the Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 50–61, ACM, Baltimore, Md, USA, November 2004, <http://doi.acm.org/10.1145/1031495.1031502>.
- [8] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Patten, “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, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1307324&tag=1.
- [9] D. Blatt and A. O. Hero, “APOCS: a rapidly convergent source localization algorithm for sensor networks,” in *Proceedings of the IEEE/SP 13th Workshop on Statistical Signal Processing*, pp. 1214–1219, July 2005, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1628781.
- [10] L. Hu and D. Evans, “Localization for mobile sensor networks,” in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, pp. 45–57, ACM, Philadelphia, Pa, USA, 2004, <http://doi.acm.org/10.1145/1023720.1023726>.
- [11] A. Baggio and K. Langendoen, “Monte Carlo localization for mobile wireless sensor networks,” *Ad Hoc Networks*, vol. 6, no. 5, pp. 718–733, 2008.
- [12] Y. Jiyoung, Y. Sungwon, and C. Hojung, “Multi-hop-based Monte Carlo localization for mobile sensor networks,” in *Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '07)*, pp. 162–171, San Diego, Calif, USA, June 2007, <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04292828>.
- [13] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [14] M. Terwilliger, C. Coullard, and A. Gupta, “Localization in ad hoc and sensor wireless networks with bounded errors,” in *Proceedings of the 15th International Conference on High Performance Computing (HiPC '08)*, pp. 295–308, Springer, Bangalore, India, 2008, <http://portal.acm.org/citation.cfm?id=1791889.1791922>.
- [15] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*, Springer, 2001.
- [16] F. Mourad, H. Snoussi, F. Abdallah, and C. Richard, “Anchor-based localization via interval analysis for mobile ad-hoc sensor networks,” *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 3226–3239, 2009.
- [17] F. Mourad, H. Snoussi, F. Abdallah, and C. Richard, “Model-free interval-based localization in manets,” in *Proceedings of the IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop (DSP/SPE '09)*, pp. 474–479, Marco Island, Fla, USA, January 2009.
- [18] L. Jaulin, E. Walter, and O. Didrit, “Guaranteed robust nonlinear parameter bounding,” in *Proceedings of the IMACS Multiconference: Symposium on Modelling, Analysis and Simulation (CESA '96)*, pp. 1156–1161, Lille, France, 1996.
- [19] C. Savarese, J. M. Rabaey, and K. Langendoen, “Robust positioning algorithms for distributed ad-hoc wireless sensor networks,” in *Proceedings of the General Track of the Annual Conference on USENIX Annual Technical Conference*, pp. 317–327, 2002, <http://portal.acm.org/citation.cfm?id=647057.713854>.
- [20] M. Rabbat, R. Nowak, and J. Bucklew, “Robust decentralized source localization via averaging,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, pp. V1057–V1060, March 2005.
- [21] J. Leger and M. Kieffer, “Guaranteed robust distributed estimation in a network of sensors,” in *Proceedings of the 35th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '10)*, pp. 3345–3381, Dallas, Tex, USA, March 2010, http://www.lss.supelec.fr/~publi/TWlJaGVsIEtJRUZGRVI=_ICASSP-Leger_10.pdf.
- [22] A. Medeisis and A. Kajackas, “On the use of the universal Okumura-Hata propagation prediction model in rural areas,” in *Proceedings of the 51st Vehicular Technology Conference (VTC '00)*, pp. 1815–1818, Tokyo, Japan, May 2000.
- [23] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, Eds., *Bounding Approaches to System Identification*, Plenum Press, New York, NY, USA, 1996.
- [24] H. Lahanier, E. Walter, and R. Gomeni, “OMNE: a new robust membership-set estimator for the parameters of nonlinear models,” *Journal of Pharmacokinetics and Biopharmaceutics*, vol. 15, no. 2, pp. 203–219, 1987.

Research Article

Ion-6: A Positionless Self-Deploying Method for Wireless Sensor Networks

Shih-Chang Huang

Department of Computer Science and Information Engineering, National Formosa University, YunLin, Taiwan

Correspondence should be addressed to Shih-Chang Huang, schuang@nfu.edu.tw

Received 15 June 2011; Accepted 27 July 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Shih-Chang Huang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sensor networks are deployed to monitor the interested area. Maximizing the sensing coverage depends on effectively determining the locations of sensors. In this paper, a self-deploying method named as Ion-6 is proposed. Sensors are modeled as ions, and the links between them are treated as ionic bonds. When the number of ionic bonds of a sensor is full, the sensor will expel others out of its field. Sensors organize themselves as the hexagonal format to maximize the network's coverage area, retain the network connectivity, and prevent from introducing the coverage holes. The sensors in the proposed method can compute their moving directions and distances independently without priori position information. Simulation results show that Ion-6 can organize sensors as the hexagonal format to maximize the coverage area. The deploying time of Ion-6 is less than the molecule model and efficiently eliminates the unnecessary movements of the virtual force (V-force) method.

1. Introduction

A sensor network consists of a large number of tiny devices, which can detect the variances of environment, execute simple computations, and use wireless communication to exchange information. Those devices are deployed over the interested regions to collect the environment information and help people to monitor the incidents.

Sensors can be deployed factitiously [1–6] or be randomly spread over the interested regions. However, in the applications to monitor the harsh terrain or the hostile environments, it is almost impossible to deploy sensors by human beings, for example, detect the radiation zone, monitor the noxious gas leaking area, trace the frontline battlefield, and explore unknown planets.

Spreading the sensors randomly cannot guarantee the precise settle-down location of each sensor. Sensors may cluster overly in a small region or may distribute too sparsely to retain the network connectivity. Those two possible unfavorable results cause the sensors losing the efficient surveillance on the environment. Therefore, the sensors with the mobile ability are used to adjust their positions after randomly spreading.

The deploying problem will become trivial if each sensor can obtain both the mobile ability and its position information [7–10]. Each sensor's final location can be scheduled to maximize the coverage area and minimize the moving distance. However, when the position information is not available, getting well coverage area and shortening the moving distance during the self-deploying will become a great challenge. Thus, some of the previous researches organize sensors into the cluster architecture [11–13]. A small set of mobile sensors are selected as the local controllers to determine the others' locations. In this architecture, the sensor-deploying problem is simplified to organize sensors in every cluster. The critical issue of this architecture is that it has the reliability problem when the cluster heads out of function.

Some previous works model the mobile sensors as the electrons [14, 15] or molecules [16, 17] to avoid the fault of cluster architecture. The received signal strength (RSS) of this message is treated as the force which pushes each other. The deploying procedure finishes when the forces work on every sensor are balanced. Sensors in this model may have *oscillation moving* that sensors move back and forth over a small region to adjust their positions before the force

becomes balance. It is not energy efficiency for the energy-limited sensors.

In this paper, we model the deploying problem as building the ionic bonds between ions. Sensors are ions, and the built links between them are the ionic bonds. Sensors do not need to have their position information. They only require the abilities to identify the direction of incoming signals and accurately estimate their distance to the other neighbors. These are two essential abilities in general self-deploying methods. The proposed method can effectively maximize the coverage and minimize the deploying time.

The rest of this paper is organized as follows. The related positioning systems are reviewed in Section 2. The proposed method is given in Section 3. In addition, this section also discusses how to adjust the deploying location when sensors do not move to their proper locations. In Section 4, we give the simulation results to prove that the proposed Ion-6 method works well. Finally, the paper conclusions are in Section 5.

2. Related Works

For the mobile sensors, the self-deploying methods have two classes. The first class is position-based methods which mobile sensors have the prior knowledge of their location information [7–10]. When sensors can get their position information, their positions can be prescheduled before they are cast over the interested region. The average moving distance of sensors can be minimized, and the coverage area can be maximized. Therefore, the object of position-based methods is to maximize the coverage area [8], minimize the moving distance [9], and eliminate the coverage holes [18, 19]. Jourdan et al. use the genetic algorithm to solve the deployment optimization problem [19]. To speed up the computational convergence of the method in [20], Xiaoling et al. use the particle swarm optimization (PSO) problem to model the self-deploying problem. Sensors are restricted to move in the limited region to save energy consumed during the sensor moving process. A similar method is proposed in [7] but involves the obstacles in the interested region. These methods divide the interested region into multiple regular grid areas. In [10], Li et al. considered the distance and orientation of every sensor to find its deploying location. The position information is necessary for this method to organize the sensors as a hexagon format.

The second class is positionless that mobile sensors do not have their position information. The simplest positionless method is organizing sensors into multiple clusters [11]. In each cluster, a mobile sensor takes charge to determine the others' locations. This mobile sensor is named *cluster header*. Sensors in each cluster will organize as the star topology that the cluster head is the center. Due to the fact that geography may greatly increase the difficulty on placing and electing the cluster heads, determining the cluster heads is the major challenge in this method. In additional, when cluster head crashes, a recovery mechanism [13, 14] is needed to retain the reliability of network.

Another positionless method is to model the mobile sensors as the molecules [12, 16, 17]. Each sensor has to

broadcast a dummy message periodically to announce its existence. By collecting the dummy message, each sensor can know how many neighbors it has. The number of neighbors implies the density of its neighborhood. Sensors will move step by step from high-density area to a low-density one. The moving direction can be computed from the direction of the incoming signal. Because the sensors use small step to adjust their next position, the deployment process usually consumes a lot of time. Mobile sensors also waste much additional energy on exchanging messages.

To deploy the sensors quickly, sensors are modeled as the electrons [5, 14, 15]. This model is called the Virtual Force (V-force) method. Similar to the molecule model, sensors have to send the dummy message. The receivers will transform the signal strength of the incoming message as the force. When the received message gives strong signal strength, a large force is created; otherwise, the force is small. The receiver computes the net forces and determines its moving direction and distance. Sensors continuously move until the value of net forces is less than the threshold. Although this method removes the reliability problem of the cluster architecture, it has a severe problem on redundant movement, which is caused by the oscillation moving.

To reduce the energy consumed on oscillation moving and to maximize the coverage area in the positionless methods, we proposed a novel method which models the sensors as ions. Sensors in the proposed method can expand their coverage area better than both molecule model and the V-Force method. Besides, sensors can be deployed rapidly than the molecule model and can save much energy wasted on the redundant movement than the V-Force method. The detail is given in next section.

3. The Ion-6 Method

3.1. Preliminaries and Assumptions. Each sensor has the following characteristics which are the common characteristics in current self-deploying methods.

- (1) Each sensor has a unique identity. Sensors are randomly spread over the interested region.
- (2) Each sensor can communicate with others without losing data.
- (3) All sensors have the same communication ranges. The coverage area of each sensor is a circular disk. The sensing range is equal to the communication range.
- (4) Sensors can precisely estimate the Euclid's distance to the sender from the received signal strength of incoming packets. The Friis transmission formula [21] is used to convert the RSS to Euclid's distance. In the later, we will consider inaccurate estimation and show how sensors detect the fault and adjust their positions.
- (5) Each sensor installs a precise antenna array, which can identify the angle of every incoming packet. Each sensor also has a precise compass to determine its moving direction.

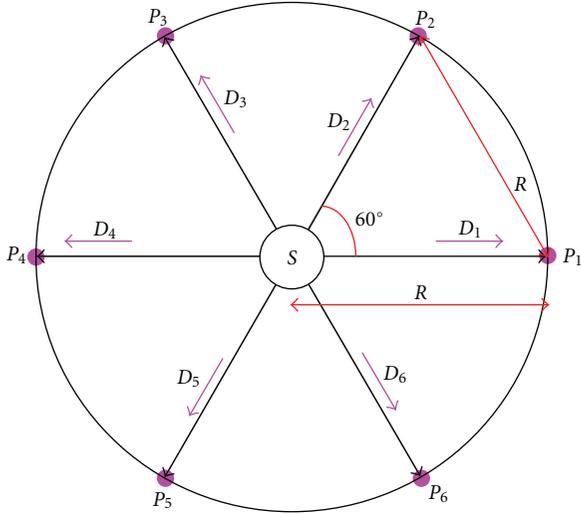


FIGURE 1: The six ionic bonds and stable slots of sensor S.

We have the following additional assumptions for our proposed method. The maximal number of neighbors which a sensor can have is defined as the number of *ionic bonds*. To organize the deploying topology as the hexagonal format, the ionic bonds of every sensor are set to six. The first sensor which starts the deploying procedure will define the direction of each ionic bond. The direction of these six ionic bonds will evenly divide the circular communication coverage into six same-sized sectors shown as Figure 1. All sensors follow the first sensor's decision and propagate this decision to others during the deploying procedure.

For each sensor, if there is a neighbor at the direction of its ionic bond I_i , denoted as D_i , and the distance between the neighbor and itself is equal to the sensing radius R , the ionic bond I_i is defined as a *stable ionic bond*. The location of this neighbor is called as the *stable slot* of ionic bond I_i , denoted as P_i . As Figure 1 shows, the directions of the six ionic bonds of sensor S are D_1, D_2, \dots, D_6 . The corresponding stable slots are P_1, P_2, \dots, P_6 . If a stable slot has obstacles in it such that the assigned neighbor cannot move to there, the sensor will remove this ionic bond.

3.2. The Ion-6 Method. Initially, all sensors are in passive mode waiting for combining with others. Because sensors have free ionic bonds, their state is *unsteady*. A random sensor S enters the active mode and starts the deploying procedure. Sensor S is called as *anchor*. The anchor S sets the default directions of the six ionic bonds and broadcasts to all neighbors via a *bond packet*. The directions of these ionic bonds are represented as unit vectors.

Let W be an unsteady sensor that can directly receive the bond packet from S , and its distance to S is $|SW|$. Sensor W uses the Frii's transmission formula [21] to compute the $|SW|$ from the RSS. We assume that $|SW| = d$. In addition, the incoming direction of the bond packet is \vec{V}_W shown as Figure 2. For each free ionic bond I_i in the bond packet,

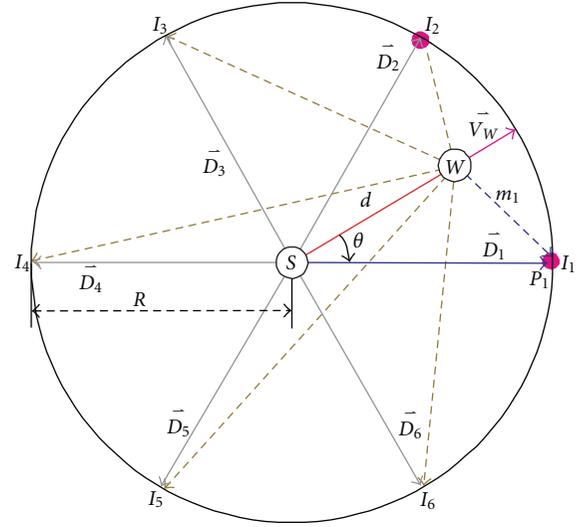


FIGURE 2: The directions of the six ionic bonds.

sensor W computes the distance m_i and direction \vec{U}_i to the corresponding stable slot P_i . Then, sensor W reports the results to S . Here, we denote the distance to stable slot P_i as m_i and the direction as \vec{U}_i .

By having the communication radius R and the distance $|SW| = d$, we can use the cosine law to compute the m_i

$$m_i = \sqrt{d^2 + R^2 - 2dR \cos \theta_i}. \quad (1)$$

The angle θ_i is the included angle of \vec{V}_W and \vec{D}_i . It can be obtained from the inner product of \vec{V}_W and \vec{D}_i

$$\theta_i = \cos^{-1} \left(\frac{\|\vec{V}_W \cdot \vec{D}_i\|}{\|\vec{V}_W\| \|\vec{D}_i\|} \right). \quad (2)$$

The moving direction \vec{U}_i can be computed from

$$\vec{U}_i = R \times \vec{D}_i - d \times \vec{V}_W. \quad (3)$$

After collecting the results from all neighbor sensors, anchor S instructs the sensor with minimal m_i to move to each P_i . These instructed sensors will change to active mode. We called them as *candidates*. After the candidates move to the stable slots, they park at the locations and notify S . Note that if a P_i has already been occupied by a sensor, S will not assign a candidate again. When all the ionic bonds become stable, S will expel all passive mode sensors out its sensing field. When S finished expelling the passive mode sensors, it changes to *lock* state. The lock state sensors will no longer move. All candidates become anchors and are notified to find its candidates. To prevent two adjacent anchor sensors from arranging two different candidates to their joint stable slots, only one anchor sensor is allowed to broadcast the bond packet at a time. The other anchor sensor must hold down

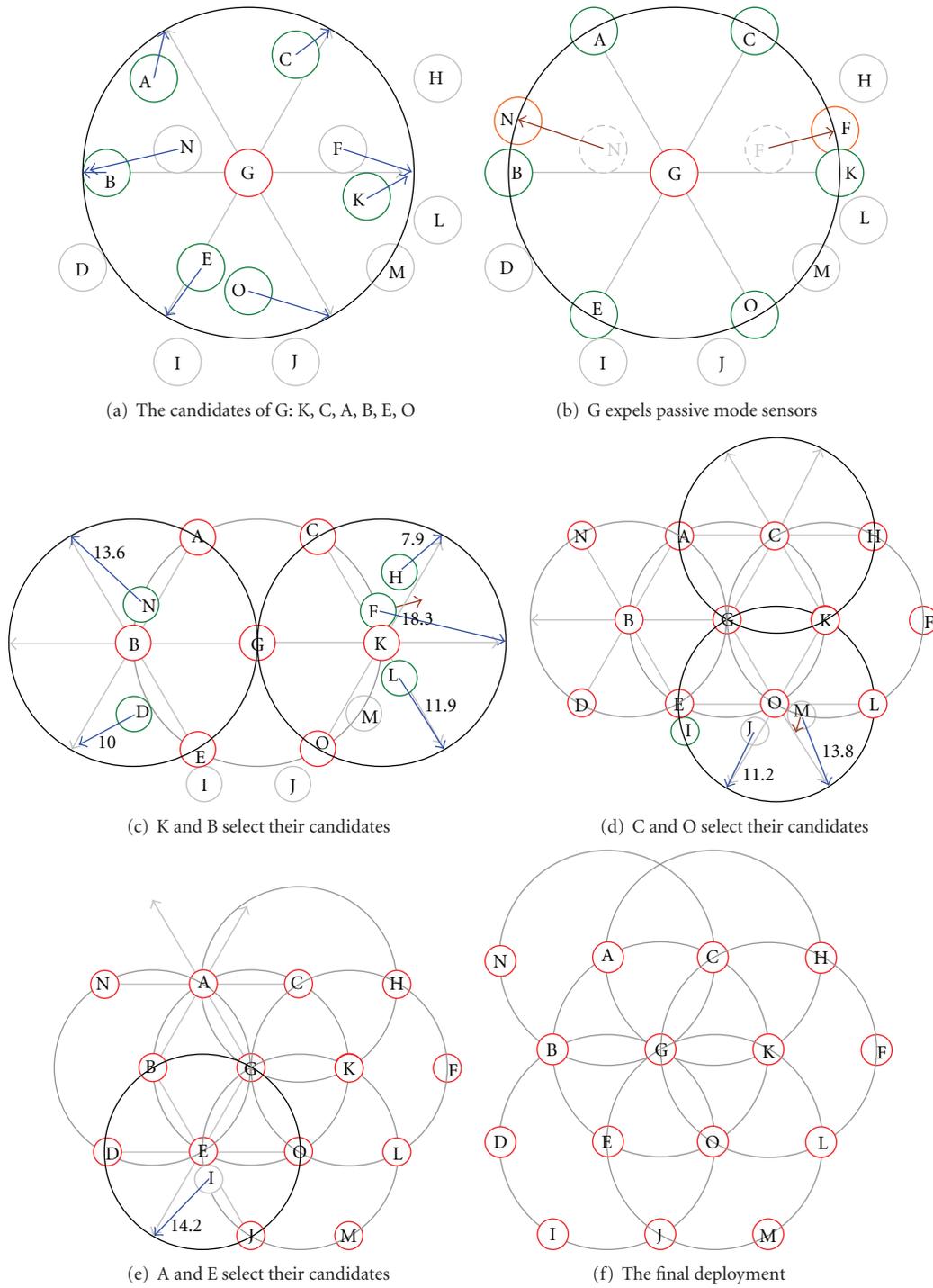


FIGURE 3: Deploying example.

until the candidate sensors of current anchor sensor move to their stable slots.

Furthermore, to prevent the expelled passive sensors from moving back to the sensing range of the lock state sensors, a level architecture is also built while determining the candidates. The first active sensor sets its level to 0. The candidates selected by active sensor X will set their level to $L_x + 1$, where L_x is the level of X . The active sensors in level

$N + 1$ can start to broadcast their bond packets when all active sensors in level N have found or not been able to find the candidate sensors. The deploying procedure stops until all sensors become the lock state.

Figure 3 gives a little example of the proposed method. Initially, sensors are deployed as Figure 3(a). The sensor G enters the active mode to broadcast the bond packet. G sets its level to 1 and determines the directions of the six ionic

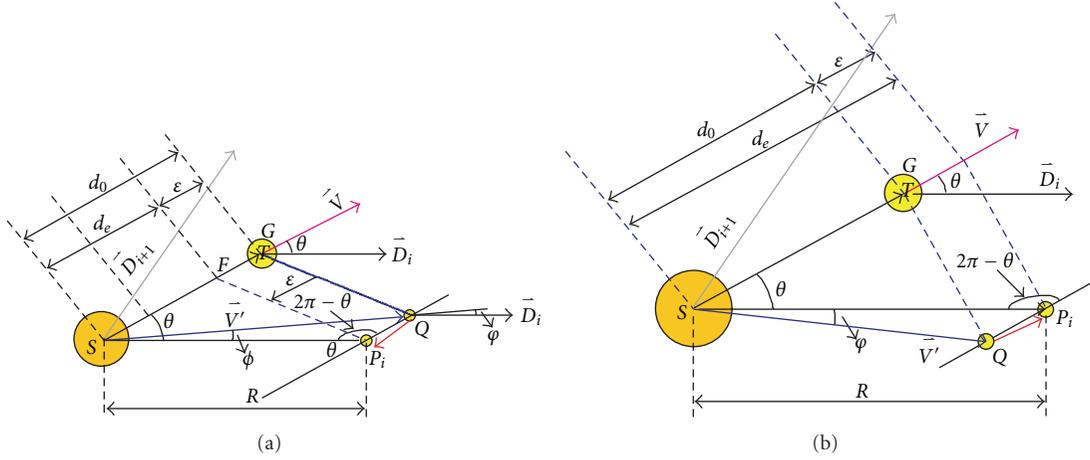


FIGURE 4: Distance adjusting mechanism.

bonds. The sensors $\{A, B, C, E, F, K, N, O\}$ receive the bond packet from G. They compute the m_i and \vec{U}_i to each stable slot of G. Sensor G selects the set $\Omega = \{K, C, A, B, E, O\}$ as its candidates after it collects the results from neighbors. The level information is also propagated to them. Sensors in set Ω set their level to 2, change to active mode, and start to move to the corresponding stable slots. When these six candidates reach the target locations, those passive mode sensors in G's sensing field will be expelled by G. The sensors N and F are expelled shown as Figure 3(b).

After G expels all passive mode sensors, it acknowledges all level 2 sensors to select their candidates. K and B are the first two sensors selected to choose their candidates shown as Figure 3(c). The level 2 candidates A, C, E, and O hold down their attempts to select candidates because the transmissions of K or B block their bond packets. Sensor K chooses H, F, and L as its candidates. Sensor B only chooses N and D because there are no other sensors. The sensors H, F, L, N, and D are set to level 3. When all the six ionic bonds of K become stable, sensor K starts to expel the passive mode sensor M out of its sensing coverage. So is the sensor B. The other level 2 candidates C, O and A, E are similar to K and B. Their results are shown as Figures 3(d) and 3(e).

After all sensors in level 2 have chosen their candidates, the sensors in level 3 will be notified to select their candidates. In this example, all level 3 sensors have no passive sensors in their sensing fields. Therefore, the deploying procedure terminates. Figure 3(f) shows the deploying results.

3.3. Adjusting the Location to Stable Slot. The accuracy distance from a sensor to each stable slot of the anchor is necessary for the Ion-6 method. However, the estimated distance may be shorter or longer than the real one in the practice case. In this section, we discuss the position adjusting mechanism to recover from inaccuracy distance estimation.

Let d_0 be the real distance between S and one of its candidate T, and d_e is the estimated distance where $d_0 > d_e$, shown as Figure 4(a). Before the candidate T moves, its position is at location G. However, candidate T thinks its location is at F because of the inaccuracy estimated distance. Candidate T uses the radius R, d_e , and includes angle of the vector \vec{V} and \vec{D}_i to compute the moving distance m_i | $m_i = |\overline{FP}| = |\overline{GQ}|$ and the moving direction $\vec{U}_i = |\overline{GQ}|$ to the stable slot P_i . By following the computed m_i and \vec{U}_i , the location of candidate T is at Q instead of stable slot P_i .

When candidate T reaches location Q, the angle of the incoming signal from S, \vec{V}' , is not aligned with \vec{D}_i . It indicates that candidate T has not moved to the stable slot P_i yet. The candidate T should adjust its location by moving toward the direction \vec{QP}_i with distance $\epsilon = |d_0 - d_e|$.

Let $\angle QSP_i$ be φ which can be computed from the inner product of \vec{V}' and \vec{D}_i . Because \overline{SG} is parallel to \vec{QP}_i , the $\angle SQP_i$ will be $\theta - \varphi$. And the ϵ can be calculated from

$$\epsilon = \frac{R}{\sin(\theta - \varphi)} \times \sin \varphi. \quad (4)$$

Furthermore, the adjusting direction \vec{QP}_i is $\vec{D}_i - \vec{V}'$. Due to the \overline{SG} and \vec{QP}_i are parallel, $\vec{QP}_i = -\vec{V}$.

For the case $d_0 < d_e$, shown as Figure 4(b), the position adjusting procedure is similar. The adjusting direction \vec{P}_iQ is \vec{V} . The difference is the angle for the sine law changes as the following:

$$\epsilon = \frac{R}{\sin \theta} \times \sin \varphi. \quad (5)$$

To determine $d_0 > d_e$ or $d_0 < d_e$, we need to compare the angle $\angle GSP_i$ and $\angle GSQ$. If $\angle GSP_i > \angle GSQ$, (4) is used; otherwise, (5) is used. The algorithm for the proposed deploying method is given in Algorithm 1.

Ω_q : the set of one-hop neighbors of sensor q .
 I_j : the j th ionic bond of a sensor q , where $j = 1, \dots, 6$.
 D_j : the direction of the j th ionic bond of a sensor q .
 P_j^z : the location of stable slot of the j th ionic bond of a sensor q .
 H_K : the location of sensor K .
 $L(S)$: the level of sensor S . The initial level of all sensors is infinite.
 G_i : the group of sensors in level i .
 V_S : incoming direction of received signal from sensor S .

- (1) Give a sensor S , sets its level $L(S)$ to 0.
 Sensor S determines the D_j of each I_j . D_j is a unit vector.
 Set $i = 0$.
- (2) If G_i is empty
- (3) Exit.
- (4) Else
- (5) For each sensor Z in G_i do {
- (6) Sensor Z broadcasts the D_j to sensor $X \mid X \in \Omega_Z$.
- (7) X computes the length and direction of $H_X P_j^z \mid j = 1, \dots, 6$.
- (8) Return the computed results to Z .
- (9) For each $P_j^z \mid j = 1, \dots, 6$, {
- (10) Z selects the sensor X'_j with $\min |H_{X'_j} P_j^z|$ as the candidate to the location P_j^z .
- (11) Skip the P_j^z , if {
- (12) (a) There is no selectable sensor in Ω_Z
- (13) (b) The location has occupied by a sensor not in passive mode
- (14) }
- (15) Sensor X'_j sets its level to $L(X'_j) = L(Z) + 1$, moves toward the direction $H_{X'_j} P_j^z$.
- (16) X'_j compares the V_Z and D_j when X'_j stops moving.
- (17) If $V_S \neq D_j$, X'_j adjusts its location until $V_S = D_j$.
- (18) Else X'_j notifies the Z that it reaches location P_j^z .
- (19) }
- (20) }
- (21) Sender Z expels the passive mode sensors when all six X'_j are ready.
- (22) Set $i = i + 1$. Redo step (2) to (21).

ALGORITHM 1: Algorithm of the proposed deploying approach.

4. Simulation Results

This section shows the simulation results. A C++ program is developed to evaluate the proposed Ion-6 method. We also implement the molecule model [16] and the V-force method [14] for comparison. The evaluation items include the *coverage*, *link density*, *effective move ratio (EMR)*, *number of movements*, and *deploying time*.

The *Coverage* is to evaluate the area detecting utilization of a deploying method. When the number of sensors is fixed, the deploying method that can maximize the coverage area will have a higher area detecting rate. In order to return the sensing date back to the collector, an additional constraint on maximizing the coverage area is retaining the network connectivity. Our evaluation will consider maximizing the coverage and retaining the network connectivity simultaneously.

The *link density* evaluates the number of one-hop neighbors of each sensor. High-density deploying topology consolidates the network connectivity but sacrifices the sensing coverage. Low link density deploying one is just in contrary and may also generate coverage holes. The best deploying structure is to organize the sensors in hexagon

structure [15]. We can evaluate the number of linked neighbors to verify whether sensors are properly deployed.

The *effective move ratio (EMR)* of a sensor is defined as d_l/d_r , where d_l is the distance directly lines from initial location to its final location and d_r is the sensors' total moving distance from initial location to its final location. The optimal EMR is 1 that a sensor does not have redundant moving distance during the deploying procedure. High EMR implies that the sensor spends a lot of energy on useless moving. It can be used to estimate whether the deploying method is energy efficiency and whether the deploying method has serious oscillation moving problem. The simulation results will show the average EMR of all deployed sensors.

The *number of moving instructions* accumulates the number of received moving commands until a sensor becomes stationary. During the deploying procedure, a sensor will receive a moving instruction before it moves to its next location. Therefore, the communication overhead of a deploying method can also be reflected from the number of moving instructions. The more instructions are issued, the higher communication overhead a deploying method will be.

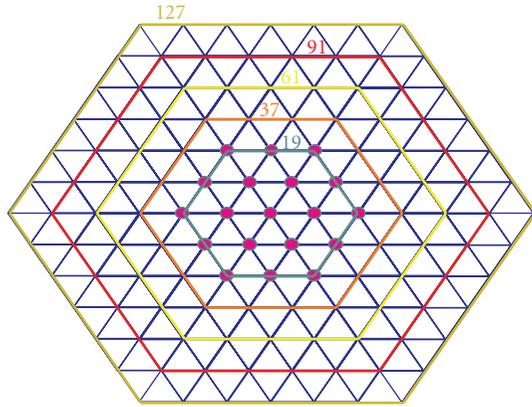


FIGURE 5: Perfect hexagon format.

The deploying time is the time period that the sensors require to finish the deployment. Short deploying time implies that the method can effectively direct each sensor moving to proper location. Sensor can quickly start the environment monitoring task. A good deploying time should shorten the deploying time.

4.1. Environment Setup. Initially, all deployed mobile sensors are randomly cast within a 100×100 area which is in the center of the interested area. Each sensor can identify its moving direction and the directions of the incoming signals. All sensors uniformly set their sensing range to 50 meters. The simulated network scale includes 19, 37, 61, 91, and 127 sensors. The number of sensors in each network scale can be organized as the perfect hexagon format shown as Figure 5. Therefore, we can draw out their optimal deploying topology for comparison.

In our simulation, the time spent for a sensor to change position is only proportional to its moving distance. The speed and geography factors are not considered. For estimating the deploying time, each sensor's moving speed is uniformly set to 2 meters/sec. Negotiating one deploying message between sensors requires 0.2 sec. The deploying procedure completes when all sensors become stationary. The simulation results are averaged from 500 random tests.

4.2. Numerical Results. Figure 6 compares the coverage area. The Ion-6 method has better coverage results than other methods in each network scale. The Ion-6 method exhibits well coverage results in large network scale. Except the network scale of 19 sensors, the Ion-6 method can have more 15% additional coverage area than the molecule model.

The V-force method has the worst coverage in our simulation. The V-force method will quickly expand the coverage area when network density is high. After a period of expanding, the network becomes sparse. The forces contributed by the neighbors will become weak. Sensors at the outer peripheral of network will stop being pushed even the sensors in the inner network still suffer great balanced forces. Therefore, the V-force's coverage area is

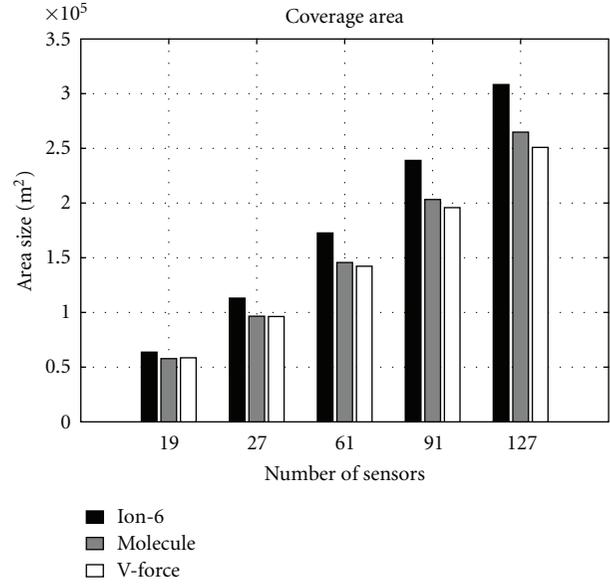


FIGURE 6: The coverage size of deploying methods.

slightly less than the molecule model. This phenomenon gradually becomes explicitly when the network scale grows.

Figure 7 shows the link density distribution of sensors in different network scale 19, 61, and 127. The optimal curves in these figures are the link density generated by the perfect hexagon topologies shown as Figure 5. The link density distribution of the Ion-6 method in each network scale is closer to the optimal one than other methods. The sensors with six neighbors dominate the major percentage. There is no sensor that has neighbors more than 6 and no sensors with a single neighbor. Similar to the optimal one, the second major percentage is four neighbors.

The first and second major percentage in V-force method and the Molecule model are 5 and 6. Because sensors have high link density, their coverage areas become small. In Figure 7, there is no sensor whose link density is zero. It implies that the Ion-6 can deploy sensors closer to the results of the optimal case.

Figure 8 displays the effective moving ratio. In the molecule model, sensors always move a small-step to adjust the moving direction. The average EMR of a sensor ranges from 1.3 to 1.6. In order to speed up the deploying of the molecule model, the sensors in the V-force method adapt a large move distance to expand quickly. When the network scale is small, each sensor's EMR is slightly more than the molecule model. However, when the network scale is more than 91, sensor's EMR rapidly increases because of the oscillation moving. The EMR in the network scale of 127 sensors is 2. It means that the redundant moving distance of every sensor is double.

By computing and selecting the suitable candidates, the Ion-6 method almost introduces zero redundant moving distance when network scale is 19. When the network scale is 127, the EMR of Ion-6 method is still less than 1.2. It also implies that Ion-6 method causes less redundant moving.

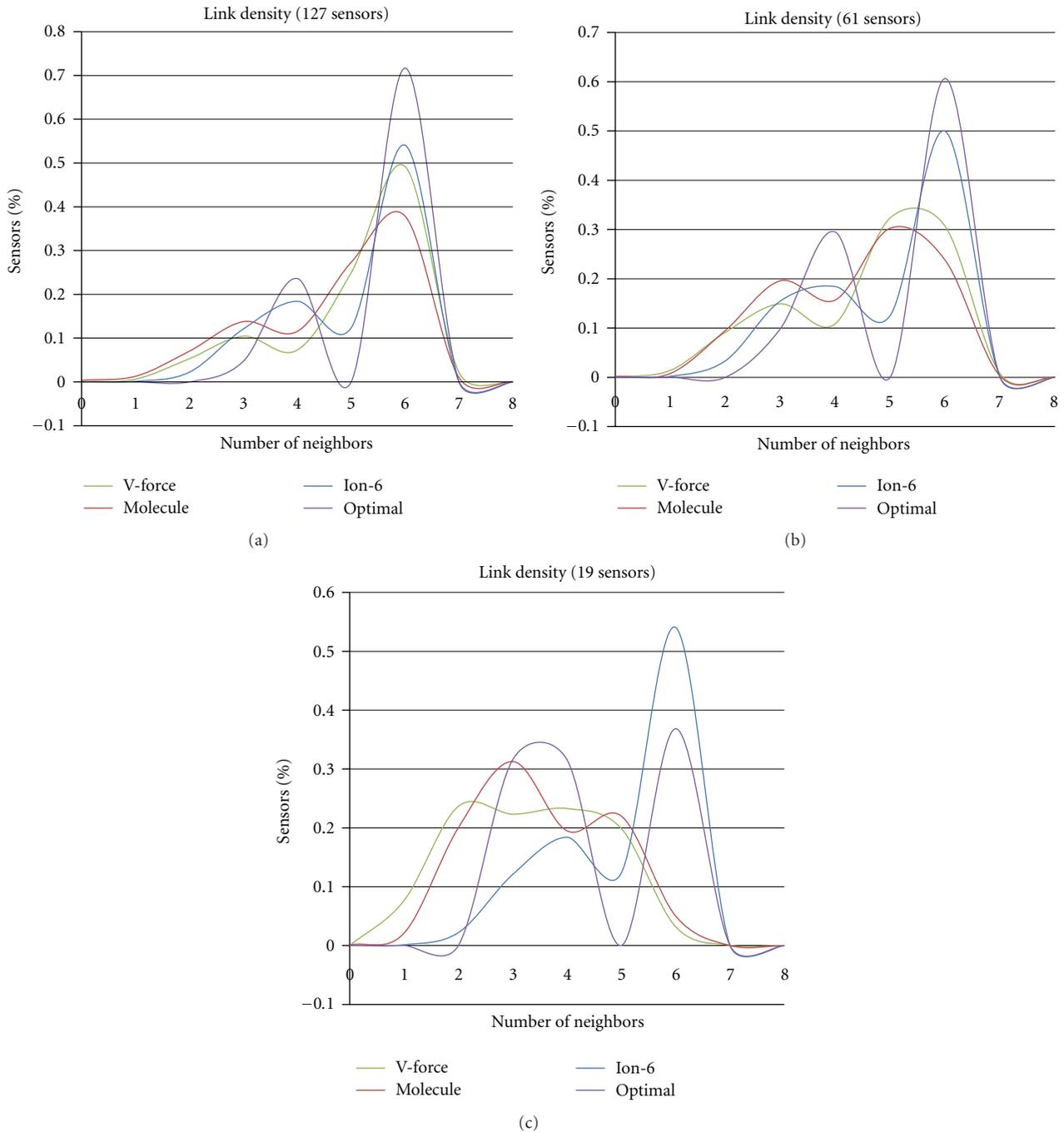


FIGURE 7: The link density (the distribution of the number of neighbors).

Figure 9 shows the average number of issued moving instructions of a sensor. In the molecule model, the small step adjusting strategy spends more communication messages to negotiate the moving direction with others. Therefore, when the network scale grows, the number of issued instructions rapidly increases. For the network scale of 127 sensors, each sensor has to send more than 170 messages. The communication overhead is heavy.

In contrary, sensors in the V-force method always try to move a large distance when they receive a moving

instruction. It can effectively reduce the number of issued moving instructions. Therefore, the average number of issued instructions of the V-force method in each network scale is less than half the number of molecule model.

The Ion-6 exhibits an outstanding result in this evaluation term. Sensors issue the moving instruction only when they are selected as candidates and ready to expel others. After a sensor has been selected as a candidate and expels others, it will no longer have to issue the moving instruction. Therefore, the Ion-6 method can effectively minimize the

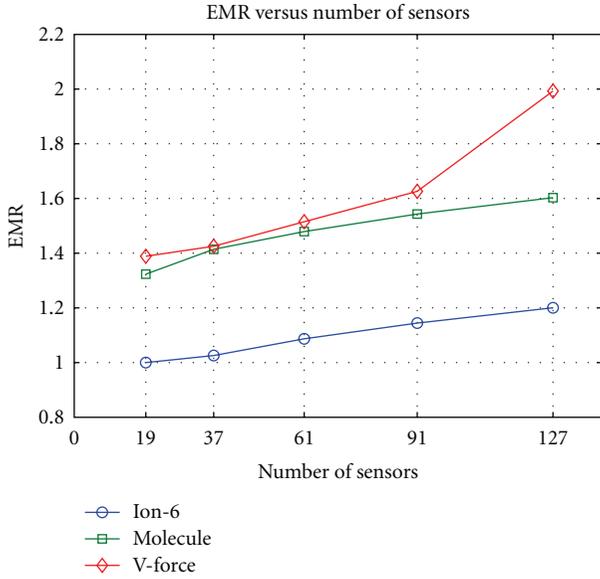


FIGURE 8: The effective move ratio (EMR).

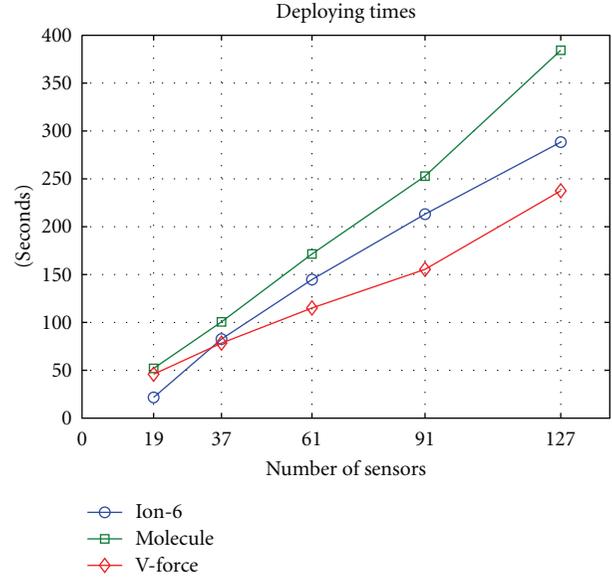


FIGURE 10: Deploying times.

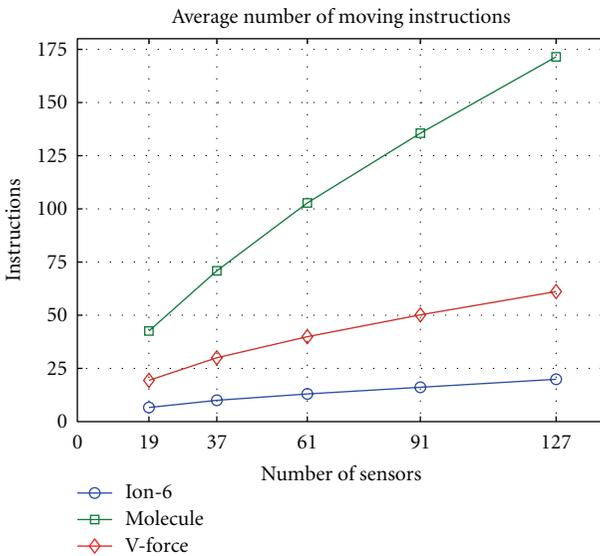


FIGURE 9: The number of moving instructions.

number of issued moving instructions. Even the network scale of 127 sensors in our simulation, the number of issued moving instructions of Ion-6 method is still less than 20.

Figure 10 shows the time to complete the deployment. Sensors in the molecule model use small moving step to adjust their position. When the number of sensors grows, time to complete the deployment increases rapidly. In the contrary, sensors in the V-force method use large moving step to adjust their positions when sensors are crowded. The adjusting step gradually shrinks when sensors spread out. By effectively adjusting the moving step, the V-force method can decrease the deploying time.

When the deploying procedure can be finished in one or two moving instructions, the deploying time of Ion-6 method is better than the V-force method, for example, the

scenario of network scale 19. When network scale is more than 37, the deploying time of Ion-6 gradually increases. In the network scale of 91 and 127, the deploying time of Ion-6 method is more than the one of V-force method about 50 seconds. However, Ion-6 is still better than the molecule model.

5. Conclusions

In this paper, we propose the Ion-6 self-deploying method which models the sensors as the ions. Sensors can compute their moving directions and distances independently. The deploying problem is to build ionic bonds between sensors. To organize the sensors as the hexagonal cellular topology, the number of ionic bonds of each sensor is set to six. The sensors selected as candidates are instructed to move. The other sensors will be expelled outside the sensing areas of the candidates. A location adjusting mechanism is also proposed to fix the fault caused by inaccurate estimating distance of the TDOA technique.

Simulation results prove that the proposed Ion-6 method can maximize the coverage and achieve the near-optimal hexagon topology without the explicitly position information. It can also efficiently control the movement of each sensor to minimize the oscillation moving problem in the V-force method. Furthermore, the Ion-6 method can reduce the communication overhead and deploying time in the molecule model.

Acknowledgment

This work was supported in part by the National Science Council, Taiwan, under Grant NSC100-2221-E-150-070.

References

- [1] Y. R. Tsai and Y. J. Tsai, "Sub-optimal step-by-step node deployment algorithm for user localization in wireless sensor networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC '08)*, pp. 114–121, Jun 2008.
- [2] Z. H. Yuan and G. F. Wang, "Sensor deployment strategy for collaborative target detection with guaranteed accuracy," in *Proceedings of the 4th International Conference on Mobile Ad-hoc and Sensor Networks (MSN '08)*, pp. 68–71, Dec 2008.
- [3] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar, "Sensor placement for grid coverage under imprecise detections," in *Proceedings of the 5th International Conference on Information Fusion*, vol. 2, pp. 1581–1587, July 2002.
- [4] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor network," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, vol. 3, pp. 1380–1387, April 2001.
- [5] D. O. Popa, H. E. Stephanou, C. Helm, and A. C. Sanderson, "Robotic deployment of sensor networks using potential fields," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04)*, vol. 1, pp. 642–647, month year.
- [6] A. Howard, M. J. Mataric, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Autonomous Robots*, vol. 13, no. 2, pp. 113–126, 2002.
- [7] N. Bartolini, T. Calamoneri, T. L. Porta, and S. Silvestri, "Mobile sensor deployment in unknown fields," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '10)*, pp. 1–5, 2010.
- [8] X. Z. Bai, L. Shu, C. J. Jiang, and Z. Z. Gao, "Coverage optimization in wireless mobile sensor networks," in *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom '09)*, pp. 1–4, 2009.
- [9] M. Tariq, Z. Zhou, Y. J. Park, and T. Sato, "Diffusion based self-deployment algorithm for mobile sensor networks," in *Proceedings of the IEEE Vehicular Technology Conference (VTC '10)*, pp. 1–5, 2010.
- [10] J. Li, L. Cui, and B. Zhang, "Self-deployment by distance and orientation control for mobile sensor networks," in *Proceedings of the International Conference on Networking, Sensing and Control (ICNSC '10)*, pp. 549–553, 2010.
- [11] K. H. Tan and M. A. Lewis, "Virtual structures for high-precision cooperative mobile robotic control," *Autonomous Robots*, vol. 4, pp. 387–403, 1997.
- [12] Y. Zou and C. Krishnendu, "Sensor deployment and target localization based on virtual forces," in *Proceedings of the IEEE 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 2, pp. 1293–1303, March-April 2003.
- [13] J. Lee, A. D. Dharne, and S. Jayasuriya, "Potential field based hierarchical structure for mobile sensor network deployment," in *Proceedings of the American Control Conference (ACC '07)*, pp. 5946–5951, Jul 2007.
- [14] X. Wu, L. Shu, M. Meng, J. Cho, and S. Lee, "Coverage-driven self-deployment for cluster based mobile sensor networks," in *Proceedings of the IEEE International Conference on Computer and Information Technology (CIT '06)*, p. 226, Sep 2006.
- [15] N. Heo and P. K. Varshney, "An intelligent deployment and clustering algorithm for a distributed mobile sensor network," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics '03*, vol. 5, pp. 4576–4581, Oct 2003.
- [16] R. S. Chang and S. H. Wang, "Self-deployment by density control in sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1745–1755, 2008.
- [17] M. R. Pac, A. M. Erkmén, and I. Erkmén, "Scalable self-deployment of mobile sensor networks: a fluid dynamics approach," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems '06*, pp. 1446–1451, Oct 2006.
- [18] C. Fang and C. P. Low, "A unified framework for movement-assisted sensor deployment," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '08)*, pp. 2057–2062, 2008.
- [19] D. B. Jourdan and O. L. de Weck, "Layout optimization for a wireless sensor network using a multi-objective genetic algorithm," in *Proceedings of the IEEE 59th Vehicular Technology Conference (VTC '04)*, vol. 5, pp. 2466–2470, Springer, New York, NY, USA, 2004.
- [20] W. Xiaoling, S. Lei, W. Jin, J. Cho, and S. Lee, "Energy-efficient deployment of mobile sensor networks by PSO," in *Proceedings of the International Workshop on Sensor Networks (IWSN '06)*, pp. 373–382, 2006.
- [21] H. T. Friis, "A note on a simple transmission formula," in *Proceedings of the Institute of Radio Engineers (IRE '46)*, pp. 254–256, 1946.

Research Article

Triangular Energy-Saving Cache-Based Routing Protocol by Energy Sieving

Chiu-Ching Tuan and Yi-Chao Wu

Graduate Institute of Computer and Communication Engineering, National Taipei University of Technology, Taipei 10608, Taiwan

Correspondence should be addressed to Yi-Chao Wu, t5419001@ntut.edu.tw

Received 19 March 2011; Revised 21 June 2011; Accepted 5 July 2011

Academic Editor: Yuhang Yang

Copyright © 2012 C.-C. Tuan and Y.-C. Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In wireless ad hoc networks, designing an energy-efficient routing protocol is a major issue since nodes are energy limited. To address energy issue, we proposed a triangular energy-saving cached-based routing protocol by energy sieving (TESCES). TESCES offered a grid leader election by energy sieving (GLEES), a cache-based grid leader maintenance (CGLM), and a triangular energy-saving routing discovery (TESRD). In GLEES, only few nodes join in grid leader election to be elected as a grid leader. New grid leader is elected directly by cache without sending extra control packets in CGLM. TESRD selects an energy-efficient path to transmit data packets. Hence, TESCES could save more energy for transmitting packets and prolong routing lifetime. Simulation results showed that TESCES could reduce 31% of energy consumption, prolong 67% of routing lifetime, and increase 19% of survival ratio of nodes. Furthermore, TESCES may be more outstanding as the number of nodes increased.

1. Introduction

Mobile ad hoc networks (MANETs) had attracted much attention recently. It consisted of a set of mobile nodes that can communicate with others through multiple hops without base stations. Packets sent by the source node are relayed by several intermediate nodes before arriving at the destination node [1–6].

Since battery technology is not likely to progress as fast as computing and communication technologies, designing an energy-efficient protocol to construct energy-efficient routing path becomes an important issue in MANETs [7–10]. The work in [11] indicated the fact that a protocol's behavior does have a significant impact on energy consumption of nodes. A node should tune its wireless interface card into doze mode whenever the node will not lower its own and the network's performance.

Among existing routing protocols, grid-based routing protocols are often used for energy-saving by tuning nodes into doze mode [12–16]. In grid-based routing protocols, once node is elected as a grid leader in its grid. Routing is conducted in a grid-by-grid manner through neighboring grid leaders. Only grid leaders must keep in active mode.

Other nodes are tuned into doze mode to save energy without demoting network connectivity.

When the remained energy of the current grid leader will be insufficient for grid management or data transmission, nodes in the same grid need to wake up and tune into active mode once receiving control packets for a grid leader election. However, some of these woken nodes with lower remained energy may consume more redundant energy for grid leader election. For routing discovery, grid-based routing protocols often select the route with minimum hops for transmitting packets without considering the required energy dissipation, such as AODV (ad hoc on demand distance vector routing protocol) [2] or DSR (dynamic source routing protocol) [1].

To address the above issues, we proposed a triangular energy-saving cached-based routing protocol by energy sieving (TESCES) in this paper. In TESCES, a grid leader election based on energy sieving (GLEES), a cache-based grid leader maintenance (CGLM), and a triangular energy-saving routing discovery (TESRD) are constructed. In GLEES, only few nodes need to join in grid leader election by GLEES. Hence, nodes with lower remained energy need not tune into active mode for saving energy. In CGLM, a node is directly to

be elected as a new grid leader without broadcasting extra control packets. TESRD builds an energy-efficient routing path for transmitting packets. TESCES therefore could reduce more energy consumption and prolong the lifetime of routes compared with a fully energy-aware and location-aware protocol (FPALA) and an energy-saving cache-based routing protocol (ESCR).

To evaluate and compare the performance of TESCES, FPALA, and ESCR clearly, we provide the mathematical formulas of energy consumption for grid leader election and maintenance. Simulation results showed the efficiency of TESCES. The rest of the paper is in the following sections. Section 2 presented the related work. Section 3 stated TESCES. Section 4 presented simulation results. Section 5 concluded this paper.

2. Related Works

Grid-based routing protocol is a kind of geographic routing protocols based on grid architecture. It partitions the network area into several square/hexagon grids by the location information such as global position system (GPS) [12, 13, 17–19], as shown in Figure 1. Routing is performed in a grid-by-grid manner. One node is elected as a grid leader in its grid. The responsibility of a grid leader includes (i) issuing routing discovery requests to its neighboring grid leaders, (ii) propagating data packets to its neighboring grid leaders, and (iii) maintaining routing paths which pass the grids. Nonleader nodes are not responsible for these jobs unless they are destinations of (i) and (ii) and sources or destinations of (iii). To reduce the unrequired collisions, the communication of nodes is divided into intragrid and intergrid modes. Routing discovery and maintenance could be modified from any of the following protocols: source routing and next-hop routing [14, 17].

However, most of grid-based routing protocols concentrated on routing discovery and maintenance without considering energy issues. To address energy issues, a fully energy-aware and location-aware routing protocol (FPALA) [12, 13] and an energy-saving cache-based routing protocol (ESCR) [14] were proposed. FPALA built a power mode management mechanism for grid leader election to save energy. All nodes need to wake up for joining a grid leader election. The node with the maximal remained energy in its grid is elected as the grid leader. Non-leader nodes then tune into doze mode to save energy without demoting the connectivity of network. For grid leader maintenance, FPALA restarts a new grid leader election whenever the remained energy of the current grid leader is insufficient. Nodes thus could save energy in grid leader elections. However, in FPALA, all nodes still need to tune into active mode to be elected as the grid leader by broadcasting extra control packets. Hence, nodes have to extra consume the redundant energy. To address this issue, energy-saving cache-based routing protocol (ESCR) was proposed [14] by us.

ESCR built a cache table in the first grid leader election. While the remained energy of current grid leader is not enough, a candidate node could be elected as a new

grid leader directly from cache without broadcasting any controlled packets. ESCR thus could save more energy than FPALA in grid leader maintenance.

However, nodes with lower remained energy still have to broadcast extra control packets in active mode to consume the unnecessary energy for grid leader election. Moreover, FPALA and ESCR both adopt the existing source routing or next-hop routing to build routing paths without considering the energy constrained for routing discovery. We therefore proposed a triangular energy-saving cached-based routing protocol by energy sieving (TESCES) in this paper.

3. Triangular Energy-Saving Cache-Based Routing Protocol by Energy Sieving

Triangular energy-saving cached-based routing protocol by energy sieving (TESCES) is a kind of energy-aware and location-aware grid-based routing protocols in MANETs. TESCES partitions the network area into several square grids based on GPS. One node in each grid is elected as a grid leader. For grid-based area, $r = 2\sqrt{2}l$, l is the side length of grid and r is the radio transmission radius of a grid leader, as shown in Figure 2 [17].

The minimum of value r implies that a grid leader is capable of talking to any one of its 8 neighboring grid leaders. Each node is set with a cache table to record gid_i , id_i , and E_i . The gid_i denotes the grid coordinates of node i , id_i denotes the identity of node i , and E_i denotes the remained energy of node i . The gid is defined as (X_i^g, Y_i^g) based on the location information from GPS. When the location of node i is at (X_i, Y_i) , X_i^g and Y_i^g are calculated as $\lfloor X_i/l \rfloor$ and $\lfloor Y_i/l \rfloor$, respectively. Each node has a unique id , such as MAC address.

In TESCES, routing is performed in a grid-by-grid manner through several grid leaders. Communication is divided into intra-grid and inter-grid modes. In intra-grid mode, node communicates directly with others with the same grid through its grid leader in one hop. In inter-grid mode, node communicates with one in different grid via its grid leader in multiple hops.

For routing discovery, TESCES uses a triangular energy-saving routing discovery (TESRD) to replace a traditional routing discovery, such as AODV [2]. In TESRD, nodes on the selected route could consume less energy by adjusting the transmission energy according to the required transmission distance while forwarding packets to next one.

3.1. Grid Leader Election by Energy Sieving. In grid-based routing protocols, the grid leader is responsible for routing, relaying packets, and maintaining correct operations of grids. Hence, an efficient grid leader election is needed.

However, in traditional grid leader election, all nodes in a grid need to turn into active mode for transmitting election packets. Some nodes thus may consume unnecessary energy because the remained energy is much lower than others in the same grid. For example, E_1, E_2, E_3 , and E_4 are 40 J, 38 J, 35 J, and 10 J, respectively. Node 4 is impossible to be elected

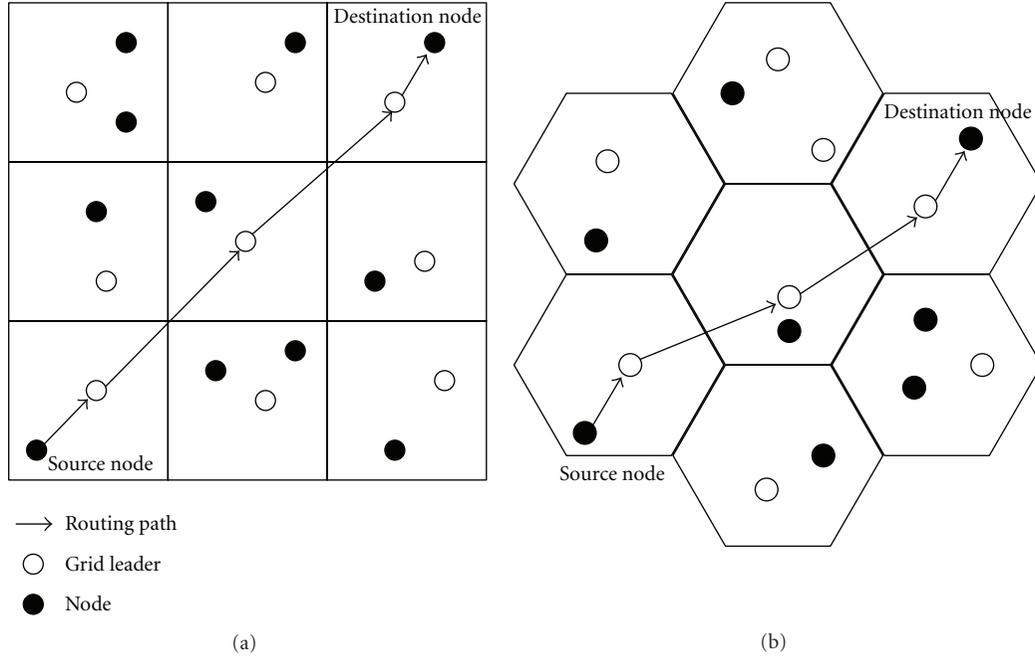
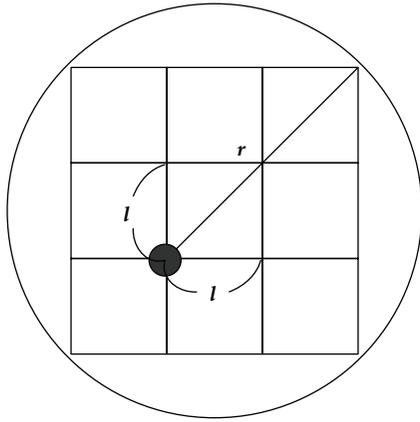


FIGURE 1: (a) Square grid, (b) Hexagon grid.

FIGURE 2: Relation between l and r in grid.

as a grid leader, but it must still consume the un-required energy for election.

Hence, in TESCES, a grid leader election by energy sieving (GLEES) is proposed to address this issue. In GLEES, each node is equipped with a GPS to get its location information. Power energy consumption of nodes could be adjusted by tuning the transmission radius. The full energy of each node is denoted as E_F .

GLEES defined a threshold value of joining in grid leader election (E_{join}). Initially, E_{join} is set to $E_F \times 0.9$. Before joining in grid leader election, each node in the same grid compares its remained energy E_i with E_{join} . While E_i is larger than E_{join} , node i tunes into active mode and joins the grid leader election. Otherwise, node i is kept in doze mode. Hence, only few nodes join the leader election. Other nodes with

lower energy are kept in doze mode to save energy. Process of GLEES is stated as follows.

- (1) First, node i compares E_i with E_{join} . While E_i is larger than E_{join} , node i broadcasts a $BID(gid_i, id_i, E_i)$ packet. Otherwise, node i is kept in doze mode.
- (2) While node j receives a BID packet, node j compares E_j with E_i listed in the received BID packet. If E_j is larger than E_i , node j replaces E_i with E_j . Then, node j broadcasts the updated BID packet with E_j to its neighboring nodes in its grid; otherwise, node j stops broadcasting.
- (3) When the last node i does not get any BID packet in a predefined time (T_{pre}), node i transfers itself into the grid leader. Then it declares its existence by broadcasting a $GATE(gid_i, id_i)$ packet to all nodes in its grid.
- (4) When node k receives a $GATE$ packet, it replies a $BID_E(gid_k, gid_k, E_k)$ packet to its grid leader.
- (5) The grid leader sorts its cache table in a descending way by E_k recorded in BID_E packets from all nodes.

GLEES could avoid no grid leader to be elected in grid leader election. When no grid leader is elected after T_{pre} , E_{join} is decreased to be multiplied by 0.9, and then GLEES restarts.

For example, E_F is set to 40 (J); thus E_{join} ($E_F \times 0.9$) is 36 (J), initially. The remained energy of 11 nodes is shown in Figure 3. Without GLEES, each node needs to consume k (J) to join a grid leader election by broadcasting a packet. The total energy consumption for joining in grid leader election is $11 \times k$. In GLEES, only nodes 1, 2, and 10 need to join the grid leader election since E_1, E_2 , and E_{10} are larger than E_{join} . The total energy consumption thus reduces to be $3 \times k$.

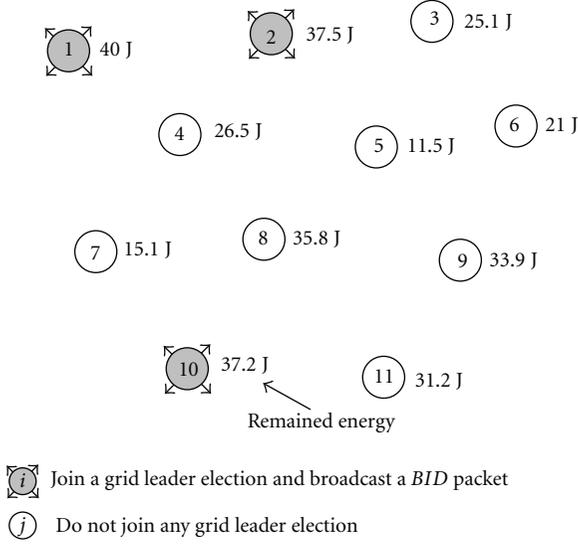


FIGURE 3: Example of GLEES.

3.2. Cache-Based Grid Leader Maintenance. Routing maintenance is to keep the lifetime of a routing as long as possible. Under TESCES, except the source and destination nodes, each intermediate node is the grid leader. Therefore, the grid leader maintenance in each grid is an important issue for routing maintenance. To address this issue, cache-based grid leader maintenance (CGLM) was proposed. When the remained energy (E_{rem}) of current grid leader is lower than the threshold energy (E_{th}) of retired grid leader, the grid leader maintenance is started. Process of CGLM is as follows.

- (1) When E_{rem} of grid leader is lower than E_{th} , the grid leader retrieves the candidate id from the first row in its cache table. Then it unicasts a $GATE_E(gid, id, NT, CT)$ packet to the new grid leader directly, where NT is the neighboring grid leader table and CT is the cache table. The former grid leader then could transfer itself into an ordinary node.
- (2) The new grid leader broadcasts a $GATE$ packet to declare its existence and then deletes the first row in its CT .
- (3) While CT becomes empty, CT has to be reestablished by GLEES.

In CGLM, new grid leader is elected directly without broadcasting any extra control packets. CGLM thus could save more power energy for data transmission.

For example, assume that E_F is 40 (J), E_{th} is set to 15 (J), and node 5 is the current grid leader, as shown in Figure 4. Once E_5 is less than E_{th} , new grid leader has to be elected. In FPALA, all nodes need to join in grid leader election. The total energy consumption thus is $10 \times k$ for election. In CGLM, only node 5 needs consuming k (J) to unicast a packet to node 2 that becomes the new grid leader directly. Other nodes tune into doze mode to save energy. Hence, CGLM could save more energy than FPALA.

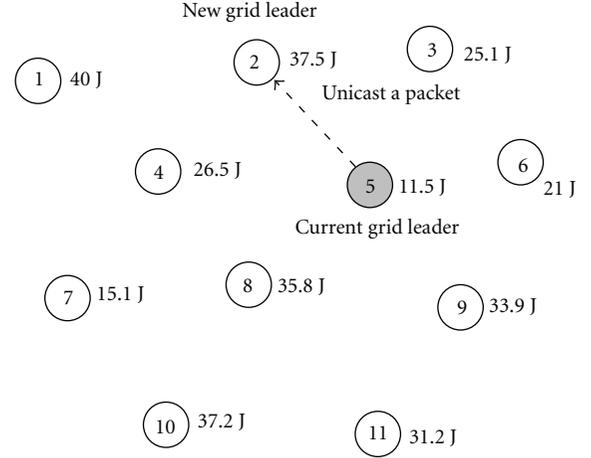


FIGURE 4: Example of CGLM.

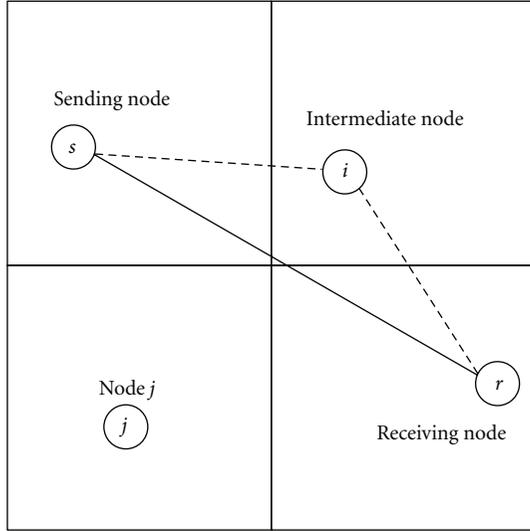
3.3. Triangular Energy-Saving Routing Discovery. In traditional grid-based routing protocols, minimum hop routing discovery is often used through several grid leaders without considering the energy constrain [12, 17]. To address this issue, we proposed a triangular energy-saving routing discovery (TESRD) by integrated HCB model [8, 9, 20] with GPSR [21].

TESRD is a two-phase process. In the first phase, packets are marked with their destinations' locations by their originator. A forwarding grid leader makes a locally optimal choice to decide the next packet's hop. The locally optimal choice of next hop is the neighboring leader that is geographically closest to the next packet's destination. Forwarding packets in this regime follows the closer geographic hops until the destination is reached.

In the second phase, TESRD adopts a greedy algorithm to compute global near-optimal power-efficient routings based on the local optimal choice for the next forwarding node. In TESRD, let s be the sending node and r the next receiving node along the routing path. The distance d is measured from s to r . Transmission energy consumption is proportional to d^α for $\alpha \geq 2$ based on HCB model [20]. Before s forwarding $RREQ$ (route request) to r , s finds an intermediate node i from its neighboring leaders to r in one hop. If $\overline{si}^2 + \overline{ir}^2 < \overline{sr}^2$, s utilizes the intermediate leader i to forward a packet instead of sending the packet directly to r , where $(\overline{si}^2 + \overline{ir}^2) < (\overline{sj}^2 + \overline{jr}^2)$ as shown in Figure 5.

To reduce energy consumption, TESRD could select a more energy-saving path based on the changes of gid of next intermediate node. Assume that the gid of node s and r are (X_s^g, Y_s^g) and (X_r^g, Y_r^g) , respectively. In case 1, if $X_s^g \neq X_r^g$ and $Y_s^g \neq Y_r^g$, TESRD selects the two intermediate grid leaders located at (X_s^g, Y_r^g) and (X_r^g, Y_s^g) to find a more energy-saving path, as shown in Figure 6.

In case 2, if $X_s^g = X_r^g$ and $Y_s^g \neq Y_r^g$, apply the four intermediate grid leaders in $(X_s^g - 1, Y_r^g)$, $(X_s^g + 1, Y_r^g)$, $(X_s^g - 1, Y_s^g)$, and $(X_s^g + 1, Y_s^g)$ to find an energy-saving path, as shown in Figure 7.



--- TESRD
— Minimum next-hop routing

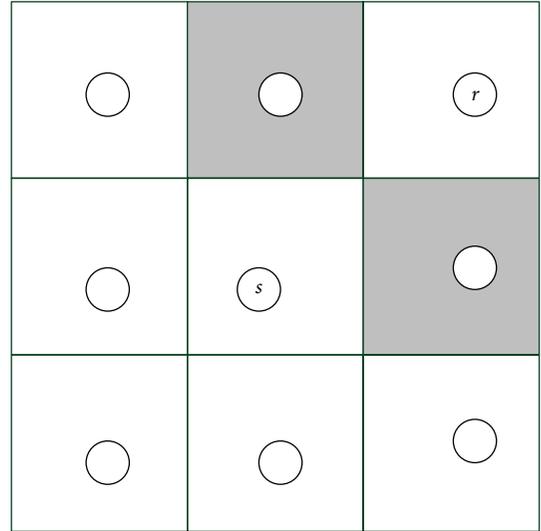
FIGURE 5: Example of TESRD.

In case 3, if $X_s^g \neq X_r^g$ and $Y_s^g = Y_r^g$, TESRD selects the four intermediate grid leaders in $(X_s^g, Y_s^g - 1)$, $(X_s^g, Y_s^g + 1)$, $(X_r^g, Y_s^g - 1)$, and $(X_r^g, Y_s^g + 1)$ for finding a more energy-saving path, as shown in Figure 8.

3.4. Medium Access Control Channel Assignment. In TESCES, routing is conducted in two levels: intra-grid and inter-grid. The former is supported by the point coordination function (PCF) of IEEE 802.11, and the latter is supported by the distributed coordination function (DCF) of 802.11. The time interval is divided evenly into a sequence of superframes for all nodes participating in the networks. We appended *BID_E* and *GATE_E* packets to the modified superframe based on FPALA. The inter-grid and intra-grid routing phases are under the superframe, as shown in Figure 9.

In the leader phase, all nodes must be awake. Only leaders have right to access their channels. If no leader exists in a grid, the next phase becomes an election phase for nodes to compete to be a grid leader. If E of the leader is below E_{th} , the next phase becomes a maintenance phase to generate a new grid leader. In the intra-grid phase, the leader polls its nodes in a round-robin manner. In the inter-grid phase, only leaders can send/receive packets. Since more than one node may try to compete as a grid leader, the *BID* packets are broadcasted in a contention basis. In TESCES, superframes need to be synchronized among all grids.

For the intra-grid routing, if a packet is targeted at a node resident in the same grid, this packet is sent to the node directly during the intra-grid phase. For the inter-grid routing, a packet is forwarded in a grid-by-grid manner during the inter-grid phase. An inter-grid routing could be modified based on the protocols: source routing or next-hop routing. However, these protocols do not address the energy



■ The possible intermediate grid leader

FIGURE 6: Case 1 in TESRD.

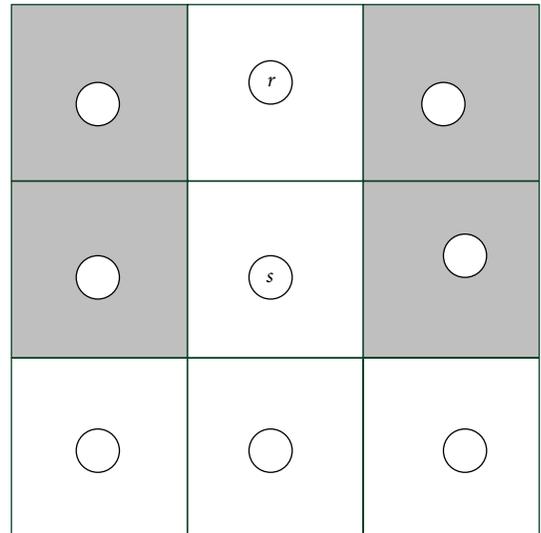


FIGURE 7: Case 2 in TESRD.

issue. Hence, we proposed a triangular energy-saving routing discovery (TESRD) in TESCES.

To avoid channel interference among neighboring grids, totally night channels are needed in TESCES, as shown in Figure 10. The number (1–9) in each grid represents the channel to be used by that grid. The channels based on frequency reuse form a pattern, called a cluster that appears repeatedly in a regular way.

3.5. Energy Consumption Formula. To evaluate the performance effectively, the notations in the energy formula were defined as listed in Table 1. We formulated the energy consumption of grid leader election in TESCES, E_{ele}^{TES} , in

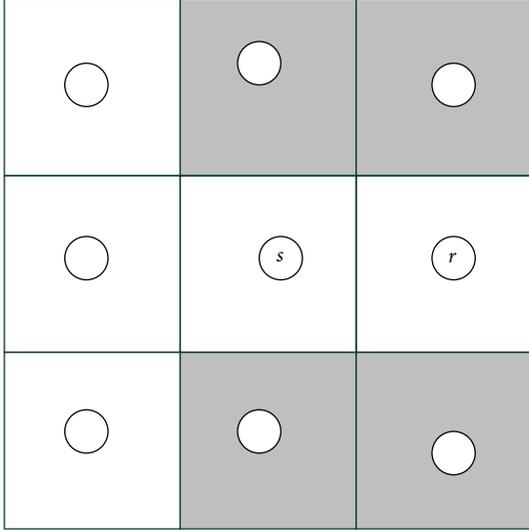


FIGURE 8: Case 3 in TESRD.

(1) including N_G^i , P_j ($0 \leq P_j \leq 1$), P_B ($0 \leq P_B \leq 1$), L_{BID} , L_{RTS} , L_{CTS} , T_{ele} , T_{lea} , T_{sup} , D , E_t , and E_d . In FPALA, $E_{\text{ele}}^{\text{FPA}}$, its energy consumption of a grid leader election is defined as (2). In ECSR, its energy consumption of a grid leader election, $E_{\text{ele}}^{\text{ECSR}}$, is calculated as (3). Since P_j equals 1 in FPALA and ECSR, $E_{\text{ele}}^{\text{TES}}$ is not larger than $E_{\text{ele}}^{\text{FPA}}$ and $E_{\text{ele}}^{\text{ECSR}}$:

$$E_{\text{ele}}^{\text{TES}} = \left(\left[\frac{N_G^i \times P_j \times (L_B + L_{\text{RTS}} + L_{\text{CTS}})}{T_{\text{ele}} \times D} \right] + N_G^i \times \left[\frac{L_G}{T_{\text{lea}} \times D} \right] \right) \times T_{\text{sup}} \times EC_a \quad (1)$$

$$+ \left(\left[\frac{N_G^i \times (1 - P_j) \times (L_B + L_{\text{RTS}} + L_{\text{CTS}})}{T_{\text{ele}} \times D} \right] \right) \times T_{\text{sup}} \times EC_d,$$

$$E_{\text{ele}}^{\text{FPA}} = \left(\left[\frac{N_G^i \times (L_B + L_{\text{RTS}} + L_{\text{CTS}})}{T_{\text{ele}} \times D} \right] + N_G^i \times \left[\frac{L_G}{T_{\text{lea}} \times D} \right] \right) \times T_{\text{sup}} \times EC_a, \quad (2)$$

$$E_{\text{ele}}^{\text{ECSR}} = \left(\left[\frac{N_G^i \times (L_B + L_{\text{RTS}} + L_{\text{CTS}})}{T_{\text{ele}} \times D} \right] + N_G^i \times \left[\frac{L_G}{T_{\text{lea}} \times D} \right] + \left[\frac{N_G^i \times (L_{\text{BE}} + L_{\text{RTS}} + L_{\text{CTS}})}{T_{\text{lea}} \times D} \right] \right) \times T_{\text{sup}} \times EC_a. \quad (3)$$

The formula of energy consumption of the grid leader maintenance of TESCES is calculated as (4). For FPALA, the formula of energy consumption of grid leader maintenance is the same as $E_{\text{ele}}^{\text{FPA}}$. In ECSR, the formula of energy

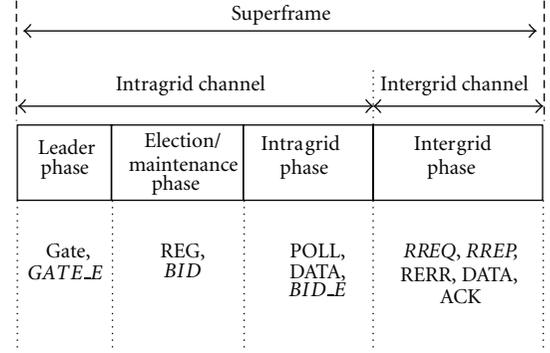


FIGURE 9: Superframe of TESCES.

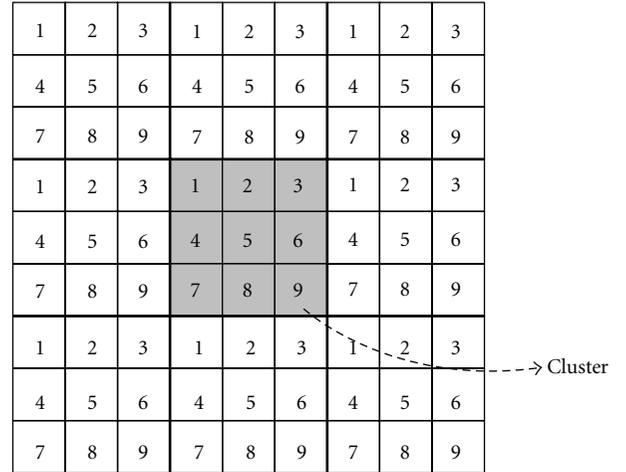


FIGURE 10: Channel Assignment in TESCES.

consumption of the grid leader maintenance is calculated as (5). Because the energy consumption of doze mode is much lower than that of active mode, it proves that TESCES could save more energy than that of FPALA and ECSR for the grid leader election and maintenance by (1)–(5):

$$E_{\text{mai}}^{\text{TES}} = N_G^i \times \left[\frac{L_{\text{GE}}}{T_{\text{lea}} \times D} \right] \times T_{\text{lea}} \times E_a + N_G^i \times \left[\frac{L_{\text{GE}}}{T_{\text{lea}} \times D} \right] \times (T_{\text{ele}} + T_{\text{tra}} + T_{\text{ter}}) \times E_d, \quad (4)$$

$$E_{\text{mai}}^{\text{ECSR}} = N_G^i \times \left[\frac{L_{\text{GE}}}{T_{\text{lea}} \times D} \right] \times (T_{\text{lea}} + T_{\text{ele}}) \times E_a + N_G^i \times \left[\frac{L_{\text{GE}}}{T_{\text{lea}} \times D} \right] \times (T_{\text{tra}} + T_{\text{ter}}) \times E_d. \quad (5)$$

Energy consumption of routing discovery is calculated based on time of transmitting packets. Hence, examining the total superframes is to obtain transmitting time for TESCES, FPALA, and ECSR. Assume that the number of grid leaders along the routing is N_{lea} . In TESCES, it needs a *RREQ* packet, a *RREP* packet, and data packets (N_{data}) to build a routing path and transmit data packets. Total lengths of these packets are $L_{\text{data}} \times N_{\text{data}} + (L_{\text{RREQ}} + L_{\text{RREP}})$. Since routing discovery is performed in inter-grid phase, the

TABLE 1: Notations of mathematical formula.

Name	Description
E_{ele}^{TES}	Energy consumption of grid leader election in TESCES
E_{ele}^{FPA}	Energy consumption of grid leader election in FPALA
E_{ele}^{ESCR}	Energy consumption of grid leader election in ESCR
E_{mai}^{TES}	Energy consumption of grid leader maintenance in TESCES
E_{mai}^{ESCR}	Energy consumption of grid leader maintenance in ESCR
E_r^{TES}	Energy consumption of routing in TESCES
E_r^{FPA}	Energy consumption of routing in FPALA
E_r^{ESCR}	Energy consumption of routing in ESCR
N_G^i	Total nodes in the i th grid
P_j	Probability of nodes joining a grid leader election
P_B	Probability of nodes broadcasting a <i>BID</i> packet in a grid leader election
L_B	Length of <i>BID</i> packet
L_{BE}	Length of <i>BID_E</i> packet
L_G	Length of <i>GATE</i> packet
L_{GE}	Length of <i>GATE_E</i> packet
L_{RTS}	Length of <i>RTS</i> packet
L_{CTS}	Length of <i>CTS</i> packet
L_{data}	Length of <i>DATA</i> packet
L_{RREQ}	Length of <i>RREQ</i> packet
L_{RREP}	Length of <i>RREP</i> packet
T_{ele}	Time interval of election phase
T_{lea}	Time interval of leader phase
T_{sup}	Time interval of a superframe
T_{tra}	Time interval of intra-grid phase
T_{ter}	Time interval of inter-grid phase
D	Transmission data rate
EC_a	Energy consumption in active mode
EC_d	Energy consumption in doze mode
E_a^{si}	Energy consumption from sender to intermediate leader i
E_a^{ir}	Energy consumption from intermediate leader i to receiver
E_a^{sr}	Energy consumption from sender to receiver
N_{data}	Number of data packets along the routing
N_{lea}	Number of total leaders along the routing
$\bar{s}r$	Distance from sending node to receiving node
$\bar{s}i$	Distance from sending node to intermediate node i
$\bar{i}r$	Distance from intermediate node to receiving node i

number of superframes could be computed as $L_{data} \times N_{data} + (L_{RREQ} + L_{RREP})$ divided by $(T_{ter} \times D)$. Because TESCES needs an intermediate node to consume less energy to forward packets, the energy consumption of routing discovery in TESCES (E_r^{TES}) is defined as (6). Since FPALA and ESCR do not use intermediate nodes to forward packets, the energy consumption of routing in both are the same as (7) and (8). Based on HCB model [20], the energy consumption E_a^{si} and

TABLE 2: Simulation parameters.

Name	Value
Simulation area	$1000 \times 1000 \text{ m}^2$
Number of grids	10×10
Number of nodes (N_n)	100, 200, 400
Side length of grid (d)	100 m
Transmission radius of a radio signal (r)	$200\sqrt{2}$ m
Data rate (D)	11 Mbits/s
Time interval time of a superframe (T_{sup})	200 ms
Time interval of leader phase (T_{lea})	1 ms
Time interval of election phase (T_{ele})	4 ms
Transmission rate	200 packet/s
Size of packet	1500 bytes
Full battery energy of a node (E_f)	40 J
Threshold value of retirement (E_{th})	8 J
Energy consumption in active mode (EC_a)	280 mW
Energy consumption in doze mode (EC_d)	10 mW

E_a^{ir} are defined as (9) and (10). Because $\bar{s}i^2 + \bar{i}r^2 < \bar{s}r^2$ in TESCES, E_r^{TES} must be less than or equal to E_r^{FPA} and E_r^{ESCR} . By (6)–(8), formulas showed that TESCES could reduce more energy consumption of routing discovery than FPALA and ESCR:

$$E_r^{TES} = \left[\frac{L_{data} \times N_{data} + (L_{RREQ} + L_{RREP})}{T_{ter} \times D} \right] \times N_{lea} \times T_{sup} \times (E_a^{si} + E_a^{ir}), \quad (6)$$

$$E_r^{FPA} = \left[\frac{L_{data} \times N_{data} + (L_{RREQ} + L_{RREP})}{T_{ter} \times D} \right] \times N_{lea} \times T_{sup} \times E_a^{sr}, \quad (7)$$

$$E_r^{ESCR} = \left[\frac{L_{data} \times N_{data} + (L_{RREQ} + L_{RREP})}{T_{ter} \times D} \right] \times N_{lea} \times T_{sup} \times E_a^{sr}, \quad (8)$$

$$E_a^{si} = E_a^{sr} \times \frac{\bar{s}i^2}{\bar{s}r^2}, \quad (9)$$

$$E_a^{ir} = E_a^{sr} \times \frac{\bar{i}r^2}{\bar{s}r^2}. \quad (10)$$

4. Simulation Results

Performance of TESCES was measured and compared with those of FPALA and ESCR by simulations coded in a C# language. First, we described the simulation environment and performance metrics and then analyzed the experimental results. The simulation parameters are listed in Table 2. The time ratio of the intra-grid phase to the inter-grid phase is 1 : 4. Energy consumption could be adjusted based on the transmission radius of nodes [12, 14].

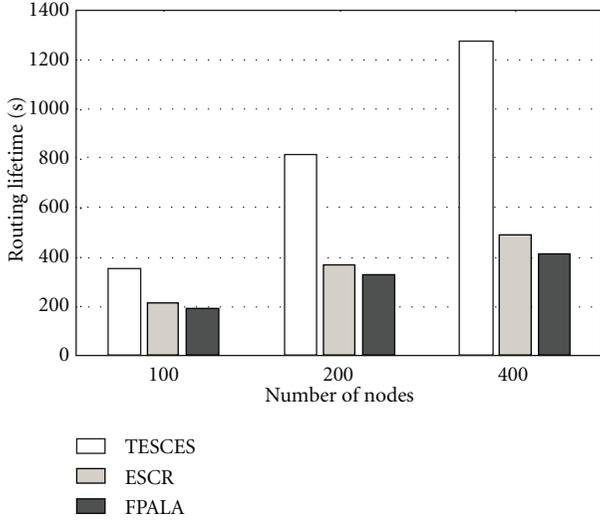
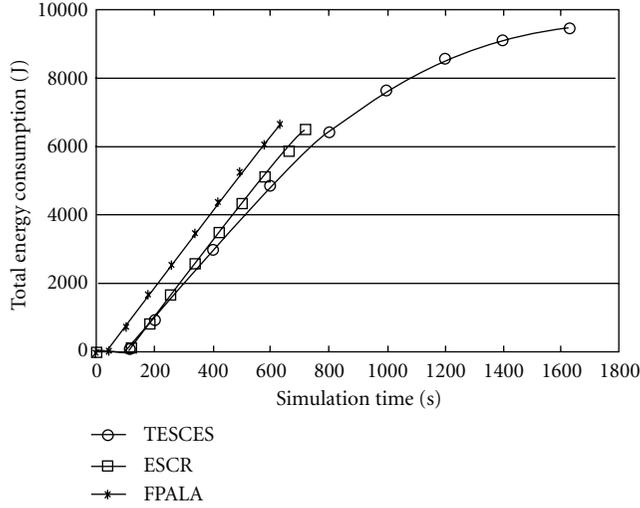


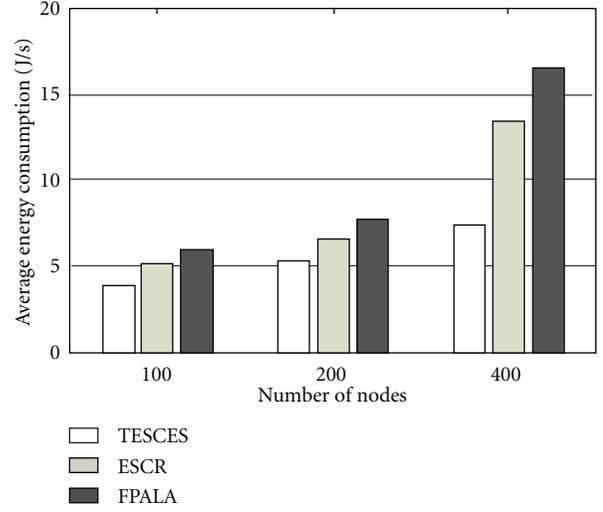
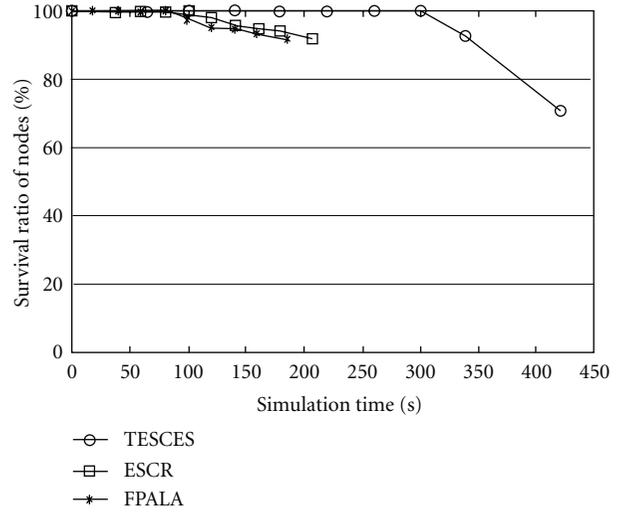
FIGURE 11: Routing Lifetime.

FIGURE 12: Total energy consumption ($N_n = 400$).

4.1. Performance Metrics. We evaluate the routing lifetime (T_r), average energy consumption (E_{avg}), and ratio of survival nodes (R_s) under TESCES, FPALA, and ESCR, respectively. T_r is defined as the time span from that the routes are living to that no path is between the source and destination nodes. E_{avg} is defined as the total energy consumption (E_t) divided by T_r . R_s denotes the ratio of survival nodes being with higher energy than E_{th} to the total nodes in the initial networks.

4.2. Experimental Results. TESCES improves 67% and 84% of T_r more than ESCR and FPALA, respectively, for N_n is 100, as shown in Figure 11. While N_n is 400, TESCES improves 90% and 2 times of T_r more than ESCR and FPALA, respectively.

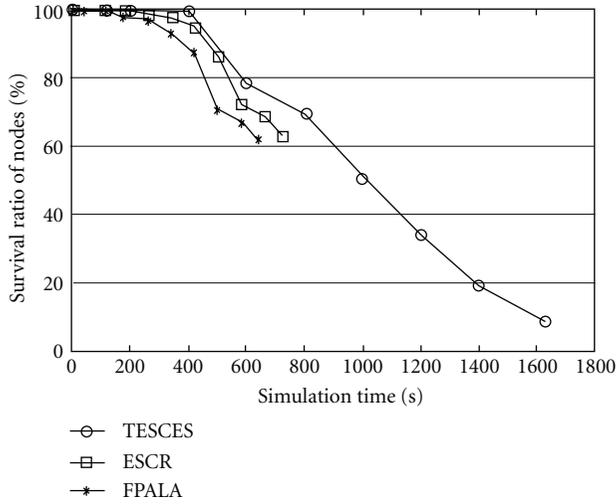
Figure 12 showed that TESCES reduces 31% of E_t compared with E_t of ESCR and 40% with E_t of FPALA when

FIGURE 13: Average energy consumption ($N_n = 100, 200$, and 400).FIGURE 14: Survival ratio of nodes ($N_n = 100$).

the simulation time is around 620 seconds. T_r is different among TESCES, ESCR, and FPALA, and we further evaluate E_{avg} , as shown in Figure 13. Figure 13 showed that TESCES could reduce more E_{avg} , as N_n is increased.

TESCES increases 9% of R_s in ESCR and 13% of R_s in FPALA, while N_n is 100 and simulation time is 200 seconds, as shown in Figure 14. While N_n is 400 and simulation time is 600 seconds, TESCES increases 11% of R_s in ESCR and 19% of R_s in FPALA, as shown in Figure 15. Figures 14 and 15 also showed that the utilization of TESCES is higher than that of FPALA and ESCR, because no path is between the sending and receiving nodes, while the unused alive nodes of FPALA and ESCR both are still over 90% of total nodes.

Energy consumption evaluation is consisted of the energy consumption of grid leader election, grid leader maintenance, and routing discovery. Hence, the formulas (1)–(8) are used in constructing these figures.

FIGURE 15: Survival ratio of nodes ($N_n = 400$).

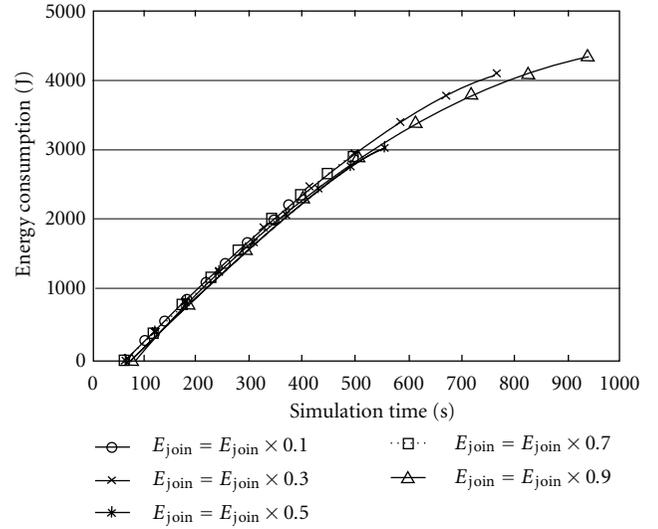
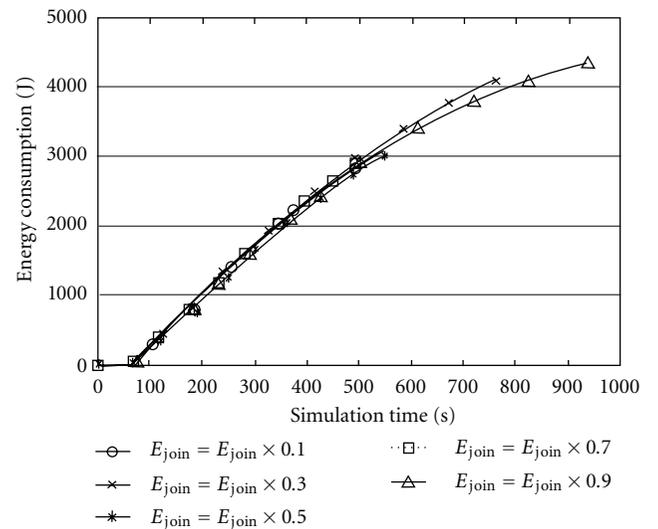
Simulation is ended off when no path exists between the source and destination nodes. The available routing paths in FPALA and ESCR both are broken earlier than TESCES. The curves for ESCR and FPALA thus do not continue for whole simulation time, as shown in Figures 12, 14, and 15.

To avoid most of nodes to consume redundant energy, E_{join} is set to $E_F \times 0.9$, initially. If no grid leader is elected in the first time, E_{join} is further decreased to be multiplied by 0.9. Different E_{join} , however, may affect E_t of TESCES. We vary E_{join} that decreased to be multiplied from 0.9 to 0.1 for evaluating E_t in 200, 400, and 800 of N_n , respectively, as shown in Figures 16, 17, and 18.

In Figures 16, 17, and 18, E_t in different E_{join} and N_n are all the same before the half of the longest simulation time. For example, in Figure 16, the largest simulation time is around 1000 (s), E_t in different E_{join} are the same before 500 (s). This is caused that the remained energy of most nodes are the same approximately before the half of simulation lifetime. Hence, the number of nodes joined in grid leader election is around the same even in different E_{join} . After the half of routing lifetime, the difference of remained energy among all nodes is increased. The number of nodes joined in grid leader election with different E_{join} is different gradually. E_t in different E_{join} thus are getting different obviously. As a result, we focus on E_t after the half of simulation time.

In Figure 16, TESCES has largest E_t with $E_{\text{join}} = E_{\text{join}} \times 0.3$, while N_n is 200. TESCES has largest E_t with $E_{\text{join}} = E_{\text{join}} \times 0.3$, as N_n is 400, as shown in Figure 17. However, TESCES has largest E_t with $E_{\text{join}} = E_{\text{join}} \times 0.1$, as N_n is 800, as shown in Figure 18. In Figures 16, 17, and 18, TESCES does not have the worst case for energy consumption in fixed E_{join} . This is caused that the remained energy of nodes may be the same or much different compared with that of others in each time of GLEES and CGLM.

In Figures 16, 17, and 18, TESCES in $E_{\text{join}} = E_{\text{join}} \times 0.9$ had the least E_t or less E_t than in other E_{join} . It proved that $E_{\text{join}} = E_{\text{join}} \times 0.9$ is the best or better value to reduce more

FIGURE 16: Energy consumption with different E_{join} ($N_n = 200$).FIGURE 17: Energy consumption with different E_{join} ($N_n = 400$).

energy consumption for TESCES. Once selecting the wrong or different E_{join} , TESCES may consume more energy.

5. Conclusions

Since the power energy of mobile nodes are limited, designing an efficient energy-saving routing protocol becomes an important issue in wireless ad hoc networks. To address this issue, many energy-aware routing protocols were proposed. Among these protocols, grid-based routing protocol is the general solution because nodes could be tuned into doze mode to save energy. The grid-based routing protocol is composed of grid leader election, grid leader maintenance, and routing discovery.

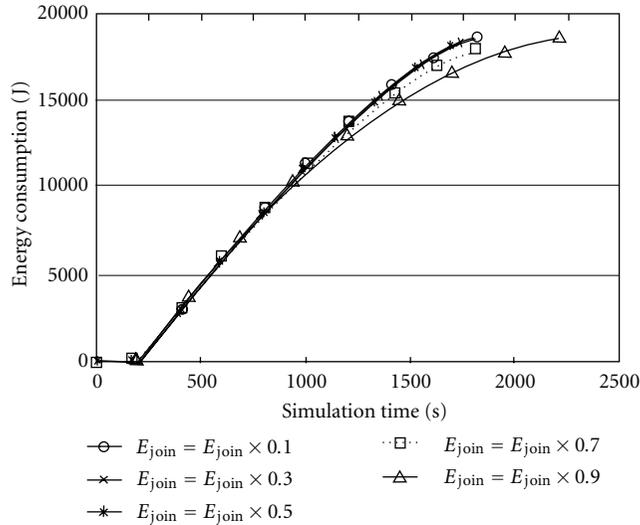


FIGURE 18: Energy consumption with different E_{join} ($N_n = 800$).

In grid leader election, each node has to consume energy to be elected as a grid leader. However, some nodes are unsuitable to be the grid leader because its remained energy is much lower than that of others. Nodes with lower remained energy need not consume the redundant energy in grid leader election.

In grid leader maintenance, each node needs to be in active mode for a grid leader election when the remained energy of current grid leader is insufficient for forwarding packets.

In routing discovery, most of grid-based routing protocols concentrated on robustness and minimum hop count of routes but ignored the required energy consumption. Nodes thus may consume more energy to transmit data.

To address the above issues, a triangular energy-saving cache-based routing protocol by energy sieving (TESCES) was proposed in this paper. In TESCES, a grid leader election by energy sieving (GLEES), a cache-based grid leader maintenance by cache (CGLM), and a triangular energy saving routing discovery (TESRD) are constructed.

In GLEES, only some nodes need to join a grid leader election to be elected as a grid leader, and other nodes are turned into doze mode to save energy. In CGLM, the new grid leader is appointed from cache table directly without broadcasting control packets to save energy while the remained energy of current grid leader is lower than the threshold. TESRD selects an energy-efficient routing path compared with the on-demand routing discovery. Therefore, TESCES could save more energy for data transmission and prolong the time of routing.

To measure and compare the performance of TESCES, FPALA, and ESCR, we conducted some simulations for evaluating grid leader election, grid leader maintenance, and routing discovery. Simulation results proved that TESCES could save more power energy than FPALA and ESCR.

Experimental results showed that TESCES prolongs 67% of ESCR and 84% of FPALA, respectively. For energy

consumption, TESCES reduces 31% of ESCR and 40% of FPALA. For survival ratio of nodes, TESCES increases 11% of ESCR and 19% of FPALA.

Furthermore, the routing lifetime, energy consumption, and survival ratio of nodes may be better in TESCES as the number of mobile nodes is increased.

References

- [1] D. Johnson, D. Maltz, and J. Jetcheva, "The dynamic source routing (DSR) protocol for mobile ad hoc networks," *Internet Engineering Task Force*, Internet Draft, 2004.
- [2] C. E. Perkins, E. M. Belding-Royer, and S. Das, "Ad hoc on demand distance vector (AODV) routing protocol," *Mobile Ad Hoc Networking Working Group*, Internet Draft, 2002.
- [3] 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.
- [4] Z. Zang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 1, pp. 24–37, 2006.
- [5] O. Liang, Y. A. Sekercioglu, and N. M. Mani, "A survey of multipoint relay based broadcast schemes in wireless ad hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 4, pp. 30–46, 2006.
- [6] L. Chen and W. B. Heinzelman, "A survey of routing protocols that support QoS in mobile ad hoc networks," *IEEE Network*, vol. 21, no. 6, pp. 30–38, 2007.
- [7] N. Vassileva and F. Barcelo-Arroyo, "A survey of routing protocols for energy constrained ad hoc wireless networks," in *Proceedings of the IEEE International Conference on Future Generation Communication and Networking (FGCN '07)*, pp. 522–527, December 2007.
- [8] S. Panichpapiboon, G. Ferrari, and O. K. Tonguz, "Optimal transmit power in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 10, Article ID 1683791, pp. 1432–1447, 2006.
- [9] B. Zhang and H. T. Mouftah, "Energy-aware on-demand routing protocols for wireless ad hoc networks," *Wireless Networks*, vol. 12, no. 4, pp. 481–494, 2006.
- [10] J. Li, D. Cordes, and J. Zhang, "Power-aware routing protocols in ad hoc wireless networks," *IEEE Wireless Communications*, vol. 12, no. 6, pp. 69–81, 2005.
- [11] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proceedings of the IEEE International Conference on Computer and Communications Societies*, vol. 3, pp. 1548–1557, April 2001.
- [12] Y.-C. Tseng and T.-Y. Hsieh, "An architecture for power-saving communications in a wireless mobile ad hoc network based on location information," *Microprocessors and Microsystems*, vol. 28, no. 8, pp. 457–465, 2004.
- [13] Y.-C. Tseng and T.-Y. Hsieh, "Fully energy-aware and location-aware protocols for wireless multihop ad hoc networks," in *Proceedings of the IEEE International Conference Computer Communications and Networks*, pp. 608–613, October 2002.
- [14] Y.-C. Wu and C.-C. Tuan, "Energy saving cache-based routing protocol in wireless ad hoc networks," in *Proceedings of the IET International Conference on Wireless, Mobile and Sensor Networks*, no. 533, pp. 466–469, December 2007.

- [15] D. Li, J. Zhou, F. Zhang, and J. Wang, "An efficient gateway election and location service in Ad Hoc networks," in *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC '08)*, pp. 252–256, April 2008.
- [16] Z. Wu, H. Song, S. Jiang, and X. Xu, "Energy-aware grid multipath routing protocol in MANET," in *Proceedings of the IEEE International Conference on Modelling and Simulation*, pp. 36–41, March 2007.
- [17] W.-H. Liao, J.-P. Sheu, and Y.-C. Tseng, "GRID: a fully location-aware routing protocol for mobile ad hoc networks," *Telecommunication Systems*, vol. 18, no. 1–3, pp. 37–60, 2001.
- [18] Y.-C. Wu and C.-C. Tuan, "Power saving routing protocol with power sieving in wireless ad hoc networks," in *Proceedings of the IEEE International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '09)*, pp. 349–352, April 2009.
- [19] Y.-C. Wu and C.-C. Tuan, "Triangular energy saving route protocol by energy sieving in wireless Ad hoc networks," in *Proceedings of the IEEE International Conference on Mobile Data Management Systems, Service and Middleware*, pp. 474–477, May 2009.
- [20] I. Stojmenovic and X. Lin, "Power-aware localized routing in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 11, pp. 1122–1133, 2001.
- [21] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th ACM/IEEE Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, August 2000.

Research Article

Intelligent Collaborative Event Query Algorithm in Wireless Sensor Networks

Rongbo Zhu

College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China

Correspondence should be addressed to Rongbo Zhu, rongbozhu@gmail.com

Received 15 June 2011; Accepted 2 August 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Rongbo Zhu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Event query processing is a very important issue in wireless sensor networks (WSNs). In order to detect event early and provide monitoring information and event query timely in WSNs, an efficient intelligent collaborative event query (ICEQ) algorithm is proposed, in which sensor nodes that are near to the boundary of events are selected to accomplish complex event monitoring and query processing through intelligent collaboration. ICEQ will select range-nearest neighbors as the basic components of surrounding nodes. Then it will identify the gaps between the surrounding nodes and try to select the nearest neighbor collaborative nodes for enclosing the event in the node selection phase, which can avoid redundant sensor nodes to join surrounding nodes via identifying a set of association surrounding nodes between the nearest sensor nodes and the query events. Detailed experimental results and comparisons with existed algorithm show that the proposed ICEQ algorithm can achieve better performance in terms of query-processing time, average number of selected collaborative nodes, and query message consumption.

1. Introduction

The rapid development in computing, sensing, and wireless communication technologies has made the availability of wireless sensor networks (WSNs) [1, 2]. Their low cost, small size, and untethered nature make them sense information at previously unobtainable resolutions [3]. WSNs can be deployed in battlefield applications, and a variety of vehicle health management, habitat monitoring, environment monitoring, and condition-based maintenance applications on industrial, military, space platforms [4, 5].

In WSNs, an important task is to monitor dynamic and unpredictable events. Since the sensor network can be viewed as a distributed database [6, 7], due to the distributed nature and resource constraint of WSNs, we cannot maintain a centralized index to support query processing in WSNs [8]. Meanwhile, because of limitations imposed by impoverished computing environment, data collection and query in WSNs must support an unusual set of software requirements. Several previous works [8, 9] proposed declarative SQL-like query which enable users to acquire the information about the network through issuing queries to the sink. Even

though each sensor node will be rather limited in terms of storage, processing, and communication capabilities, they will be able to accomplish complex event monitoring and query processing through intelligent collaboration, especially in large-scale WSNs.

Since sensor nodes have rigid energy constraints, it is hard to displace sensor nodes in the monitoring region [8, 10] due to unattended and untethered deployment. Most existing data collection systems [11–14] are query-based ones. Most existed event query schemes will decrease the lifetime of WSN greatly due to power consumption for the real-time monitoring. Traditional query-processing techniques of WSNs mainly deal with retrieving sensor node locations, sensing values, and aggregating the sensed values. However, in a lot of applications, users expect event and data information about areas of their interests. If a sensor is queried by many users, it may experience congestion and great power consumptions. Thus, a natural requirement is that each user sets a proper query range to both avoid overhead and achieve a global optimality at the same time.

In order to balance the inherent tradeoff between query reliability versus energy consumption in query-based

wireless sensor systems, an adaptive fault-tolerant quality of service (QoS) control algorithms based on hop-by-hop data delivery utilizing “source” and “path” redundancy is proposed in [11] to maximize the lifetime of the system. In order to allocate the multihop query range for each user such that a certain global optimality is achieved, Han et al. [15] investigated the NP complete scheme in its generic form and proposed a distributed heuristic to resolve the query problem. A data-querying scheme was proposed in WSN [16] where queries formed for each sensing task are sent to task sets, and the sensed data is retrieved from a sensor network in the level of detail specified by users, and a tradeoff mechanism between data resolution and query cost is provided. To disseminate data required for processing monitoring queries in a WSN, the notion of event-monitoring queries and algorithms were proposed in [17] for building and maintaining efficient collection trees that provide the conduit to minimize important resources such as the number of messages exchanged among the nodes or the overall energy consumption. In order to improve the performance of area query-processing in wireless sensor networks, an energy-efficient in-network area query processing scheme is proposed in [18], which partitioned the monitored area into grids and constructed a reporting tree to process merging areas and aggregations and conserve energy consumption. In [19], two approaches were proposed for processing such queries in WSN in-network instead of collecting all data at the base station of the spatiotemporal queries in WSN.

In order to improve the query performance, range nearest-neighbor (RNN) query [20], and nearest-surrounding (NS) queries [21], retrieve data based on location information in sensor networks. This kind of query schemes may enable us to find out surrounding nodes needed and is an efficient way to monitor event with less power consumption. In [22], a distributed Bayesian algorithm was proposed based on the concept of spatial correlation. However, it assuming that event measurements are either much larger or much smaller than normal measurements. In order to find out approximate real boundary, an efficient event query scheme is proposed in [23], which considers that WSN is composed of two distinct homogeneous regions. In order to achieve the boundary node efficiently, the localized fault-tolerant event boundary detection scheme was proposed in [24]. An efficient noise-tolerant event boundary detection algorithm was proposed in [25], which defined boundary nodes as sensor nodes which lie within real boundary with certain confidence interval guarantee. The problem of in-network processing and queries of trajectories of moving targets in a sensor network is investigated in [26], which exploits the spatial coherence of target trajectories for opportunistic information dissemination with no or small extra communication cost, as well as for efficient probabilistic queries searching for a given target signature in a real-time manner. In [27], collaborative query processing among multiple heterogeneous sensor networks was investigated and formulated into an optimization problem with respect to energy efficiency. WinyDB [28], a relational query-processing system on Windows CE-based personal

digital assistants (PDAs) for sensor networks, is proposed to improve both the energy efficiency and the data quality collaboratively. To overcome the faulty data query problem to improve the accuracy of data query, an efficient fault-tolerant event query algorithm (FTEQ) was proposed in [29], which takes the short-term and the long-term spatial and temporal similarities between sensors and environment into consideration to decrease faulty detection rate and data query cost.

Although a number of event query schemes have been proposed to improve the query performance in WSNs, event query processing is still a very challenging task due to its complexity and ill-posed nature, and all of these works do not comprehensively consider the correlation between sensors and environment. And the most existing research work has focused on data aggregation to provide efficient data transmission. The overhead of query processing is generally ignored with the assumption that query transmission contributes to only a small portion of overall data transmission in the sensor network. However, there are many cases where this assumption does not hold any more. Therefore, the methods mentioned above all use statistical methods to differentiate whether sensor nodes are boundary nodes or not.

Another problem is that existed work always assumes that the monitoring nodes are often interested in obtaining either the actual readings or their aggregate values; from sensor nodes that detect interesting events, the detection of such events can often be identified by the readings of each sensor node. In such scenarios, each sensor node is not forced to include its measurements in the query output at each epoch, but rather such query participation is evaluated on a per epoch basis, depending on its readings and the definition of interesting events. However, in actual complex environment, due to the characteristics of WSNs, sensors are usually deployed in a noneasily accessible or harsh environment, and sensors are prone to failure, and these faulty sensors are likely to report arbitrary data very different from the true environmental phenomenon, and the faulty data of sensors are very common, which greatly influence the accuracy of data query. Hence, how to select appropriate nodes to accomplish complex event monitoring and query processing through intelligent collaboration is an important task.

Motivated by the above reasons, an efficient intelligent collaborative event query (ICEQ) algorithm is proposed, in which sensor nodes that are near to the boundary of events are selected to accomplish complex event monitoring and query processing through intelligent collaboration. ICEQ includes initial phase and node selection phase. In initial phase, ICEQ will select range nearest-neighbors as the basic components of surrounding nodes. Then, it will identify the gaps between surrounding nodes and try to select nearest neighbor collaborative nodes for enclosing the event in node selection phase, which can avoid redundant sensor nodes to join the surrounding nodes via identifying a set of association surrounding nodes between the nearest sensor nodes and query events. The main contributions of ICEQ may be summarized as follows.

- (i) To retrieve a set of the nearest collaborative nodes of a specific event, ICEQ can identify a set of association surrounding nodes between the nearest sensor nodes and the query events that frequently appear in the system, which converts the demographic values and sensed data items presented in each query transaction into demographic types and event categories, respectively. Hence, ICEQ can select the nodes appropriately to decrease the number of selected nodes and prolong the lifetime of WSNs.
- (ii) ICEQ is able to identify where gaps exit between surrounding nodes by finding large or frequent demographic query itemsets of query, and then try to select proper collaborative nodes for enclosing the event with rule decision and computing confidence between rules. Hence, ICEQ can select the appropriately nodes according to the network topology and environment.

The rest of this paper is organized as follows. The proposed intelligent collaborative event query algorithm is given in Section 2. Performance studies are conducted in Section 3. This paper concludes with Section 4.

2. Proposed Algorithm

2.1. Problem Description. In a distributed WSN, assume that each sensor node s_i has a unique identity (ID) and is aware of their locations via global positioning system (GPS) devices. Each sensor node s_i has a fixed communication range c_i and a fixed sensing range r_i . And the communication range of a sensor node s_i follows unit disk graph model. Therefore, a sensor node s_i can communicate with a sensor node s_j if they are in each others' communication range. Otherwise, the sensing range of a sensor node s_i is also a disk and smaller than its communication ranges generally. The deployment of sensor nodes is random (or grid) and dense enough over a two-dimensional monitoring region. Euclidean distance is used as a metric to measure the distance between nodes. The Euclidean distance between any two nodes s_i and s_j is denoted by $d(s_i, s_j)$. The goal of the proposed intelligent collaborative event query (ICEQ) algorithm is to appropriately select a set of nearest collaborative nodes of a specific event. When given a set of rough boundary nodes B_S which are near to a real event boundary, an approximate boundary of the event can be obtained to bound the event region. Hence, ICEQ is to find out a set S of the nearest sensor nodes to such area that sensing ranges of adjacent nodes in S must be overlapping to enclose the event. In other words, adjacent sensor nodes s_i, s_j in S must satisfy the condition

$$d(s_i, s_j) < r_i + r_j, \quad s_i \in S, s_j \in S, \quad (1)$$

where $d(s_i, s_j)$ is the Euclidean distance between adjacent nodes s_i and s_j ; r_i and r_j are sensing ranges nodes; s_i and s_j individually.

In order to select the nodes appropriately, the proposed ICEQ algorithm will identify a set of association surrounding nodes between the nearest sensor nodes and the query events

that frequently appear in the system, which will consider the spatial and the temporal correlation between sensors and environment. Suppose there are k demographic attributes with domains being D_i ($i \in [1, k]$). Let $B = \{b_1, b_2, \dots, b_r\}$ be the set of sensed data items of sensor nodes. An aggregation hierarchy on the i th demographic attribute, denoted $H(D_i)$, is a tree with leaf nodes corresponding to the different D_i values and internal nodes representing groupings of D_i values. A taxonomy on B , denoted $H(B)$, is a tree with the set of leaves being equal to B , and internal nodes indicate sensor node categories. A link represents and is a relationship. To facilitate mining sensing data profile association rules, we group the aggregated sensor nodes of the same demographic information, resulting in a new type of query transaction called demographic query transaction. Specifically, the demographic query transaction of the i th query is represented as a tuple:

$$t_i = \langle d_{i,1}, d_{i,2}, \dots, d_{i,k}, b_{i,1}, b_{i,2}, \dots, b_{i,s} \rangle, \quad (2)$$

$$i \in [1, n], \quad k \geq 1, \quad s \geq 1,$$

where $d_{i,j}$ is a leaf in $H(D_j)$ that represents the j th demographic attribute value of the aggregated sensor nodes, and $b_{i,t}$ is a leaf in $H(B)$ that represents the sensed data items of sensor nodes that is the i th query.

Since the goal is to identify the associations between demographic types and event categories; the demographic values and sensed data items presented in each query transaction must be converted into demographic types and event categories, respectively, resulting in an extended query transaction. Here we include all demographic types of each demographic value and all sensed data categories of all item appeared in the sink node. Therefore, the i th query transaction can be translated to the extended query transaction:

$$t'_i = \langle d'_{i,1}, d'_{i,2}, \dots, d'_{i,u}, b'_{i,1}, b'_{i,2}, \dots, b'_{i,m} \rangle, \quad (3)$$

$$i \in [1, n], \quad u \geq 1, \quad m \geq 1,$$

where $d'_{i,j}$ and $b'_{i,j}$ are internal nodes in $H(D_j)$ and $H(B)$, respectively. Note that a demographic type could be a conjunction of several primitive demographic types. We use $d = (d_1, d_2, \dots, d_l)$ to denote a complex demographic type d' that is a conjunction of d_1, d_2, \dots, d_l . We say that the query transaction t_i supports a demographic type $d' = (d_1, d_2, \dots, d_l)$ if $\{d_1, d_2, \dots, d_l\} \subset t'_i$, where t'_i is the extended query transaction of t_i .

Similarly, we say that t_i supports a query event category c if $c \in t'_i$. A generalized profile association rule is an implication of the form $X \rightarrow Y$, where X is a demographic type and Y is a query event category. The rule $X \rightarrow Y$ holds in the query transaction set T with a confidence $c\%$ if c percent of the query transactions in T supports both X and Y . The rule $X \rightarrow Y$ also has support $s\%$ in the query transaction set T if s percent of the query transactions in T supports both X and Y . Therefore, given a set of query transactions T and several demographic aggregation hierarchies $H(D_j)$, $j \in [1, k]$, and one sensor taxonomy

$H(B)$, the problem of mining generalized profile association rules from query transaction data is to discover all rules that have support and confidence greater than the query-specified minimum support called Min_{sup} and minimum confidence called Min_{conf} . These rules are named strong rules.

2.2. Intelligent Collaborative Event Query Algorithm. The proposed ICEQ algorithm consists of two phases: initial phase and node selection phase. ICEQ will select range nearest neighbors as the basic components of surrounding nodes in initial phase. Node selection phase is to identify gaps between the rough surrounding nodes and then try to select proper surrounding nodes for monitoring the event by intelligent collaborative processing among nodes to decrease power consumption.

In the initial processing step, let Q denote a priority queue, let S_{E_i} denote a randomly selected end node, let $S_{E_i, \text{NN}}$ denote the nearest neighbour node of S_{E_i} , let L_i denote a query line, let S_{S, L_i} be the start node of a query line L_i , let $S_{S, L_i, \text{NN}}$ be the nearest neighbour node of S_{S, L_i} , let S_{E, L_i} be the end node of a query line L_i , let $S_{E, L_i, \text{NN}}$ be the nearest neighbour node of S_{E, L_i} , let A_{E, L_i} be the end node set of a query line L_i that it can divide L_i into several subsegments through these end nodes, let $A_{E, O}$ be a set of the end nodes covered by the spatial object O , let S_{E_i} denote a randomly selected end node, let D_{MAX} be the maximum distance, and let S be the results of event query. The initialization values of the parameters are as follows:

$$\begin{aligned} d(S_{E_i}, S_{E_i, \text{NN}}) &\leftarrow \infty, \\ S_{S, L_i, \text{NN}} &\leftarrow \text{NULL}, \\ S_{E, L_i, \text{NN}} &\leftarrow \text{NULL}, \\ S &\leftarrow \text{NULL}, \\ A_{E, O} &\leftarrow \text{NULL}, \end{aligned} \quad (4)$$

where $d(S_{E_i}, S_{E_i, \text{NN}})$ is the distance between S_{E_i} and $S_{E_i, \text{NN}}$; Q is initialized with root node.

In order to differentiate different segments of corresponding query-line nearest-neighbor nodes, end nodes of subsegments of a specified query line L_i are obtained by doing the intersection between a perpendicular bisector of current scanned nodes, neighboring LNN nodes, and L_i . Thus, for each endpoint S_{E_i} which belong to the query line L_i , all points of L in $[S_{E_i}, S_{E_{i+1}}]$ have the same nearest neighbor node defined as $S_{E_i, \text{NN}}$. It is possible that sensor nodes scanned later are much closer than sensor nodes for certain subsegments in the nearest neighbor node list. Therefore, it needs to check whether this sensor node covers some endpoints which are obtained by nodes previously scanned. If there is a currently scanned sensor s_j whose distance $d(s_j, S_{E_i})$ is smaller than $d(S_{E_i}, S_{E_i, \text{NN}})$ for some S_{E_i} , it means the end node S_{E_i} is covered by s_j . Since there are endpoints of subsegments obtained from intersection of the perpendicular bisector and the specified query line, the currently scanned sensor s_j is the nearest neighbor node. The algorithm proposed removes the end node S_{E_i} , adds

new end nodes S'_{E_i} by s_j , and updates $S'_{E_i, \text{NN}}$ accordingly. Also, a threshold D_{MAX} which determines the number of surrounding node candidates visited needs to be updated as maximum $d(S_{E_k}, S_{E_k, \text{NN}})$ of the current nearest neighbour list. Finally, we prepare a queue S to gather the results of the nearest neighbour lists and sort them counterclockwise with reference to the center of the approximate polygonal boundary of the event. These will be parts of the selection of our surrounding nodes of the event. And the proposed ICEQ algorithm is shown in Algorithm 1.

2.3. Collaborative Node Selection. The goal of collaborative node selection phase is to select proper sensors to enclose the event. From Algorithm 1, we can see that the rough nearest surrounding nodes of the event have been put in the selected nodes set S . Then, we construct neighbourhood relationships for each sensor node s_i in S first and find out the final collaborative nodes to monitor the event.

If we sort nodes in the queue S counter clockwise with reference to the centre point of the approximate polygonal boundary of the event, a sensor node s_i indexed i in the queue S sets its left-hand side neighbour as a sensor node indexed $i+1$ and its right-hand side neighbour as a sensor node indexed $i-1$ in the queue S .

In the phase, each sensor node s_i needs to keep information of their one-hop neighbors to construct neighborhood relationship. Each sensor node s_i will store its neighbors within communication range in its adjacency list. Because we only have partial results of nearest surrounding nodes of the event and these sensors are too few to enclose the event, there may be gaps between adjacent nodes with respect to their sensing ranges.

The proposed ICEQ algorithm will check whether gaps exit between this node and its adjacent neighbors in S . When a sensor node s_i ($s_i \in S$) is accessed, it first checks the distance $d(s_i, s_{i, \text{LH}})$ between s_i and its left-hand side neighbor $s_{i, \text{LH}}$. If $d(s_i, s_{i, \text{LH}})$ satisfies

$$d(s_i, s_{i, \text{LH}}) > r_{s_i} + r_{s_{i, \text{LH}}}, \quad (5)$$

which means that there is a gap between them.

A neighbor node s_k in adjacency lists selected as surrounding nodes must satisfy one of the following condition that:

$$\text{Min}(d(s_i, s_k) + d(s_k, s_{i, \text{LH}})), \quad (6)$$

$$\text{Min}(d(s_i, s_k) + d(s_k, s_{i, \text{RH}})). \quad (7)$$

We also construct neighborhood relationship for s_k as to two neighbors, s_i , and $s_{i, \text{LH}}$. Then, s_k will be inserted at the end of the queue S . Similarly, it will also check whether there is a gap between s_i and s_i 's right-hand side neighbor, $s_{i, \text{RH}}$, and select proper s_k to enclose it. This process will continue until all elements in S have been checked.

In order to select the appropriate node, we need to identify generalized profile association rules in the rough node set S , the itemsets that will interest us are of the following form $\langle d_{i_1}, d_{i_2}, \dots, d_{i_j}, b \rangle$, where d_{i_j} is an internal node in $H(D_{i_j})$ and b is an internal node in $H(B)$. By finding

```

Input: Rough event boundary  $B_E$ , and root node  $S_R$ .
Output: Selected nodes set  $S$ .
1: Initialization (4).
2: for each query line  $L_i$  of  $B_E$  do
3:   Dequeue node  $S_j$  from  $Q$ ;
4:   if ( $\min d(S_j, L_i) < D_{\max}$  and  $S_{S,L_i,NN}$  and  $S_{E,L_i,NN}$  are NULL) then
5:      $S_{S,L_i,NN} \leftarrow S_j, S_{E,L_i,NN} \leftarrow S_j, D_{\max} \leftarrow \min d(S_j, L_i)$ ;
6:   else
7:     Add an end node  $S'_{E_i}$  into event query line set  $A_{Q,L_i}$ ;
8:     Update  $d(S'_{E_i}, S'_{E_i,NN})$ ;
9:     Add an end node  $S'_{E_{i+1}}$  into event query line set  $A_{Q,L_i}$ ;
10:     $A_{E,S_j} \leftarrow \emptyset$ ;
11:   end if
12:    $S \leftarrow A_{Q,L_i}$ ;
13: end for
14: Collaborative node selection;
15: return  $S$ ;

```

ALGORITHM 1: Intelligent collaborative event query algorithm.

large or frequent demographic query itemsets, we can easily derive the corresponding generalized profile association rules.

Let F_k denote the frequent itemsets of the form $\langle d_{i_1}, d_{i_2}, \dots, d_{i_l}, b \rangle$. A candidate itemset C_{k+1} is generated by joining F_k and F_{k-1} , except that the k join attributes must include on query event type b , and the other $k-1$ demographic attributes types from $d_{i_1}, d_{i_2}, \dots, d_{i_l}$. We first extend each query transaction t_i as expressed in (1). The set of extended query transactions is denoted ET. After scanning the data set ET, we obtain large demographic 1-itemsets $L_1(D)$ and large event 1-itemsets $L_1(B)$. If an item is not a member of $L_1(D)$ or $L_1(B)$, it will not appear in any large demographic query itemset and is, therefore, useless. We delete all the useless items in every query transaction of ET in order to reduce its size. The set C_1 of candidate 1-itemsets is defined as $L_1(D) \times L_1(B)$. Data set ET is scanned again to find the set L_1 of large demographic query 1-itemsets from C_1 . A subsequent pass, say pass k , is composed of two steps. First, we use the above-mentioned candidate generation function to generate the set C_k of candidate itemsets by joining two large $(k-1)$ itemsets in F_{k-1} on the basis of their common $k-2$ demographic attribute values and the query attribute value. Next, data set ET is scanned and the support of candidates in C_k is counted. The set F_k of large k -itemsets are itemsets in C_k with minimum support.

Considering that some of the strong generalized profile association rules could be related to each other in either the demographic itemset part or the sensor nodes, and, therefore, the existence of one such rule could makes some others not interesting. To overcome the problem, let Π be the set of all demographic attribute types:

$$\Pi = \bigcup_{i=1}^k H(D_i). \quad (8)$$

We call a rule R_1 :

$$R_1 : D' \rightarrow b_1, \quad D' \subseteq \Pi, b_1 \in B. \quad (9)$$

a D-ancestor of another rule R_2 :

$$R_2 : D'' \rightarrow b_2, \quad D'' \subseteq \Pi, b_2 \in P, \quad (10)$$

if $b_1 = b_2$, and for all $d_1 \in D'$, there exists $d_2 \in D''$, such that d_1 is equal to or an ancestor of d_2 in the associated demographic concept hierarchy. Similarly, we call a rule $R_1 : D' \rightarrow b_1$ a P-ancestor of another rule $R_2 : D'' \rightarrow b_2$ if and is equal to or an ancestor of b_2 in the node taxonomy. Also, R_2 is called a D-descendant of R_1 if R_1 is a D-ancestor of R_2 . Note that D-descendant and B-descendant together form a lattice on the generalized profile association rules.

In context of collaborative nodes selection, we say a rule is valid if it can be used for making decision. Given a set of strong rules say Ω the candidate of a generalized profile association rule $R : D \rightarrow b \in \Omega$ is the confidence of the rule:

$$D - \bigcup_{i=1}^l D^i \rightarrow b - \bigcup_{i=1}^{l'} b_i \in \Omega, \quad (11)$$

where $R_i : D^i \rightarrow b, i \in [1, l]$ are the immediate D-descendants of R in Ω , and $R_i : D \rightarrow b_i, i \in [1, l']$ are the immediate B-descendants of R in Ω . Also, we say R is valid if the candidate of R is no less than Min_{conf} . From (11), we can see that the rule $R : D \rightarrow b$ is consulted only when we need to decide whether to select a node in $b - \bigcup_{i=1}^{l'} b_i$ to a query event with demographic type in $D - \bigcup_{i=1}^l D^i$. However, the difficulties of identifying candidate of a strong rule lie in computing the confidences of its DB-deductive rule.

Let the immediate D-descendants of R be $R_i : D^i \rightarrow b, i \in [1, l]$; $D - \bigcup_{i=1}^l D^i \rightarrow b$ let be called the D-deductive rule of R .

Let the immediate B-descendants of R be $R_i : D \rightarrow b_i, i \in [1, l']$, and $D \rightarrow b - \bigcup_{i=1}^{l'} b_i$ let be called the B-deductive rule of R .

Suppose that we have obtained the confidences of both the D-deductive rule and the B-deductive rule of a given rule R . Let $E\text{-Conf}(\text{DB-deductive rule}|\text{D-deductive rule})$ be the

estimated confidence of R , DB-deductive rule given R ; let D -deductive rule and $E_Conf(\text{DB-deductive rule}|\text{B-deductive rule})$ be the estimated confidence of R 's DB-deductive rule given R , B-deductive rule. We have

$$\begin{aligned}
& \text{Conf}(\text{DB-deductive}) \\
& \leq E_Conf(\text{DB-deductive rule} | D\text{-deductive rule}) \\
& = \text{Conf}\left(D - \bigcup_{i=1}^l D^i \longrightarrow b\right) \times \frac{|b - \bigcup_{i=1}^l b_i|}{|B|}, \\
& \text{Conf}(\text{DB-deductive}) \\
& \leq E_Conf(\text{DB-deductive rule} | B\text{-deductive rule}) \\
& = \text{Conf}\left(D \longrightarrow b - \bigcup_{i=1}^l b_i\right). \tag{12}
\end{aligned}$$

Therefore, we define the estimated interestingness of R , denoted $E_Interest(R)$, which is the estimated confidence of R 's DB-deductive rule, to be

$$\begin{aligned}
& E_Interest(R) \\
& = \text{Min}\left\{\text{Conf}\left(D - \bigcup_{i=1}^l D^i \longrightarrow b\right) \right. \\
& \quad \left. \times \frac{|b - \bigcup_{i=1}^l b_i|}{|b|}, \text{Conf}\left(D \longrightarrow b - \bigcup_{i=1}^l b_i\right)\right\}. \tag{13}
\end{aligned}$$

We approximate the confidence of a D -deductive rule by using the following theoretic results.

Lemma 1. Let D^i be mutually disjoint, and the confidence of $D - \bigcup_{i=1}^l D^i \longrightarrow b$ be

$$\begin{aligned}
& \text{Conf}\left(D - \bigcup_{i=1}^l D^i \longrightarrow b\right) \\
& = \frac{\text{sup}(D, b) - \sum_{i=1}^l \text{sup}(D^i, b)}{\text{sup}(D) - \sum_{i=1}^l \text{sup}(D^i)} \\
& = \frac{\text{sup}(D, b) - (\text{sup}(D', b) + \dots + \text{sup}(D^l, b))}{\text{sup}(D) - (\text{sup}(D') + \dots + \text{sup}(D^l))}. \tag{14}
\end{aligned}$$

Proof. Since D^i ($i \in [1, l]$) are disjoint, $\text{sup}(\bigcup_{i=1}^l D) = \sum_{i=1}^l \text{sup}(D^i)$.

Similarly, $\text{sup}(\bigcup_{i=1}^l D, b) = \sum_{i=1}^l \text{sup}(D^i, b)$.

Therefore,

$$\begin{aligned}
& \text{Conf}\left(D - \bigcup_{i=1}^l D^i \longrightarrow b\right) \\
& = \frac{\text{sup}(D, b) - \sum_{i=1}^l \text{sup}(D^i, b)}{\text{sup}(D) - \sum_{i=1}^l \text{sup}(D^i)} \\
& = \frac{\text{sup}(D, b) - (\text{sup}(D', b) + \dots + \text{sup}(D^l, b))}{\text{sup}(D) - (\text{sup}(D') + \dots + \text{sup}(D^l))}. \tag{15}
\end{aligned}$$

□

Theorem 2. Without loss of generality, let D^i ($i \in [1, l]$) be mutually disjoint. Assume that $\text{Conf}(\sum_{j=i+1}^l D^j - \sum_{j=1}^i D^j \longrightarrow b) \geq \text{Min}_{\text{conf}}$. If $D - \bigcup_{i=1}^l D^i \longrightarrow b$ has sufficient confidence, we have

$$\frac{\text{sup}(D, b) - (\text{sup}(D', b) + \dots + \text{sup}(D^l, b))}{\text{sup}(D) - (\text{sup}(D') + \dots + \text{sup}(D^l))} \geq \text{Min}_{\text{conf}}. \tag{16}$$

Proof. Let $D_1 = \sum_{j=1}^i D^j$, and let $D_2 = \sum_{j=i+1}^l D^j - D_1$. Obviously, D_1 and D_2 are disjoint. Since both $D_2 \longrightarrow b$ and $D - (D_1 \cup D_2) \longrightarrow b$ have sufficient confidences, we have

$$\frac{\text{sup}(D_2, b)}{\text{sup}(D_2)} \geq \text{Min}_{\text{conf}}, \tag{17}$$

and, by Lemma 1,

$$\frac{\text{sup}(D, b) - \text{sup}(D_1, b) - \text{sup}(D_2, b)}{\text{sup}(D) - \text{sup}(D_1) - \text{sup}(D_2)} \geq \text{Min}_{\text{conf}}. \tag{18}$$

By adding the denominators and numerators, respectively, from the left-hand sides of the two equations, we can obtain

$$\frac{\text{sup}(D, b) - \text{sup}(D_1, b)}{\text{sup}(D) - \text{sup}(D_1)} \geq \text{Min}_{\text{conf}}. \tag{19}$$

Since D^i ($i \in [1, l]$) are mutually disjoint, we have

$$\frac{\text{sup}(D, b) - (\text{sup}(D', b) + \dots + \text{sup}(D^l, b))}{\text{sup}(D) - (\text{sup}(D') + \dots + \text{sup}(D^l))} \geq \text{Min}_{\text{conf}}. \tag{20}$$

□

Now we discuss how to compute the confidence of a B-deductive rule $R_B : D \longrightarrow b - \bigcup_{i=1}^l b_i$. The query transactions that support R_B must have included nodes that fall outside $\bigcup_{i=1}^l b_i$. We say node categories b_i and b_j are siblings if they have a common parent in the respective concept hierarchy. Let $\text{NST}(D, b_i)$ denote the set of query transactions that support (D, b_i) but do not support any sibling of R_B . To calculate $\text{NST}(D, b_i)$, we associate a flag f_{NST} on each node category of an extended query transaction. $\text{NS}(b, \text{et})$, where b is a query category and et is an extended query transaction,

```

Input: A queue  $S$  which contains RNN results.
Output: A set  $S$  of selected collaborative nodes.
1: for each  $s_i$  ( $s_i \in S$ ) do
2:   Find out strong rules  $\Omega$ ;
3:   Calculate sufficient confidence with (14) and (16);
4:   if (sufficient confidence of  $s_i \geq \text{Min}_{\text{conf}}$ ) then
5:     if ( $d(s_i, s_{i,\text{LH}}) > r_{s_i} + r_{s_{i,\text{LH}}}$ ) then
6:       Choose  $s_k$  from  $s_i$  and  $s_{i,\text{LH}}$  adjacency list with (6);
       Construct neighbourhood relationship for  $s_k$ ;
       Insert  $s_k$  at the end of  $S$ ;
7:     else if ( $d(s_i, s_{i,\text{RH}}) > r_{s_i} + r_{s_{i,\text{RH}}}$ ) then
8:       Choose  $s_k$  from  $s_i$  and  $s_{i,\text{RH}}$  adjacency list with (7);
       Construct neighbourhood relationship for  $s_k$ ;
       Insert  $s_k$  at the end of  $S$ ;
9:     end if
10:  end if
11: end for

```

ALGORITHM 2: Collaborative node selection.

is equal to 1 if there exists no sibling of b in et and 0 otherwise. Therefore, we have

$$\text{NSSup}(D, b_i) = \frac{\sum_{et \text{ supports}(D, b_i)} \text{NS}(b_i, et)}{n}, \quad (21)$$

where n is the total number of query transactions.

If $r : D \rightarrow b - \bigcup_{i=1}^l b_i$ has sufficient confidence, we can get

$$\frac{\text{sup}(D, b) - \sum_{i=1}^l \text{NSSup}(D, b_i)}{\text{sup}(D)} \geq \text{Min}_{\text{conf}}. \quad (22)$$

$\text{NSSup}(D, b_i)$ for a demographic node itemset (D, b_i) can be computed when counting the support for (D, b_i) by expression (21). Expression (22) shows that $\text{sup}(D, b) - \sum_{i=1}^l \text{NSSup}(D, b_i) / \text{sup}(D)$ is an upper bound of the confidence of $D \rightarrow b - \bigcup_{i=1}^l b_i$. Therefore, if the upper bound is less than Min_{conf} , we drop the rule because it cannot have sufficient confidence, and consequently R is considered not interesting.

At last, we report all sensor nodes in S as nearest surrounding nodes of the event. And the proposed collaborative node selection algorithm is shown in Algorithm 2.

3. Simulation Results

3.1. Simulation Setup. In order to evaluate the performance of the proposed ICEQ algorithm, we implemented the ICEQ in the well-known simulation tool NS-2 [30]; the range nearest neighbor (RNN) query algorithm [20] is simulated as discussed here. There are 5000 sensor nodes deployed in our monitoring region. The shape of the event that occurs in the monitoring region is a circle. The approximate polygonal boundary of the event can be obtained via the boundary nodes of the event. Thus, the deployment strategy totally ensures the assumption that there is no communication hole in the network. And the system generates critical and

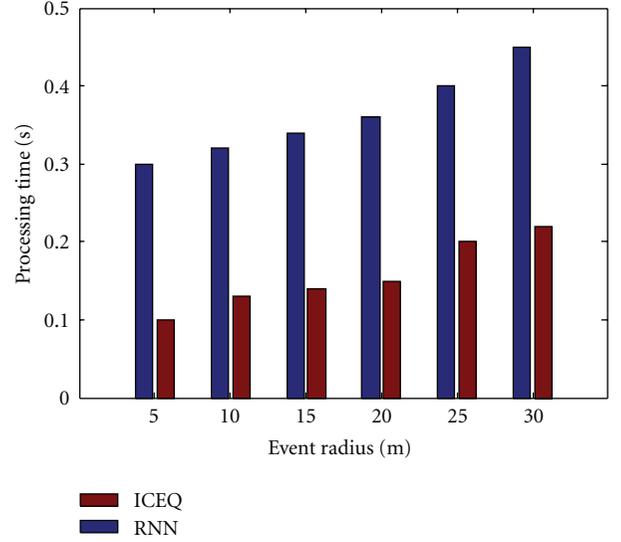


FIGURE 1: Processing time with varying event radius in grid topology.

noncritical events randomly. The performance analysis has been done by deploying variable number of nodes on the fixed squared area, to verify the effect of node densities on the data gathering path length and average number of hop counts. The deployment of sensor nodes is dense enough so that we can find out surrounding nodes of the event. And we set the Min_{conf} as 0.6. Two kinds of deployment, grid distribution and random distribution, are applied individually for comparison. There are three metrics used to compare the performance of proposed methods which are described as follows: query processing time, selected numbers of collaborative nodes, and total message consumption. Our simulation results are all from the average of 1000 runs.

3.2. Validation in Different Range of Event. In the first scenario, we vary the size of the event with varying its radius from 5 m to 30 m. Figures 1 and 2 show the query-processing time and average number of selected nodes of different algorithms, respectively.

As shown in Figure 1, it is observed that the proposed ICEQ outperforms in terms of query-processing time irrespective of the event radius. For ICEQ, the query processing is less than 0.23 s when the event radius, increases, while for RNN, the query processing time is higher than 0.3 s. The reason is that RNN needs to search RNNs edge by edge according to the approximate polygonal boundary of the event, and the cost of RNN is essentially higher than ICEQ. It is noticed that the event radius has some impact on the query-processing time for both ICEQ and RNN. It is reasonable since larger event radius means that more nodes will be evaluated in node selection. So when we increase the size of the event, the cost to find out surrounding nodes of the event will raise accordingly.

Figure 2 shows the numbers of selected nodes for the monitoring event. With the number of event radius increasing, the number of selected nodes of two schemes will

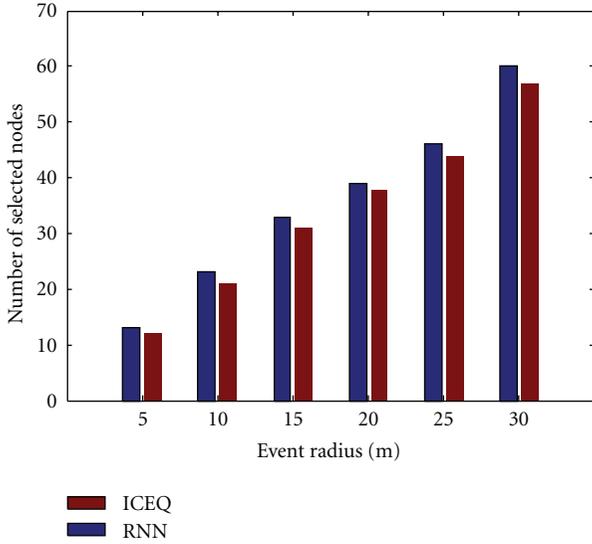


FIGURE 2: Number of selected nodes with varying event radius in grid topology.

increase obviously. When the event radius increases from 5 m to 30 m, the number of selected nodes of RNN increases to 61, which is slightly larger than that of ICEQ. We can see that the number of selected nodes of ICEQ always is slightly less than that of RNN. The reason is that the objective of the ICEQ is to minimize the selected nodes for the given constrained conditions. So in collaborative node selection phase, ICEQ considers the sufficient confidence and rules between nodes, which will avoid redundant sensor nodes to join surrounding nodes via identifying a set of association surrounding nodes between nearest sensor nodes and query events.

Figures 3 and 4 show the query-processing time and average number of selected nodes of different algorithms in random topology, respectively.

From Figure 3, we can see the similar results as in Figure 1. And the proposed ICEQ outperforms in terms of query-processing time irrespective of the event radius. For ICEQ, the query processing is less than 0.5 s, and it will increase slightly when the event radius increases. While for RNN, the query-processing time is obviously higher than that of ICEQ. When the event radius increases, the query-processing time of RNN will increase suddenly. And the query-processing time will larger than 2.5 s. It is noticed that the event radius has great impact on the query-processing time for RNN. The reason is the RNN needs to search RNNs edges, which depends on the network topology. Hence, the cost of RNN is essentially higher than ICEQ. For ICEQ, the reason of processing time increasing slightly is that ICEQ will visit more candidates with the radius increasing.

The results of number of selected nodes of different algorithms with the varying event radius are shown in Figure 4. As the event radius increases, the number of selected nodes of different algorithms will increase obviously. Especially for RNN, the number of selected nodes increases to 57. The main reason is that RNN selects surrounding

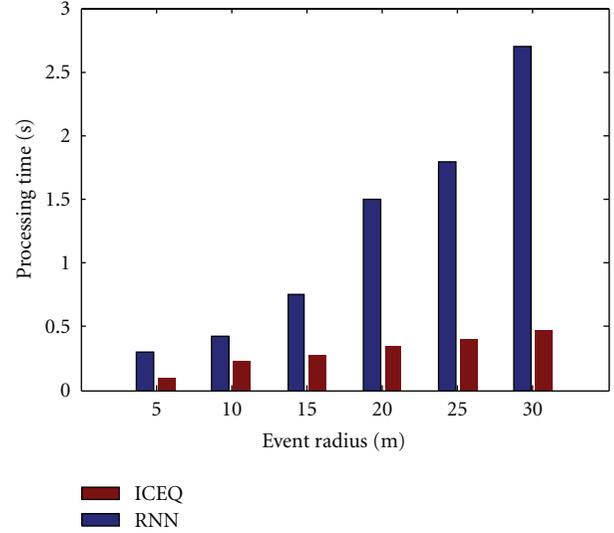


FIGURE 3: Processing time with varying event radius in random topology.

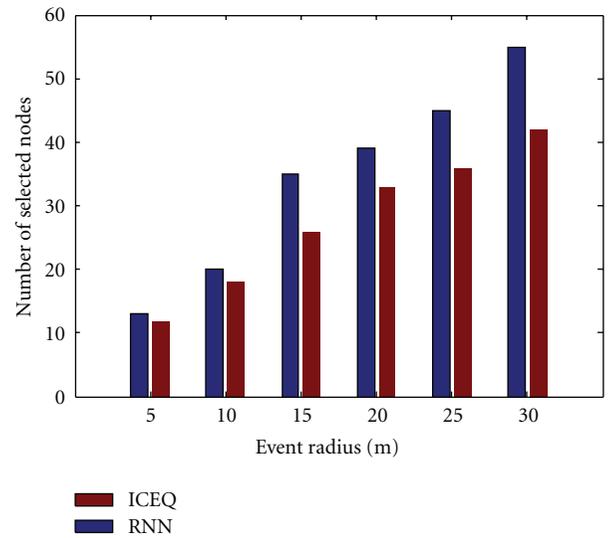


FIGURE 4: Number of selected nodes with varying event radius in random topology.

nodes by searching according to the approximate polygonal boundary of the event, which leads to select more nodes. So the event radius has a great influence on RNN. The event detection ratio of RNN is higher than that of ICEQ clearly. Because ICEQ can identify the gaps between surrounding nodes and try to select nearest neighbor collaborative nodes for enclosing the event in node selection phase, which can avoid redundant sensor nodes to join surrounding nodes via identifying a set of association surrounding nodes between the nearest sensor nodes and the query events. For the proposed ICEQ, the number of selected nodes is less than that of RNN. For example, when the event radius increases to 30m, the number of selected nodes of ICEQ increases to 42. Compared with RNN, ICEQ tries to select proper collaborative nodes for enclosing the event with rule

decision and computing confidence between rules. And the collaborative node selection scheme of ICEQ also helps to decrease the number of selected nodes.

3.3. Validation in Different Query Lines. In this scenario, we vary query lines in random topology. And sensor nodes are deployed in random topology. The radius of the circular event is fixed at 15 m, and we assume that it occurs arbitrarily in the monitoring region so that we obtain different number of edges of the approximate polygonal boundary. Figures 5 and 6 show the query-processing time and the average number of selected nodes of different algorithms with varying query lines in random topology, respectively.

The results of query-processing time of different algorithms with the varying number of query lines are shown in Figure 5. As the number of query lines increases, the query-processing time of different algorithms will increase. Especially for RNN, the query-processing time increases to 0.98 s when the number of query lines increases to 15. The main reason is that RNN needs to search edges according to the approximate polygonal boundary of the event, which leads to longer delay time and increases query-processing time. So the number of query lines has a great influence on RNN. For the ICEQ, the query-processing time is obviously lower than that of RNN. When the number of query lines increases to 15, the query processing time of ICEQ only increases to 0.19 s, which is less than that of RNN 0.79 s. Another phenomenon is that the number of query lines has no much effect on the ICEQ algorithm. The query-processing time almost keeps in the range of 0.15 s to 0.2 s. The reason is that ICEQ will determine the number of surrounding node candidate to traverse in the search process. Therefore, ICEQ can adaptively select appropriate collaborative nodes to decide the number of surrounding node candidates to form the approximate polygonal boundary. We also notice that the advantage of ICEQ over RNN becomes more evident when the number of query lines becomes large.

Figure 6 shows the number of selected nodes by different schemes when the number of query lines varies from 10 to 15. It can be seen that the number of selected nodes of two schemes oscillates slightly as the number of query lines increases. For RNN, the average number of selected nodes increases slightly, from 20 (10 query lines) to 23 (15 query lines). The reason is that RNN must find out the edge and the search surrounding nodes, which will increase the selected nodes when query lines increase. However, ICEQ always outperforms RNN due to the concurrent use of the rule mining among nodes and collaborative nodes. For instance, it achieves average 18.4% nodes saving compared with RNN scheme. Less selected nodes lead to longer network lifetime since sensors can turn to the power-saving mode once the event monitoring in their region is done.

3.4. Validation in Query Consumption. In this scenario, we investigate the total message consumption with varying network size from 100 to 10000 sensor nodes. Figures 7 and 8 show the query total message consumption by varying network size from 100 to 10000 sensor nodes. The duration of each query is set by 300 s and number of event types.

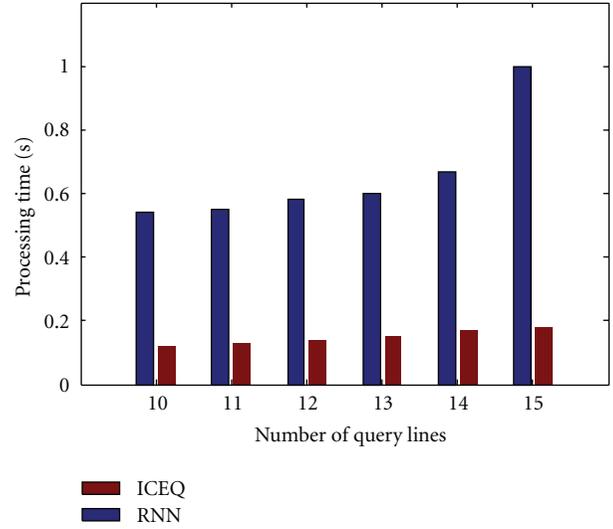


FIGURE 5: Processing time with varying query lines in random topology.

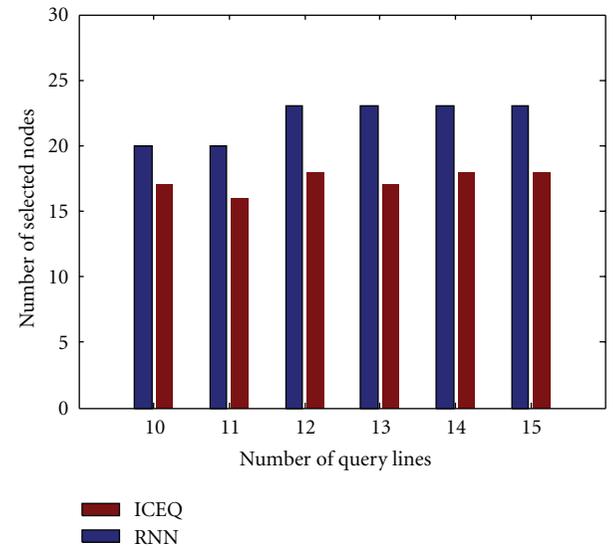


FIGURE 6: Number of selected nodes with varying query lines in random topology.

In Figure 7, the total number of messages of two mechanisms increases with the network size. The network size impacts on a number of messages significantly in RNN because the number of sensor nodes selected increases as the network size grows. In ICEQ, only a few numbers of reply messages exist so that the network size impact ICEQ a little due to the increasing network size. As the network size is small, the total number of messages of ICEQ is smaller than RNN. This is because overhead rose from the change of rough event boundary node larger than the benefit obtained from the nodes due to small network size. That is because network size is large enough, and the benefit of surrounding nodes is cost effective. Hence, ICEQ works efficiently in a large-scale WSN. Moreover, in ICEQ, the

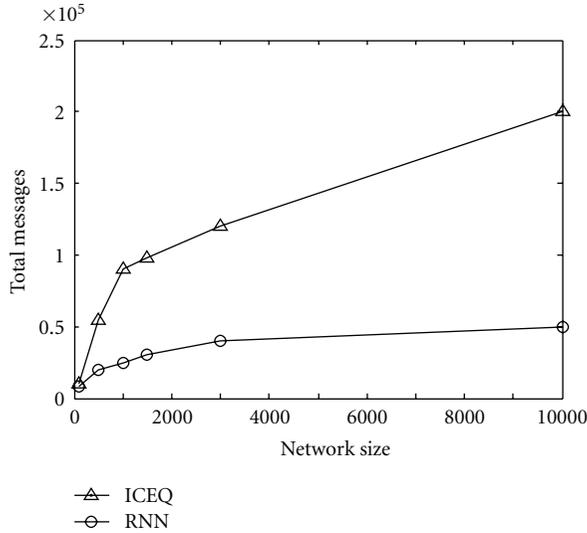


FIGURE 7: Total messages with varying network size.

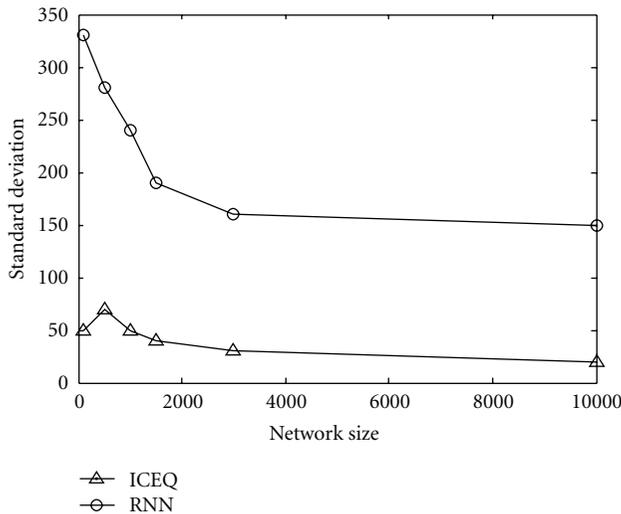


FIGURE 8: Standard deviation with varying network size.

involvement reduces the duplicated data packets, making that ICEQ outperforms RNN in a large-scale network.

Figure 8 compares the degree of power balance of all mechanisms in various network sizes. Each sensor node has the same probability for detecting event data, and the network traffic is uniformly distributed over the whole WSN. When the network size is smaller than 500, standard deviation of RNN is small. This is because that the network size is too small, and the sensor nodes intend to select rough event boundary nodes; therefore, the difference of query event on nodes is small. However, as the network size increases, the RNN sensor nodes that are close to event boundary have higher traffic than nodes in other location, and; therefore, their standard deviations is higher than ones of ICEQ mechanisms as the network size is 300 nodes. However, as the network size increases larger than 400, the standard deviations of RNN decrease significantly because

the number of detected event is fixed and the event detected is distributed in the whole WSN. The ICEQ mechanism obtains a smaller standard deviation as network size is larger than 500. The results also validate the effectiveness of the proposed ICEQ in power consumption.

4. Conclusion

In this paper, we presented an efficient intelligent collaborative event query (ICEQ) algorithm to detect the event early and provide monitoring information and event query timely in WSNs. ICEQ can identify a set of association surrounding nodes between the nearest sensor nodes and the query events that frequently appear in the system, which converts the demographic values, and sensed data items presented in each query transaction into demographic types and event categories, respectively. Hence, ICEQ can select the nodes appropriately to decrease the number of selected nodes and prolong the lifetime of WSNs. ICEQ is able to identify where gaps exit between surrounding nodes by finding large or frequent demographic query itemsets of query, and then try to select proper collaborative nodes for enclosing the event with rule decision and computing confidence between rules. Hence, ICEQ can select the appropriately nodes according to the network topology and environment. Through ICEQ, we can select a set of surrounding nodes of the event instead of all the sensor nodes in the monitoring region to check if there is any event evolution. Therefore, sensor nodes which are not surrounding nodes can enter into sleep modes temporarily to save their battery energies and thus extend the lifetime of sensor networks. The future work will focus on the issues of query moving objects and track objects in WSNs.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 60902053), the Science and Technology Research Planning of Educational Commission of Hubei Province of China (no. B20110803), and the Natural Science Foundation of Hubei Province of China (no. 2008CDB339). The author also gratefully acknowledges the helpful comments and suggestions of the reviewers.

References

- [1] Y. Rachlin, R. Negi, and P. K. Khosla, "The sensing capacity of sensor networks," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1675–1691, 2011.
- [2] C. Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.
- [3] H. Ying, M. Schlösser, A. Schnitzer et al., "Distributed intelligent sensor network for the rehabilitation of Parkinson's patients," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 2, pp. 268–276, 2011.
- [4] O. Chipara, C. Lu, J. A. Stankovic, and G.-C. Roman, "Dynamic conflict-free transmission scheduling for sensor network queries," *IEEE Transactions on Mobile Computing*, vol. 10, no. 5, pp. 734–748, 2011.

- [5] W. Yu, T. N. Le, J. Lee, and D. Xuan, "Effective query aggregation for data services in sensor networks," *Computer Communications*, vol. 29, no. 18, pp. 3733–3744, 2006.
- [6] G. He, R. Zheng, I. Gupta, and L. Sha, "A framework for time indexing in sensor networks," *ACM Transactions on Sensor Networks*, vol. 1, no. 1, pp. 101–133, 2005.
- [7] G. S. Iwerks, H. Samet, and K. P. Smith, "Maintenance of k-nn and spatial join queries on continuously moving points," *ACM Transactions on Database Systems*, vol. 31, no. 2, pp. 485–536, 2006.
- [8] J. Gehrke and S. Madden, "Query Processing in Sensor Networks," *IEEE Pervasive Computing*, vol. 3, no. 1, pp. 46–55, 2004.
- [9] R. Kannan, S. Sarangi, and S. S. Iyengar, "Sensor-centric energy-constrained reliable query routing for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 839–852, 2004.
- [10] J. Guang and N. Silvia, "Towards spatial window queries over continuous phenomena in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 4, pp. 559–571, 2008.
- [11] I.-R. Chen, A. P. Speer, and M. Eltoweissy, "Adaptive fault-tolerant QoS control algorithms for maximizing system lifetime of query-based wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 2, pp. 161–176, 2011.
- [12] M. Demirbas, X. Lu, and P. Singla, "An in-network querying framework for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 8, pp. 1202–1215, 2009.
- [13] O. Chipara, C. Lu, J. A. Stankovic, and G.-C. Roman, "Dynamic conflict-free transmission scheduling for sensor network queries," *IEEE Transactions on Mobile Computing*, vol. 10, no. 5, pp. 734–748, 2011.
- [14] R. Zhu, Y. Qin, and J. Wang, "Energy-aware distributed intelligent data gathering algorithm in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 235724, 13 pages, 2011.
- [15] B. Han, J. Leblet, and G. Simon, "Query range problem in wireless sensor networks," *IEEE Communications Letters*, vol. 13, no. 1, pp. 55–57, 2009.
- [16] E. Cayirci, V. Coskun, and C. Cimen, "Querying sensor networks by using dynamic task sets," *Computer Networks*, vol. 50, no. 7, pp. 938–952, 2006.
- [17] V. Stoumpos, A. Deligiannakis, Y. Kotidis, and A. Delis, "Processing event-monitoring queries in sensor networks," in *Proceedings of the 24th International Conference on Data Engineering (ICDE '08)*, pp. 1436–1438, April 2008.
- [18] C. Ai, L. Guo, Z. Cai, and Y. Li, "Processing area queries in wireless sensor networks," in *Proceedings of the 5th International Conference on Mobile Ad-hoc and Sensor Networks (MSN '09)*, pp. 1–8, December 2009.
- [19] M. Bestehorn, K. Böhm, E. Buchmann, and S. Kessler, "Energy-efficient processing of spatio-temporal queries in wireless sensor networks," in *Proceedings of the 18th ACM International Conference on Advances in Geographic Information Systems, (ACM SIGSPATIAL GIS '10)*, pp. 340–349, November 2010.
- [20] H. Hu and D. L. Lee, "Range nearest-neighbor query," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 78–91, 2006.
- [21] K. C. K. Lee, W.-C. Lee, and H. V. Leong, "Nearest surround queries," in *22nd International Conference on Data Engineering (ICDE '06)*, pp. 85–95, April 2006.
- [22] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *Sigmod Record*, vol. 31, no. 3, pp. 9–18, 2002.
- [23] Y. Xu, W.-C. Lee, J. Xu, and G. Mitchell, "Processing window queries in wireless sensor networks," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*, pp. 658–675, April 2006.
- [24] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized fault-tolerant event boundary detection in sensor networks," in *Proceedings of the 24th IEEE International Conference of Computer and Communications Society (INFOCOM '05)*, vol. 2, pp. 902–913, 2005.
- [25] J. Guang and S. Nittel, "NED: an efficient noise-tolerant event and event boundary detection algorithm in wireless sensor networks," in *Proceedings of the 7th International Conference on Mobile Data Management (MDM '06)*, pp. 151–153, May 2006.
- [26] D. Zhou and J. Gao, "Opportunistic processing and query of motion trajectories in wireless sensor networks," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 1197–1205, April 2009.
- [27] Y. He, M. Li, Y. Liu, J. Zhao, W. L. Huang, and J. Ma, "Collaborative query processing among heterogeneous sensor networks," in *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 25–30, May 2008.
- [28] T. W. Chiu and Q. Luo, "Collaboratively querying sensor networks through handheld devices," in *Proceedings of the IEEE International Conference on Mobile Data Management, (MDM '07)*, pp. 30–35, May 2007.
- [29] R. Zhu, "Efficient fault-tolerant event query algorithm in distributed wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2010, Article ID 593849, 7 pages, 2010.
- [30] The Network Simulator. NS-2, <http://www.isi.edu/nsnam/ns/>.