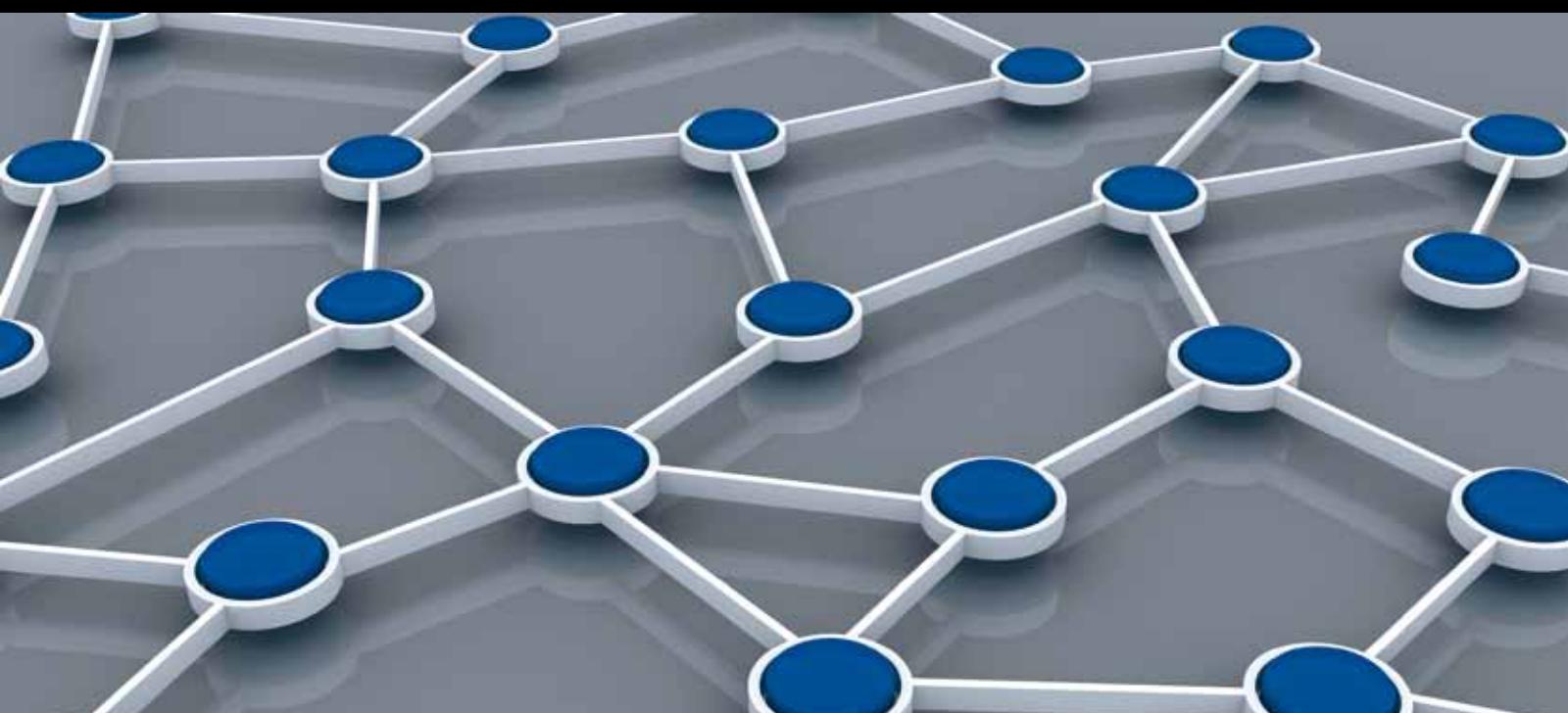


DISTRIBUTED MOBILE SENSOR NETWORKS FOR HAZARDOUS APPLICATIONS

GUEST EDITORS: FRANK EHLERS, DON SOFGE, MANDAR CHITRE, AND JOHN POTTER





Distributed Mobile Sensor Networks for Hazardous Applications

Distributed Mobile Sensor Networks for Hazardous Applications

Guest Editors: Frank Ehlers, Don Sofge, Mandar Chitre, and John Potter



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	Minglu Li, China	Hairong Qi, USA
Richard R. Brooks, USA	Jing Liang, China	Joel Rodrigues, Portugal
Stefano Chessa, Italy	Weifa Liang, Australia	Jorge Sa Silva, Portugal
W.-Y. Chung, Republic of Korea	Wen-Hwa Liao, Taiwan	Sartaj K. Sahni, USA
George P. Efthymoglou, Greece	Alvin S. Lim, USA	Weihua Sheng, USA
Frank Ehlers, Italy	Zhong Liu, China	Zhi Wang, China
Paola Flocchini, Canada	Donggang Liu, USA	Sheng Wang, China
Yunghsiang S. Han, Taiwan	Yonghe Liu, USA	Andreas Willig, New Zealand
Tian He, USA	Seng Loke, Australia	Qishi Wu, USA
Baoqi Huang, Australia	Jun Luo, Singapore	Qin Xin, Norway
Chin-Tser Huang, USA	J. R. Martinez-deDios, Spain	Jianliang Xu, Hong Kong
S. S. Iyengar, USA	Shabbir N. Merchant, India	Yuan Xue, USA
Rajgopal Kannan, USA	Aleksandar Milenkovic, USA	Fan Ye, USA
Miguel A. Labrador, USA	Eduardo F. Nakamura, Brazil	Ning Yu, China
Joo-Ho Lee, Japan	Peter Csaba Ölveczky, Norway	Tianle Zhang, China
Shijian Li, China	M. Palaniswami, Australia	Yanmin Zhu, China
Yingshu Li, USA	Shashi Phoha, USA	
Shuai Li, USA	Cristina M. Pinotti, Italy	

Contents

Distributed Mobile Sensor Networks for Hazardous Applications, Frank Ehlers, Don Sofge, Mandar Chitre, and John Potter
Volume 2012, Article ID 970831, 3 pages

Wireless Communication Networks for Gas Turbine Engine Testing, Xuewu Dai, Konstantinos Sasloglou, Robert Atkinson, John Strong, Isabella Panella, Lim Yun Cai, Han Mingding, Ang Chee Wei, Ian Glover, John E. Mitchell, Werner Schiffers, and Partha S. Dutta
Volume 2012, Article ID 212876, 18 pages

The Effects of Motion on Distributed Detection in Mobile Ad Hoc Sensor Networks, Xusheng Sun and Edward J. Coyle
Volume 2012, Article ID 460924, 14 pages

Modelling and Performance Analysis of a Network of Chemical Sensors with Dynamic Collaboration, Alex Skvortsov and Branko Ristic
Volume 2012, Article ID 656231, 9 pages

Energy-Balanced Density Control to Avoid Energy Hole for Wireless Sensor Networks, Jie Jia, Jian Chen, Xingwei Wang, and Linliang Zhao
Volume 2012, Article ID 812013, 10 pages

Linear Decision Fusion under the Control of Constrained PSO for WSNs, Sisi Jiang, Zhiwen Zhao, Sheng Mou, Zushun Wu, and Yi Luo
Volume 2012, Article ID 871596, 11 pages

Capacity-Preserved, Energy-Enhanced Hybrid Topology Management Scheme in Wireless Sensor Networks for Hazardous Applications, A. Jawahar, S. Radha, and R. Sharath Kumar
Volume 2012, Article ID 280932, 12 pages

Constructing a Distributed AUV Network for Underwater Plume-Tracking Operations, Stephanie Petillo, Henrik Schmidt, and Arjuna Balasuriya
Volume 2012, Article ID 191235, 12 pages

Editorial

Distributed Mobile Sensor Networks for Hazardous Applications

Frank Ehlers,¹ Don Sofge,² Mandar Chitre,³ and John Potter⁴

¹ Research Department for Underwater Acoustics and Marine Geophysics, Bundeswehr Technical Centre for Ships and Naval Weapons, Naval Technology and Research (WTD 71), Klausdorfer Weg 2, 24148 Kiel, Germany

² Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, 4555 Overlook Avenue S.W., Washington, DC 20375, USA

³ Acoustic Research Laboratory, National University of Singapore, 21 Lower Kent Ridge Road, Singapore 119077

⁴ Systems Technology Department, NATO Undersea Research Centre (NURC), Viale S. Bartolomeo 400, 19126 La Spezia, Italy

Correspondence should be addressed to Frank Ehlers, drfrank.ehlers@gmail.com

Received 8 February 2012; Accepted 8 February 2012

Copyright © 2012 Frank Ehlers 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.

1. Introduction

A key justification for the use of robotic systems is for applications that are dull, dirty, or dangerous, where the use of robotic systems may reduce or eliminate the danger to humans working in an environment known to be hazardous.

Hazards may include nuclear, biological, or chemical contaminants, flood, fire, or earthquake damage as well as underwater, under-ice, down mines, and space exploration.

Hazardous applications include rescue missions, dealing with hostile targets, addressing potential terrorist threats, or any operation performed in a hazardous environment.

Distributed networks of robots may be deployed in such hazardous environments to effectively perform missions such as reconnaissance, surveillance, search and recovery, or resource harvesting. When mobile robots are used to host sensor and communication nodes for a distributed sensing network, the sensor network may achieve improved capabilities for detection and classification of potential targets and other objects of interest.

Intervention begins with gathering information about the nature of the problem. Sensor fusion in the context of distributed sensor networks has emerged as a leading method for implementing robust and efficient surveillance.

The “designer” of such distributed sensing systems has simple rules for the behavior of individual team members on one end and the overall objective function of the entire team at the other end. Observed in biological swarms, emergent behaviors are able to link these two ends, as also engineering approaches for the design of “Systems of Systems” can do.

For hazardous environments and applications, this special issue contains papers describing concepts, simulations, and real applications. The focus lies on sensor networks that have the additional features of control parameters in order to optimize the sensor and fused data quality. Implementing the resulting adaptive behavior into the robots lets the network advance towards its high-level goal.

2. Fusion and Performance Evaluation

S. Jiang et al. propose a new Linear Decision Fusion Algorithm under the Control of Constrained-PSO for WSNs. They state that a major application of a distributed WSN (Wireless Sensor Network) is to monitor a specific area for detecting some events such as disasters and enemies. In order to achieve this objective, each sensor in the network is required to collect local observations which are probably corrupted by noise, make a local decision regarding the presence or absence of an event, and then send its local decision to a fusion center. After that, the fusion center makes the final decision depending on these local decisions and a decision fusion rule, so an efficient decision fusion rule is extremely critical. It is obvious that the decision-making capability of each node is different owing to the dissimilar signal noise ratios and some other factors, so it is easy to understand that a specific sensor’s contribution to the global decision should be constrained by this sensor’s decision-making capability, and based on this idea, we establish a novel linear decision fusion model for WSNs. The authors employ the constrained

particle swarm optimization (constrained-PSO) algorithm to control the parameters of this model. They also apply the typical penalty function to solve the constrained-PSO problem. The emulation results indicate that their design is capable to achieve a very high accuracy.

A. Skvortsov and B. Ristic discuss the modelling and performance analysis of a network of chemical sensors with dynamic collaboration. The problem of environmental monitoring using a wireless network of chemical sensors with a limited energy supply is considered. Since the conventional chemical sensors in active mode consume vast amounts of energy, an optimisation problem arises in the context of a balance between the energy consumption and the detection capabilities of such a network. A protocol based on “dynamic sensor collaboration” is employed: in the absence of any pollutant, majority of sensors are in the sleep (passive) mode; a sensor is invoked (activated) by wakeup messages from its neighbors only when more information is required. The authors propose a mathematical model of a network of chemical sensors using this protocol. The model provides valuable insights into the network behavior and near optimal capacity design (energy consumption against detection). An analytical model of the environment, using turbulent mixing to capture chaotic fluctuations, intermittency and nonhomogeneity of the pollutant distribution, is employed in their study. A binary model of a chemical sensor is assumed (a device with threshold detection). The outcome of their study is a set of simple analytical tools for sensor network design, optimisation, and performance analysis.

X. Sun and E. J. Coyle address the effects of motion on distributed detection in mobile ad hoc sensor networks. A large set of mobile wireless sensors observe their environment as they move about. The authors consider the subset of these sensors that each made observations about a brief, localized event at the time when near that location. As the sensors continue to move, one of them eventually finishes processing its observations, decides that an event of interest occurred, and wants to determine if other sensors confirm its results. This sensor thus assumes the role of a Cluster-Head (CH) and requests that all other sensors that collected observations at that time/location reply to it with their decisions. The motion of the sensors since the observation time determines how many wireless hops their decision must cross to reach the CH. The authors analyze the effect of this motion in the 1D case by modeling each sensor’s motion as a Correlated Random Walk (CRW), which can account for realistic transient behavior, geographical restrictions, and nonzero drift. The authors also account for observation errors and errors in each hop in the wireless channel. Quantities, such as the error probability of the final decision at the CH and the minimum energy required to collect the local decisions from all relevant sensors, can then be directly calculated as functions of time and the parameters of the CRW, the measurement noise, and the channel noise. These results thus allow a rapid characterization of the time dependence of distributed detection algorithms that are being executed in realistic mobile sensor networks.

3. Energy and Topology Optimization

J. Jia et al. present theoretical analysis and simulations of energy balanced density control to avoid energy hole for wireless sensor networks. Density control is of great relevance for wireless sensor networks monitoring hazardous applications where sensors are deployed with high density. Due to the multihop relay communication and many-to-one traffic characters in wireless sensor networks, the nodes closer to the sink tend to die faster, causing a bottleneck for improving the network lifetime. The authors investigate systematically the theoretical aspects of the network load and the node density. Furthermore, the authors prove the accessibility condition to satisfy that all the working sensors exhaust their energy with the same ratio. By introducing the concept of the equivalent sensing radius, a novel algorithm for density control to achieve balanced energy consumption per node is thus proposed. Different from other methods in the literatures, a new pixel-based transmission mechanism is adopted, to reduce the duplication of the same messages. Combined with the accessibility condition, nodes on different energy layers are activated with a nonuniform distribution, so as to balance the energy depletion and enhance the survival of the network effectively. Extensive simulation results are presented to demonstrate the effectiveness of the new algorithm.

A. Jawahar et al. present a new capacity-preserved, energy-enhanced hybrid topology management scheme in wireless sensor networks for hazardous applications. A wireless sensor network is composed of large number of sensor nodes which are densely deployed in the field. These nodes monitor the environment, collect the data, and route it to a sink. The main constraint is that the nodes in such a network have a battery of limited stored energy, and if the nodes start to die, the network lifetime gets reduced. There are various topology management schemes such as SPAN, STEM, GAF, and BEES, for improving network parameters such as capacity, lifetime, coverage, and latency. None of these schemes will improve all the mentioned network parameters. In Sustainable Physical Activity in Neighbourhood (SPAN), some of the nodes become coordinators to form the backbone path and can only forward messages. A noncoordinator will check periodically whether it should become coordinator. SPAN preserves network capacity, decreases latency but provides less energy savings. Sparse Topology and Energy Management scheme (STEM) improves network lifetime by putting the sensor nodes either in monitoring state or in transfer state. STEM does not try to preserve capacity resulting in great energy savings and high latency. In the scheme proposed by the authors, a new coordinator rule is implemented in SPAN, and then integrated with STEM. The authors observe that the energy conserved increases by about 3.18% to 4.17% without sacrificing network capacity. Due to definite path in the proposed scheme the latency is reduced by almost half the latency of STEM scheme.

4. Applications

X. Dai et al. apply techniques for wireless communication networks for gas turbine engine testing. A new trend

in the field of aeronautical engine health monitoring is the implementation of wireless sensor networks (WSNs) for data acquisition and condition monitoring to partially replace heavy and complex wiring harnesses, which limit the versatility of the monitoring process as well as creating practical deployment issues. Augmenting wired with wireless technologies will fuel opportunities for reduced cabling, faster sensor and network deployment, increased data acquisition flexibility, and reduced cable maintenance costs. However, embedding wireless technology into an aero engine (even in the ground testing application considered here) presents some very significant challenges, for example, a harsh environment with a complex RF transmission channel, high sensor density, and high data rate. The authors discuss the results of the Wireless Data Acquisition in Gas Turbine Engine Testing (WIDAGATE) project, which aimed to design and simulate such a network to estimate network performance and de risk the wireless techniques before the deployment.

S. Petillo et al. investigate the construction of a distributed AUV network for underwater plume-tracking operations. In recent years, there has been significant concern about the impacts of offshore oil spill plumes and harmful algal blooms on the coastal ocean environment and biology as well as on the human populations adjacent to these coastal regions. Thus, it has become increasingly important to determine the 3D extent of these ocean features (“plumes”) and how they evolve over time. The ocean environment is largely inaccessible to sensing directly by humans, motivating the need for robots to intelligently sense the ocean for us. The authors propose the use of an autonomous underwater vehicle (AUV) network to track and predict plume shape and motion, discussing solutions to the challenges of spatiotemporal data aliasing (coverage versus resolution), underwater communication, AUV autonomy, data fusion, and coordination of multiple AUVs. A plume simulation is also developed as the first step toward implementing behaviors for autonomous, adaptive plume tracking with AUVs, modeling a plume as a sum of Fourier orders, and examining the resulting errors. This is then extended to include plume forecasting based on time variations, and future improvements and implementation are discussed.

Acknowledgments

The guest editors of this special issue are much indebted to their authors and reviewers for their effort and dedication to make this issue a reality.

*Frank Ehlers
Don Sofge
Mandar Chitre
John Potter*

Research Article

Wireless Communication Networks for Gas Turbine Engine Testing

Xuewu Dai,¹ Konstantinos Sasloglou,² Robert Atkinson,² John Strong,³ Isabella Panella,³ Lim Yun Cai,⁴ Han Mingding,⁴ Ang Chee Wei,⁴ Ian Glover,² John E. Mitchell,¹ Werner Schiffers,⁵ and Partha S. Dutta⁶

¹Department of Electronic and Electrical Engineering, University College London, London WC1E 7JE, UK

²Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G1 1XW, UK

³SELEX Galileo, Basildon SS14 3EL, UK

⁴Institute for Infocomm Research, A*STAR, Singapore

⁵Strategic Research Centre, Rolls-Royce plc, P.O. Box 31, Derby DE24 8BJ, UK

⁶Advanced Technology Centre, Rolls-Royce Singapore Pte Ltd., 16 International Business Park, No. 03-01 M+W Zander, Singapore 609929

Correspondence should be addressed to John E. Mitchell, jmitchel@ee.ucl.ac.uk

Received 1 June 2011; Revised 28 September 2011; Accepted 20 October 2011

Academic Editor: Frank Ehlers

Copyright © 2012 Xuewu 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.

A new trend in the field of Aeronautical Engine Health Monitoring is the implementation of wireless sensor networks (WSNs) for data acquisition and condition monitoring to partially replace heavy and complex wiring harnesses, which limit the versatility of the monitoring process as well as creating practical deployment issues. Augmenting wired with wireless technologies will fuel opportunities for reduced cabling, faster sensor and network deployment, increased data acquisition flexibility, and reduced cable maintenance costs. However, embedding wireless technology into an aero engine (even in the ground testing application considered here) presents some very significant challenges, for example, a harsh environment with a complex RF transmission channel, high sensor density, and high data rate. In this paper we discuss the results of the Wireless Data Acquisition in Gas Turbine Engine Testing (WIDAGATE) project, which aimed to design and simulate such a network to estimate network performance and derisk the wireless techniques before the deployment.

1. Introduction

Wireless sensors are increasingly used for monitoring structures and machinery. A large number of such systems exist already on the market [1]. Most systems comprise a relatively small number of nodes with low date rates; however, there are clear signs that wireless sensor technology is maturing [2]. The work described in this document explores a wireless sensor system for monitoring vital parameters during aero gas turbine engine development tests with a long-term aim to do the same during engine on wing operation. A typical engine test phase requires measurements of up to 3000 parameters from transducers on the engine connected to the data acquisition system through very long cables. These wired data acquisition systems require as much as 12 km of wiring and

involve long and expensive setup and instrumentation times which significantly increases time to market.

Despite these limitations, wired instrumentation is a mature and well-understood approach used widely in the aero industry. Replacing it with wireless solutions will require significant changes in not only the technology but also in the associated engineering processes. In the absence of sufficient know-how about the performances of wireless sensors for engine test data acquisition, replacing the existing instrumentation process is fraught with risks. In this context, the WIDAGATE project has developed robust and experimentally validated simulations of WSNs to generate insights into their performance for engine testing applications. This project aims to provide the aero engine testing industry the tools to conduct an effective risk-benefit trade-off analysis and

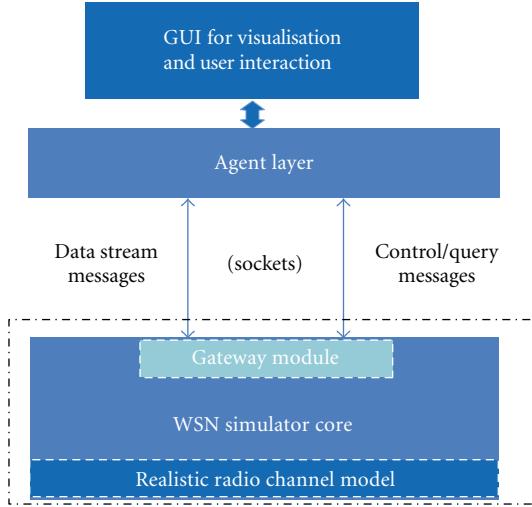


FIGURE 1: System structure of the software simulator developed.

support intelligent investment choices regarding WSN-based instrumentation.

Specific advantages of using WSN-based instrumentation are in the replacement of part of the wiring infrastructure with wireless communication to offer significant benefits in cost and time, flexibility, interoperability, weight, and improved robustness. To achieve these goals a number of long-term challenges need to be addressed, in particular, the issue of communicating in the harsh and dynamic environment of gas turbines involving high-speed rotations, rapid airflows, high temperatures, and large-amplitude vibrations. In terms of wireless communication, the WSNs in the application of engine testing face four challenges:

- (1) severe RF interference;
- (2) non-line-of-sight propagation. RF signals are transmitted in an environment that is largely composed of metal and it is highly possible that nodes are not in line-of-sight;
- (3) high-density sensors. Engine testing may eventually involve up to a few thousands sensors;
- (4) high-data-rate and near real-time transmission requirement with accurate synchronisation.

The WIDAGATE project delivers an application-specific diagnostic tool for network performance, network architecture, and communication protocols analysis in a relatively short time scale (i.e., one run of engine testing), whilst also addressing many generic, long-term WSN research challenges. The main achievement of the WIDGAGTE project is the development of both an accurate and experimentally validated simulation model and a system demonstrator of a wireless sensor network for data gathering and health monitoring during gas turbine engine testing.

As shown in Figure 1, the developed software simulation platform is comprised of three parts: the realistic radio channel model, the wireless network simulator core, and the Agent layer for user interaction and optimisation. The

impacts of the complex engine testing environment on wireless communication are modelled by an empirical radio channel model. The radio channel model is integrated into an event-based Wireless Sensor Network (WSN) simulator core. The Agent layer provides the friendly graphical user interface to visualise the network performance, allows users to both access and control the underlying simulator core on-the-fly (e.g., querying sensor nodes for information, change the network architecture by the repositioning of nodes, etc.), and performs multiagent optimisation to improve network performances (such as network throughput).

This paper is organised as follows: Section 2 gives details of the application considered within the WIDAGATE project and details related work. Section 3 gives an overview of the WIDAGATE system design. Section 4 presents details of the modelling of the radio channel in engine testing environment. The Medium Access Control (MAC) protocols and the software simulator of WSN are described in Section 5, followed by Section 6 presenting the design and implementation of a multiagent application layer which allows end-users to interact and control the developed WSN simulator and the optimisation of network performance. Section 7 presents the development of a hardware evaluation test bed and the results of both lab tests and engine tests, with Section 8 presenting the conclusions of the work.

2. Application Scenario

Data sampling and transmission of the samples are key issues when developing a new instrumentation system. Generally, there are two kinds of schemes for data sampling and transmission. (1) Offline transmission: the measurement data are sampled and stored at the sensor nodes during the course of engine testing and transmitted to the data logger (referred to as data SINK or data concentrator) at the end of the testing. The advantage of this two-step scheme is the simplicity of the communication system while a disadvantage is the large storage requirement and non-real-time operation. (2) Online transmission: the sensor nodes sample the physical signals and immediately transmit the sampled data through the wireless network. Although the online method can reduce the storage required at each sensor node, it puts a real-time requirement of the communication network, demanding higher throughput and lower latency.

The features of the engine testing and the requirements to the wireless communication system are listed as follows.

Periodic Traffic Load. The periodic sensor measurements generate a periodic data flow from the sensors to the collectors. This implies a schedule-based MAC approach, in order to effectively exploit this pattern to maximise performance. Some spatial correlation between the sensor measurements is expected. The MAC protocol must have the ability to manage the local data communication in a manner that enables the available data redundancy to be exploited.

Near Real-Time Requirement/Latency Requirement. This project does not attempt to provide real-time operational

TABLE 1: Environmental conditions during development and production testing.

Oil system temperature	250°C
Air temperature (beneath core cover)	350°C
Metal temperature	1100–1300°C
Pressure	40–42 bar
Vibration	40 g

data; however, rapid delivery of results and timing accuracy of the data are vital for acceptable operation.

High Sensor Density. During engine development testing, there are over three thousands sensors (1000 thermocouples, 1500 pneumatic lines, and 500 accelerometers) required to measure and record the temperature, pressure, and vibration, respectively. Most of the sensors are deployed in the limited space around or within the engine which leads to a high density of sensor points.

High Data Rate and High Spectral Efficiency MAC Protocol. The periodically generated sensor data and high density of sensors result in a huge amount of data to be transmitted across the network. Thus, a high data rate is necessary to achieve near real-time and low latency operation. Maximising the system spectral efficiency (throughput per unit bandwidth per unit area) in the multihop sensor network is essential to minimising latency and maximising energy efficiency. Therefore, a high system spectral efficiency MAC protocol is required that minimises data forwarding delay between the tiers of the network hierarchy and maximise the number of sensors communicating simultaneously.

Scalability Requirement. The number of sensors and their location is fixed throughout the engine test. However, in development and production testing, a small number of sensors may be added later.

Harsh Environment. Development and production testing takes place under the extreme environmental conditions summarised in Table 1.

In addition to the extreme vibration and temperature environment within which the wireless communication network has to work, interferences due to other industrial electrical/electronic devices may also have an adverse impact on the performance of the wireless link [3].

Robustness and Coexistence Requirement. The communication protocol has to be designed carefully to make the wireless communication robust enough against interference and enable it to coexist with other electrical equipment.

With the recent advances in wireless sensor networks (WSNs), the realisation of low-cost embedded industrial automation systems has become feasible [1, 4]. Small-scale condition monitoring using wireless technologies for engine testing and in-service engine monitoring are well discussed and demonstrated in [2, 5], where the Bluetooth techniques

are adopted for small networks. In [6], a Bluetooth-based demonstrator with 5 nodes connecting the thermocouples and sound sensors has been developed for acquisition and visualisation of the engine's temperature. A wireless sensor network for monitoring the health of aircraft engines is described in [7, 8]. In [9] the authors provide an overview of the architectures of wireless networks for engine and aircraft health monitoring.

It has been shown that the MAC protocol dominates the network performance and, recently, many researchers have been engaged in developing schemes that address the unique challenges of industrial wireless sensor networks. A number of MAC protocols have been proposed for wireless sensor networks. The most common MAC is contention-based channel access, namely, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), in which nodes transmit data if the medium is sensed idle and use a back-off mechanism in case of busy channel or collisions. Both IEEE 802.11 and the IEEE 802.15.4 are based on the CSMA/CA. However, CSMA/CA is not optimal to handle real-time applications with high data rates, various priority levels, and Quality of Service (QoS) requirements. As congestion increases, contention-based MACs spent most time on back-off to avoid collision and the bandwidth and energy are wasted. Although the IEEE 802.11 standard defines a centralized polling-based channel access method, the Point Coordination Function (PCF), to support time-bounded services, this contention-free approach is based on the contention-based DCF (Distributed Coordination Function) and thus is not efficient due to inefficient polling and a large overhead.

In contrast to the distributed contention-based MAC, centralised channel access can avoid collisions and reduce the amount of time used for backing off, making it more appropriate for a real-time high data throughput applications. Synchronous MAC protocols based on Time Division Multiple Access (TDMA) have attracted considerable interest because of their collision-free operation, higher spectrum efficiency, and low power consumption. While the medium access is coordinated by a controlled schedule, the collisions are avoided and the node's duty cycle can be optimised so that sensor nodes may place themselves in sleep mode for a longer time without sensing the medium. A TDMA-like protocol (called MaCARI) was proposed for industrial wireless sensor networks in OCARI project [10]. A polling-based TDMA MAC protocol with duty cycles is proposed for industrial applications and its performance is analysed in [11]. It has been shown that the polling-based MAC protocol is a special case of TDMA and shows a better performance in terms of scalability and self-organization.

3. Network Design

Considering the features described in Section 2, a hybrid wireless/wired data gathering architecture is considered suited for high data rate engine testing. Figure 2 illustrates a typical configuration of engine testing, where the whole engine measurement system is comprised of a number of modules with each having its own communication system to transmit the data collected within it. It is natural to divide

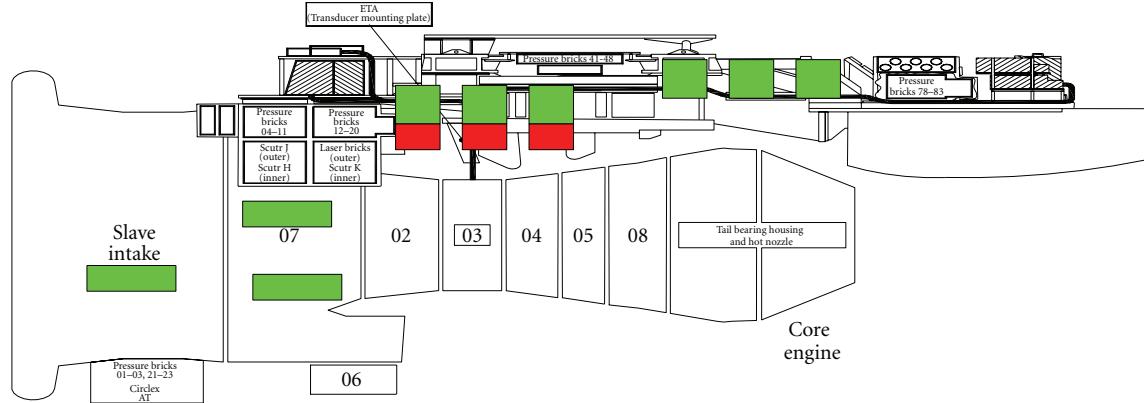


FIGURE 2: Typical engineering configuration, with permission of Rolls-Royce.

the sensors into subsets allocated at different engine modules and, in turn, the whole communication network is comprised of a set of linear cluster networks, which is ideal for providing communications in systems that have relay stations deployed along a line. As a result, the wireless sensors are hierarchically organised into clusters using the tree-cluster architecture which has been adopted in the recent standard specifications (e.g., the 802.15.4 standard [12] and the ZigBee Alliance specifications [13]).

As shown in Figure 3, the proposed linear cluster network (LCN) consists of three kinds of devices, *Sensor Nodes* (SNs), *Cluster Heads* (CHs), and *Wired Sinks* (WSs), and they are hierarchically organised into clusters. A sensor node connects to a set of transducers (channels) and samples the physical signals periodically. A group of SNs which are close to each other geographically or related in terms of physical measurements comprises a cluster. Each cluster has a special node as its own CH and the SNs transmit their data to the associated CH via a single-hop communication link. Usually, the CH would have more computation capability and storage space than the SNs and a SN may be associated with one more cluster to improve the reliability and scalability. Furthermore, one cluster may be too far to communicate directly with the wired sink (WS) by single hop. In this case, the CHs relay the data received from the child SNs to the WS, either directly or via a multihop path through other intermediate CHs.

In the proposed LCN topology, these clusters and their CHs are organised in a chain with each chain being deployed along the surface of the engine module. As shown in Figure 3, the whole LCN comprises of multiple lines of linear clusters in parallel and each chain, referred to as line (i.e., line 1, line 2, ...), consists of a set of cluster heads and a WS. The linear cluster structure is a combination of star and mesh topology. In such a convergecast network there are two kinds of communication, namely, single-hop SN-to-CH communication and multihop CH-to-CH communication. From the proposed engine testing application perspective, in order to achieve a higher network throughput, it is reasonable to make use of all available radio channels, while we assume that the CH nodes are equipped with a double-radio wireless module. The double-radio module potentially uses two

different standards and different frequency bands to avoid cochannel interference between SN-to-CH and CH-to-CH communications. This independency enables us to simulate and study the behaviour of the sensor-to-CH and CH-to-CH communication independently. In this paper, we focus on the SN-to-CH communication and study the performances of CSMA/CA and polling in SN-to-CH. With some minor modification, these protocols can be applied to CH-to-CH communication.

Another benefit of the linear cluster network is the simplification of routing protocols, which helps to reduce the communication overhead, save CPU time and energy consumption, and improve the robustness and life time. As the SN always sends the data to its associated CH in a single-hop manner, there is no need for routing. The multihop routing functions are needed only within the cluster heads. Since the LCN has a linear topology, the routing protocol is simplified to a great extent. It is worth noting that, depending on the transmission power, the interference range of CHs may cover the whole network. Thus, the routing problem turns into a media access problem and is solved by a joint design of MAC-routing protocols. The performance of the routing protocol in CH-to-CH communication depends on the underlying MAC protocols which is the focus of WIDAGATE.

4. Modelling the Radio Environment

A prerequisite to the engineering of WSN in any environment is a physical-layer wireless channel model that can be used to predict the channel characteristics. In the context of WIDAGATE this implies the channel between any pair of nodes lying on a gas turbine engine surface. Such models may be narrowband or broadband. The former is simpler and appropriate if the dispersion of the channel is small compared to the symbol duration of the signals which the channel will carry. The latter is more complex but must be used if dispersion is a significant fraction of symbol duration. The geometry of a gas turbine engine is essentially cylindrical and an estimate of the maximum data rate that can be properly accommodated by a narrowband channel model (i.e., flat fading without equalisation) can be made by considering

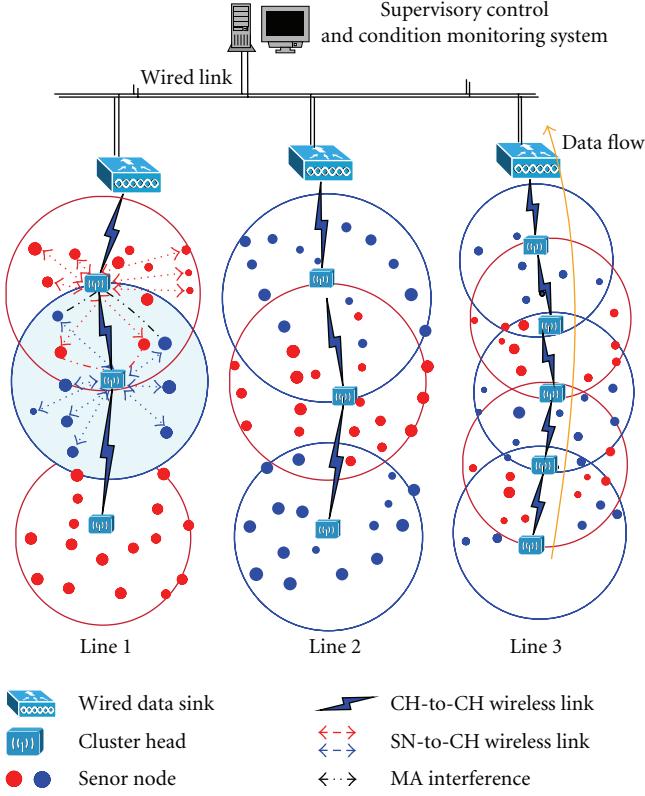


FIGURE 3: Linear cluster-tree topology (linear cluster network).

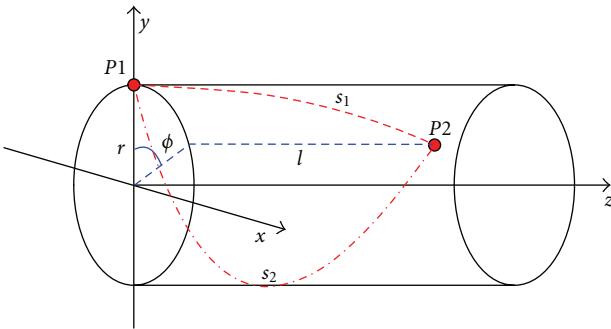


FIGURE 4: Schematic diagram of two-path geometry.

two-path propagation between a pair of nodes as shown in Figure 4.

The maximum differential path length for any pair of nodes is given by

$$\Delta s = s_2 - s_1 = \sqrt{[r(2\pi - \phi)]^2 + l^2} - \sqrt{(r\phi)^2 + l^2}, \quad (1)$$

where r is engine radius, ϕ is angular separation between P_1 and P_2 projected onto a plane perpendicular to the engine axis, and l is the separation of the planes perpendicular to the engine axis containing P_1 and P_2 . And the corresponding differential propagation delay for a propagation velocity is $\Delta T = (1/c)(\sqrt{l^2 + (2\pi r)^2} - l)$. Assuming (worst case) binary modulation and that time dispersion must be not greater

than 10% of the symbol duration, the maximum bit-rate is given by

$$R_b \leq \frac{0.1}{\Delta T} = \frac{0.1c}{\sqrt{[r(2\pi - \phi)]^2 + l^2} - \sqrt{(r\phi)^2 + l^2}}. \quad (2)$$

Choosing extreme values of $\phi = 0$, $r = 1.0$ m, and $l = 2.0$ m then R_b is limited to 6.53 Mbit/s. Since this is greater than the bit-rate envisaged from each SN in the WIDAGATE application a narrowband channel model is appropriate. This suggests that the multipath in engine testing environment has negligible influence on the link quality and thus there is no need for a multipath model. Nevertheless, besides the path loss, the thermal noise and environmental interference are also taken into account in our model (as shown later in this section) to give a better link quality simulation [3].

Channel Measurements. The scattering transmission parameter S_{21} was measured across the ISM frequency band (2.4–2.5 GHz) between pairs of points distributed over a rectangular grid on the cylindrical surface of a Gnome gas turbine engine. This particular frequency band was selected based on the majority of WSN devices currently available. The measurements were made between a pair of low-gain (approximately 0 dB) omnidirectional microstrip antennas using an Agilent N5230A vector network analyser both in the absence of (dataset 1) and in the presence of (dataset 2) an engine cowling. The arrangements of measurement grid points for the two datasets are shown schematically in Figure 5.

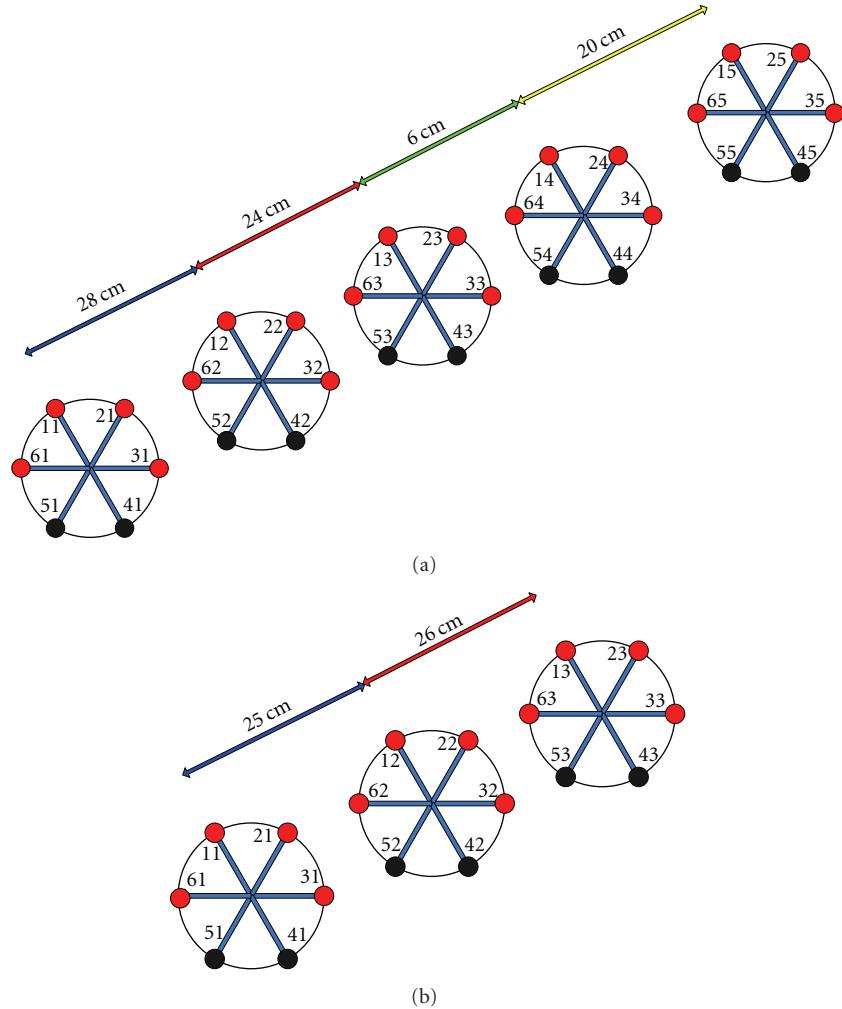


FIGURE 5: Schematic diagram of engine measurement points: (a) without cowling: dataset 1 and (b) with cowling: dataset 2. Black (darker) points omitted.

Dataset 1 was obtained in the Gnome Test Laboratory at Rolls-Royce in Derby, UK. The engine for these measurements had no cowling (Figure 5(a)). Six potential measurement points were distributed evenly around the circumference of the engine in five planes perpendicular to the engine axis along the length of the engine. The separation between adjacent planes was 28 cm, 24 cm, 6 cm, and 20 cm. There were, therefore, 30 potential measurement points in total. Since measurement time was limited, and since measurements between all pairs of potential points include redundant geometries, two points on each plane were omitted from the measurement process. These are represented by hollow circles in Figure 5. The total number of measurements made was 136. The residual redundancy in measurement geometries, however, means that all geometries are satisfactorily represented in the measurement database.

Dataset 2 was obtained in the Radio Science and Wireless Communications Laboratory at the University of Strathclyde, UK. The Gnome engine used was identical in type but a different instance to that used for dataset 1. The surface detail of the two engines was similar but not identical. Since

the surface detail represents an essentially random distribution of scatterers the use of two engines is not thought to materially reduce the usefulness of the resulting statistical model. These measurements were made in the presence of an engine cowling manufactured by SCITEK Consultants Ltd from stainless steel mesh to a specification provided by Rolls-Royce. Access to the measurement points was via slots cut in the cowling. When not being used the slots were covered by aluminium foil fixed in place using a conducting grease. We selected an Al foil with a typical thickness of 16 micron which is $16/1.66 = 9.64$ skin depths. The RF attenuation provided by the foil is then $20\log_{10}e^{9.64} = 84$ dB. The protection against leakage (out of one slot and back in another slot) is therefore 168 dB. The RF leakage is therefore considered negligible.

Channel Modelling. An empirical transmission loss model has been derived from the measurement datasets. In order to make the model generic, such that it can be applied to engines of arbitrary size, the model is parameterised in terms of path length, s , and path curvature, κ . The path length of

TABLE 2: Transmission model coefficients.

	Coefficients			
	A	B	C	D
Dataset 1	-0.99986	0.00043	-0.01678	-0.30833
Dataset 2	-0.99983	-0.010042	-0.01532	0.02659

each measurement is that of a helical segment connecting transmitter location (P_1) and receiver location (P_2). The arc length s is given by

$$s = \sqrt{(r\phi)^2 + l^2}. \quad (3)$$

The path curvature, reciprocal of radius of curvature, is given by

$$\kappa = \frac{r}{r^2 + (l/\phi)^2}. \quad (4)$$

The mean transmission gain in dB (<0), $\overline{G_T}$, has been modelled as a function of s and κ using

$$As + B\kappa + C\overline{G_T} + D = 0. \quad (5)$$

The resulting best-fit surface for datasets 1 and 2 is shown in Figure 6. Each transmission loss measurement in Figure 6 is the average of 6401, equally spaced, spot frequencies within the 100 MHz ISM band. Table 2 contains the best-fit coefficients for each dataset.

Since the scatter of points about the model is large (due to the large random variation of the engine surface from that of a smooth cylinder), an error model for the quantity

$$\Delta = G_{T,\text{measured}} - \overline{G_T} \text{ dB} \quad (6)$$

has also been derived by quantising the 2-dimensional space spanned by s and κ into a 4 (s) by 3 (κ) grid and calculating the mean and standard deviation of the resulting histograms of Δ within each 2-dimensional quantisation interval. The dependence of μ on s and κ and of σ on s and κ is then found using the same approach as that used for $\overline{G_T}$. The final value of transmission gain thus becomes a sum of a deterministic and a random component, that is,

$$G_T(s, \kappa) = \overline{G_T}(s, \kappa) + \Delta[\mu(s, \kappa), \sigma(s, \kappa)]. \quad (7)$$

The measurements and modelling described above were specific to the ISM frequency band. Further measurements have been made to extend the frequency range of the model up to 11 GHz and a source of thermal noise (determined by the receiving sensor node noise bandwidth, noise temperature and antenna temperature) has been incorporated. Should a bit-by-bit simulation be necessary a time-series model of interference drawn from the standard EUROCAE ED-14E [14] has also been made available. (The simulations presented here are packet level only and replace interference with an equal amount of white Gaussian noise.) Figure 7 is a block diagram of the complete channel model. This channel model has been implemented using Simulink.

5. MAC Protocols for Operation

In the proposed LCN topology, the data transmission involves two main communication schemes: the single-hop SN-to-CN communication and the multihop CN-to-CN communication. Since the SN-to-SN is a single-hop communication, its performance is dominated by the medium access control scheme. Although the CSMA/CA protocol has been widely accepted in wireless communication, considering the requirement of WIDAGATE that demands high data throughput, a predesigned, demand-based, and scheduled bandwidth allocation scheme with higher spectral efficiency is more favourable. In this section, a polling protocol is proposed and its performance is compared with the CSMA/CA. The simulation results of both CSMA and polling MAC protocols for SN-to-CH communication are presented, being of greatest concern for the operation of the network. For the CH-to-CH communication, due to the simplified topology of the proposed LCN, the routing protocol among CHs turns into a linear routing scheme and the CH-to-CH multihop communication is dominated by the MAC protocols as well. Hence, the results and conclusions achieved for SN-to-CH can be extended to the CH-to-CH with some minor modification.

5.1. Configuration and Performance Metrics. For a fair comparison of different MAC protocols, the network configuration (packet length, etc.) and performance metrics are described as follows. According to the engine testing requirement, all sensor nodes generate 102 Byte (40 Bytes payload for 8-channel measurement data plus overhead like channel ID, time stamp, packet header, etc.) data packets periodically at a sampling interval of 0.03 s. Both the acknowledgement packet (ACK) and the polling packet (REQ) have the same length fixed at 38 Bytes (14 Bytes for MAC layer and 24 Bytes for long preamble PHY layer). For the purpose of real-time data transmission in engine testing, the throughput is calculated on the basis of how many DATA packets are received during the course of an engine test run. Note that, since all the data are buffered at the SNs, some data may be transmitted to the data sink after the end of engine testing. Let Thr_{PKT} denote the number of received data packets while engine is running, an effective data bandwidth (throughput), Thr_{EDB} , in bits per second (bps) is defined as

$$\text{Thr}_{\text{EDB}} = \frac{8rL}{T} \text{Thr}_{\text{PKT}} \text{ (bps)}, \quad (8)$$

where L is the packet length (in Byte), r is the payload-overhead ratio of DATA packet, and T is the duration of engine testing. The latency, termed as sampling-to-receiving delay (SRD), is measured as the time from the SN sampling the physical signal to the data packet being received by the CH. The SRD includes queuing delays at the MAC layer and usually is longer than the access delay. If retransmission occurs due to transmission failure, a large SRD may appear which might be greater than the sampling interval. If failed packets are not discarded, then the SRD will accumulate.

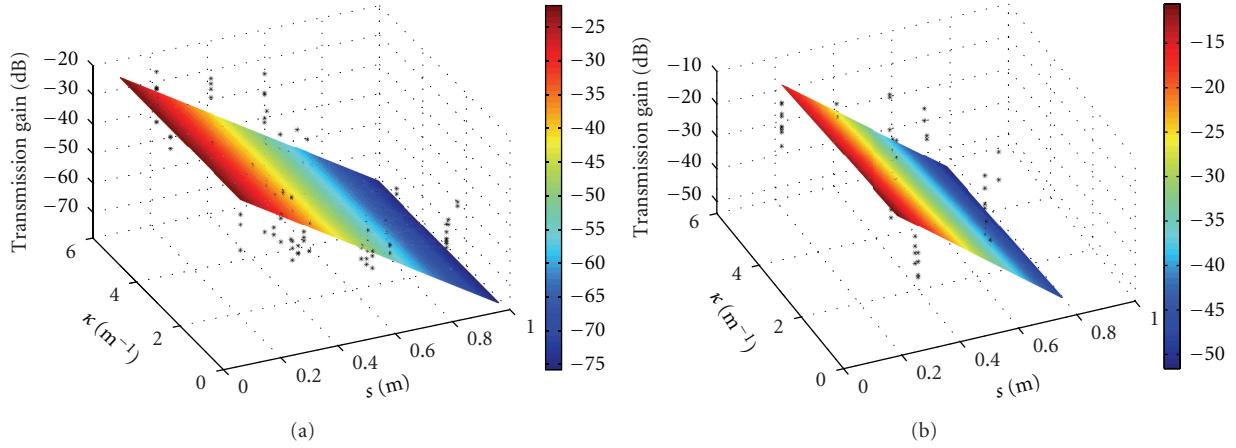


FIGURE 6: Best-fit plane surfaces to measurements (a) without cowling and (b) with cowling.

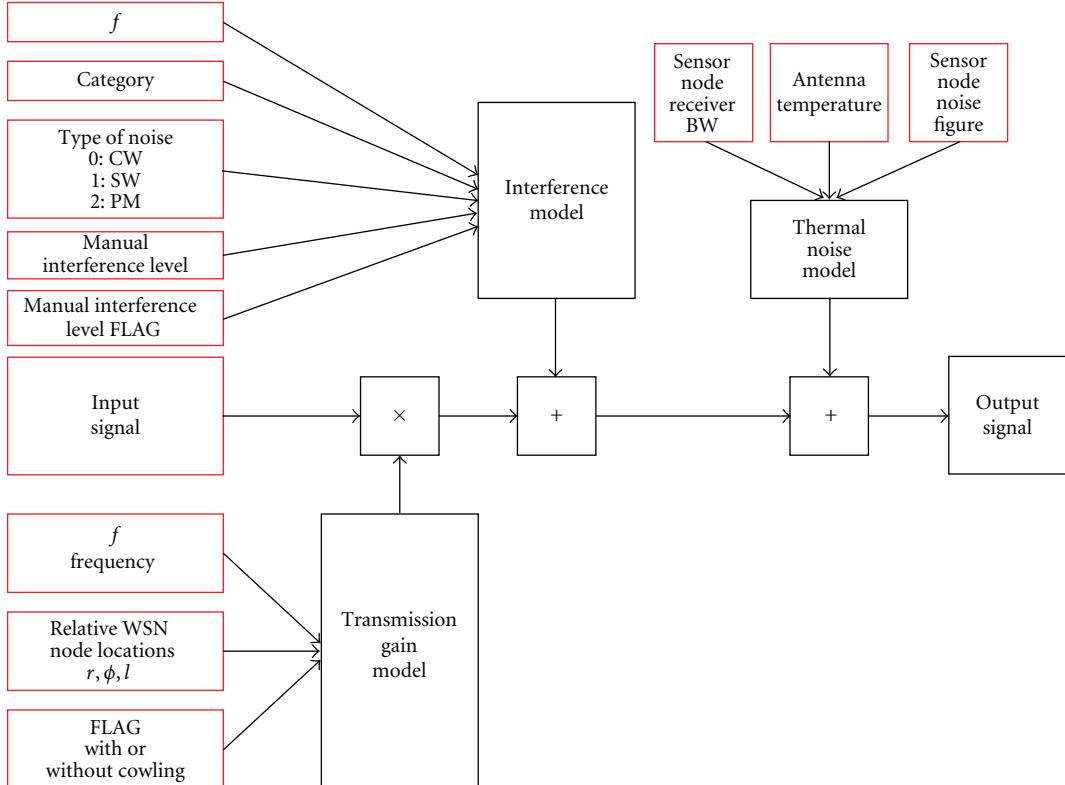


FIGURE 7: Channel model.

5.2. MAC Protocol Description

5.2.1. CSMA/CA Protocol. CSMA/CA is a decentralised random access mechanism in which nodes decide autonomously when a packet transmission starts. A node wishing to transmit must first sense the radio channel to determine if another node is transmitting. If the medium is not busy, the transmission may proceed. The CSMA/CA protocol avoids collisions by utilising a random back-off time if the node's physical or logical sensing mechanism indicates a busy medium. The

data delivery in CSMA/CA is based on an asynchronous, best-effort, connectionless delivery of MAC layer data with no guarantee that the packet will be delivered successfully. More details of the CSMA/CA can be found at [12].

5.2.2. Polling Protocol. The polling MAC protocol is a centralised access mechanism. Although it works by a “listen before talk scheme,” the SNs in a polling protocol listen to the request packet from the CH rather than by carrier sensing.

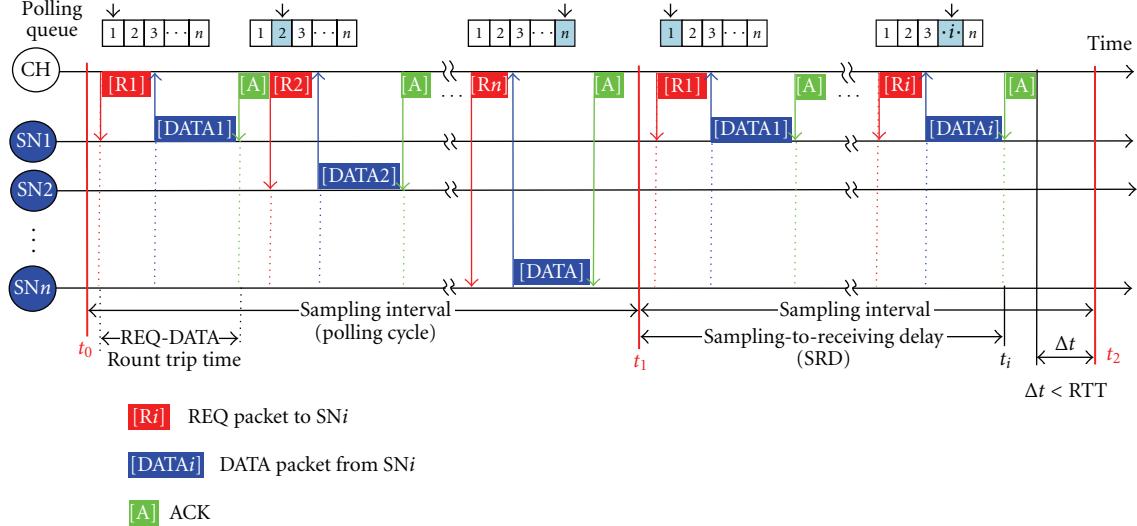


FIGURE 8: Illustration of the polling MAC protocol.

As shown in Figure 8, the operation of the proposed polling MAC protocol can be described in terms of cycles. Each cycle (which is the same length as the sampling interval) starts at the beginning of a sampling period by polling the child sensor nodes one by one in the cluster. At the SN side, once a measurement is sampled, a DATA packet is generated and queued at the MAC layer to wait for the polling packet from the CH. At the CH side, the CH's MAC layer maintains a polling queue storing the SN's IDs (i.e., MAC addresses). When a new polling cycle starts, the CH reads the first element of the polling queue and broadcasts a data request packet REQ. Only the SN matching the ID accesses the medium by replying with its DATA, while all other SNs keep silence. Once sending out a REQ, the CH sets a timeout. If no DATA packet is received within this timeout, the CH will poll the next SN. The selection of the timeout value depends on the REQ-DATA round trip time (RTT). The RTT in our scenario is 752 μ s for a 102 Bytes data packet at 2 Mbps data rate. If the CH receives the required data packet within the timeout, it replies with an ACK, followed by retrieving the next ID from the polling queue and sending a new REQ to poll the next SN. This process is repeated until either the end of the polling queue is reached or a new sampling period starts. For instance, Figure 8 shows two polling cycles for a cluster with n SNs. As the polling queue is an increasing sequence from 1 to n , SN1 is always polled first followed by SN2, 3 and so on. The polling in the first cycle ends when all the n sensor nodes are polled and none of them have any data to send. The second polling cycle ends earlier at i th polling ($i < n$), because the time Δt left for polling a node before next sampling is less than a REQ-DATA RTT. The reason for terminating the polling cycle earlier is to reduce the SRD and avoid the delay accumulation, because new sampled data will be ready for transmission when the new sampling period starts.

Note that the polling sequence determines when an SN will be polled, thus the medium access is coordinated by the CH and the collision is avoided. The polling queue can be configured as a random sequence or, by default, an increasing

sequence as shown in Figure 8. Hence, the bandwidth allocation is fully controlled by the CH and can be adaptive according to the data's priority and SRD requirement (e.g., for safety critical data) or be random indicating that all sensor nodes have the same priority. It is also worth noting that the polling protocol does not require time synchronisation to avoid collision. In order to guarantee every data packet is received by the CH, the proposed polling assumes a no packet drop policy, which means a DATA packet will be kept at the SN unless an ACK is received. This is implemented by a DATA queue at the SN's MAC layer.

5.3. Performance Comparison. Turning our attention to the simulation results of the IEEE 802.11 CSMA/CA and the proposed polling MAC protocol, the network throughput, packet loss, and sampling-to-receiving delay (SRD) are simulated and depicted in Figures 9 and 10.

Throughput. Figure 9 shows the relationship between the throughput and the cluster size, where the cluster size is defined as the number of child SNs in the cluster. Values of throughput are shown in terms of both how many packets are received by the cluster head in 30 seconds and the effective throughput (kbps). Two groups of throughput performances are shown in Figure 9. One is the throughput in an interference-free environment, while the other is subject to external interferences (EIs). The throughput achieved in the interference-free environment can be regarded as the upper bound of the throughput, since the throughput purely depends on the protocol itself.

In the interference-free case, as the cluster size increases from 5 to 26, the traffic load increases from 53 kbps to 275 kbps, and the throughput of CSMA/CA goes up steady and linearly with respect to the increasing traffic load. However, when the cluster size is greater than 26, throughput goes down steadily after reaching the maximum throughput of

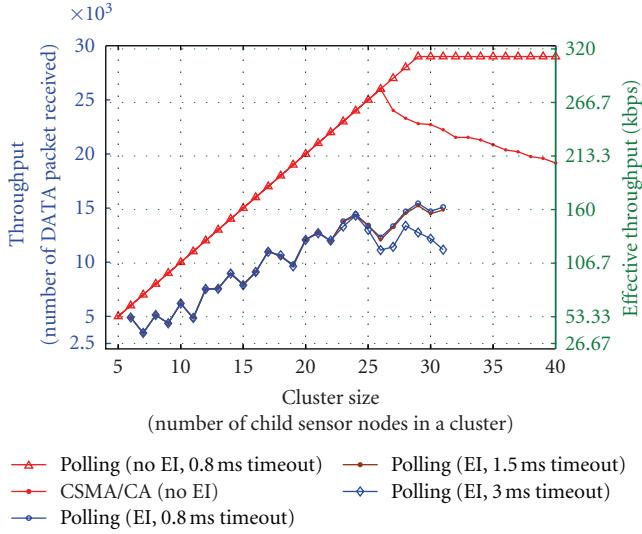


FIGURE 9: Network throughputs. Throughput of CSMA/CA and polling.

275 kbps. This shows that the CSMA/CA reaches the saturated condition at 275 kbps, representing the maximum network throughput in overload conditions. In contrast, the throughput of the polling protocol keeps increasing linearly until the cluster size is 29 and then becomes fixed at the highest throughput of 310 kbps even when the cluster size increases further. The simulation results also reveal that the polling protocol works better than the CSMA/CA at high traffic loads (i.e., over saturation). This is because, when the network is saturated, more collisions occur in CSMA/CA and more bandwidth is wasted, thus the throughput of the CSMA/CA decreases. Since the polling protocol is a collision-free scheme, no bandwidth is wasted on collisions and a higher throughput is achieved resulting in higher spectrum efficiency. Furthermore, as the polling has a flat throughput at the saturated condition, the polling protocol is more robust than CSMA/CA.

It is worth noting that, although no collision occurs in the polling protocol, the bandwidth has to be split between the transmission of DATA packet and REQ/ACK packet. The polling protocol can be further improved by either increasing the length of data packets (to increase the share of data transmission) or using a multipolling/multi-ACK scheme (to reduce the share of REQ-ACK).

When the external interference is presented, the packet loss is not only due to collisions but also due to the high level of interference. As a result, the throughput becomes lower than the upper bound. It can be seen that the throughput, in the presence of interference, is about half of the upper bound, suggesting packet loss is about 50% when interference is presented. As the REQ-DATA timeout impacts on the network performance, which is particularly true for short DATA packets at saturated condition, Figure 9 shows the throughput of the polling protocol at three timeout values of {0.8, 1.5, 3} ms. It can be seen that they are the same at low traffic load whereas a shorter timeout gives a bit higher

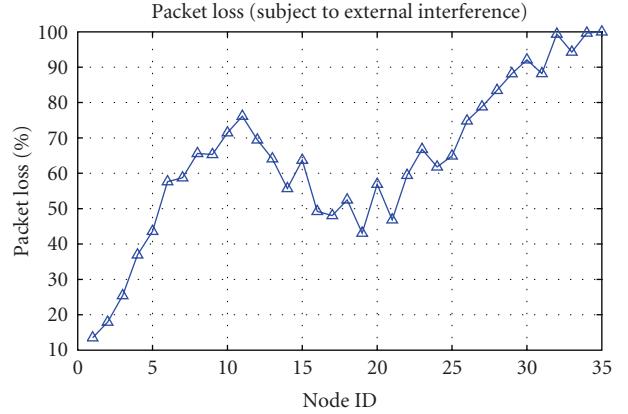


FIGURE 10: Packet loss of the polling protocol subject to the environment interference.

throughput at high traffic load. This is because, when packet loss occurs, a shorter timeout allows more slots for REQ-DATA exchanges in a polling cycle and more polling can be applied to compensate for the packet loss.

Packet Loss. In order to further study the impacts of the SN's location and the polling protocol on the throughput in a harsh environment subject to external interference, the packet loss rates of individual SNs in the polling protocol are depicted in Figure 10. In this simulation, 34 SNs in total are deployed along a belt circling the engine's surface in the clockwise direction. The CH is at the top of the engine, the first node SN(1) is closest to the CH, SN(1)–SN(9) are deployed from top to bottom at the left side of the engine, the SN(10) at the bottom is opposite to the CH (non-line-of-sight), and SN(11)–SN(34) are placed at the right side of the engine from bottom to top towards the CH.

It can be seen that the packet loss rates are affected by nodes' placement at the engine surface and the polling sequence. The first part of the packet loss curve (Node ID = 1, ..., 10) is dominated by the SNs position, where all these nodes can be polled in each polling cycle and the node's position at the engine surface is the main reason for the increase of packet loss. Since SN(1) is the nearest one to the cluster head, it has the lowest path loss and thus the lowest packet loss rate. Since SN(10) at the bottom is not line-of-sight to the CH, it has the largest pass loss and its packet loss reaches a peak of 75%. While the node ID increases, the node gets closer to the CH and the packet loss decreases accordingly due to the decreasing path loss. However, when the node ID becomes greater than 15, the polling sequence dominates the packet loss. These nodes are at the tail of the polling sequence having less chance to be polled and the number of data transmitted to the CH decreases. Therefore, the polling scheme at the tail works as a nonuniform polling. The average data loss rate for the first ten nodes ($SN(i)$, $i = 1, \dots, 10$) is 45.60%. As the first ten nodes are always polled, the packet loss in the first ten nodes is mainly due to the external interference.

Sampling-to-Receiving Delay (SRD). The SRDs of every node and their standard deviation are shown in Figure 11. The

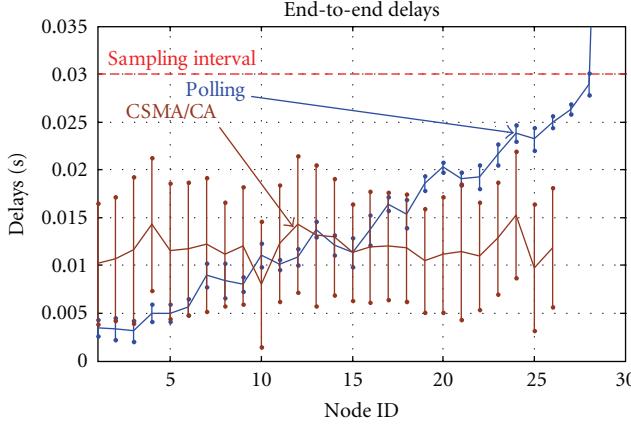


FIGURE 11: Sampling-to-receiving packet delays in the CSMA/CA and the polling MAC protocols.

blue line is for the polling protocol when the cluster size is 30 and the brown line is for the CSMA/CA of 26 sensor nodes. Due to its random access behaviour, the SRD of CSMA/CA is a random process with a mean of 0.012 ms and an average standard deviation of approximately 0.005. It is noted that, due to the large variances, the maximum SRD of CSMA/CA reaches nearly the sampling interval of 0.03 s. On the other hand, as the polling queue in the polling protocol is a fixed increasing sequence, the smaller ID number the SN has, the earlier the SN is polled. Thus the SRD increases linearly with respect to the SN's ID. Since the SN is polled nearly at the same time slot in every polling cycle, the variance of the SDRs is small with an average standard deviation of 0.0012.

From the simulation results, some conclusions can be drawn. (a) The throughput performance of both CSMA/CA and polling is similar at low and moderate traffic load when the cluster size is smaller than 25. (b) When the traffic load increases further, the network goes into saturation and polling is superior to CSMA/CA, where the throughput of CSMA/CA degrades significantly, but that of polling increases further and supports up to 29 sensor nodes for time-bounded data transmission. (c) In terms of latency, the polling shows much smaller jitter resulting in a better phase relationship among data in engine testing. This is a favourable feature for WIDAGATE. Overall, since WIDAGATE has high traffic load which makes the network saturated, the polling is more appropriate for WIDAGATE in terms of both throughput and latency.

6. Agent-Based Control and Optimisation

In contrast to the usual approach of running network simulations as a batch process, the Agent-based application layer provides not only flexibility of logging data for offline analysis and visualisation of the data/metric streams while the simulator core is running, but also provides an interface for the user to interact with the WSN simulator core. The user can make online queries and change node parameters (e.g., location, sampling rate, and traffic load) on the fly. The Agent-based approach also enables an intelligent online

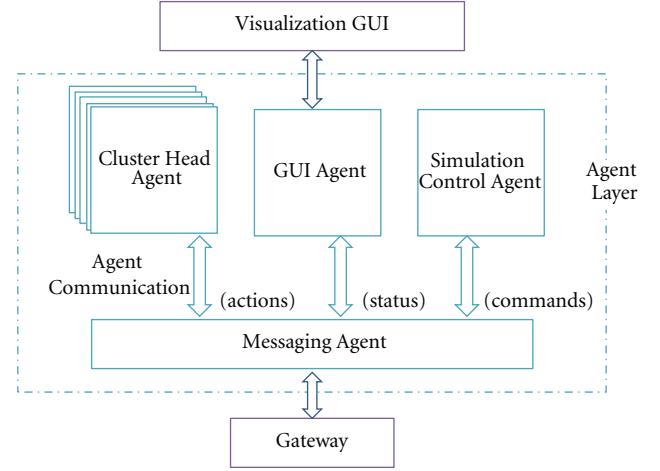


FIGURE 12: Architectural design of Agent Layer.

optimisation to improve the network performance. More specifically, the multiagent system provides the following functionalities to the WIDAGATE.

- Performance Measurement—the Agent Layer serves as an intermediate bridge to enable the user to obtain real-time network performance metrics from the WSN simulator core, for individual cluster head nodes, as well as for the entire simulation.
- Visualisation—the Agent Layer provides a Graphical User Interface (GUI) to enable the user to interact with the WSN simulator and visualise the node deployment, as well as network performance metrics and status updates.
- Interactivity—the Agent Layer enables the user to change parameters in the WSN simulator, such as changing node locations, so as to evaluate the effects on network performance.
- Optimisation—the Agent layer provides an intelligent optimisation algorithm to optimise the location of sensor nodes.

The agent layer architecture and functional design diagram are shown in Figure 12 and the functionalities of each agent module are listed in Table 3. The gateway is at the simulator core providing an interface to the Agent Layer. A TCP socket connection is used for this interface, as this allows the Agent Layer to be abstracted from the WSN simulator. In this way, the Agent Layer module can be easily ported to interface with other underlying platforms, such as a test-bed implementation.

The proposed multiagent system is implemented using JADE (Java Agent Development Environment) [15], as it can be easily ported from development and simulation to a real-world implementation. JADE also provides a set of FIPA compliant (the Foundation for Intelligent Physical Agents (FIPA, <http://www.fipa.org/>) is an IEEE Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other

TABLE 3: Agent module functionalities.

Agent	Functionality
Simulation control agent	(i) Controls the WSN simulator (start/stop/pause/resume) and creates all other agents (ii) Initialises the simulation parameters from input test data and configuration files
Cluster head agent	(i) Represents each cluster head in the WSN simulator (ii) Stores cluster head's state information and performance metrics (iii) Implements decision-making and network optimisation
GUI agent	(i) Implements 3D visualisation and graph plotting capabilities and interactive GUI for users to control simulation parameters
Messaging agent	(i) Implements the agent interface for WSN-Agent Layer integration (ii) Converts ACL (Agent Communication Language) messages in Agent Layer to WSN simulator's message format (e.g., commands, queries) and vice versa (e.g., status updates) (iii) Maintains the socket connection with the WSN Gateway

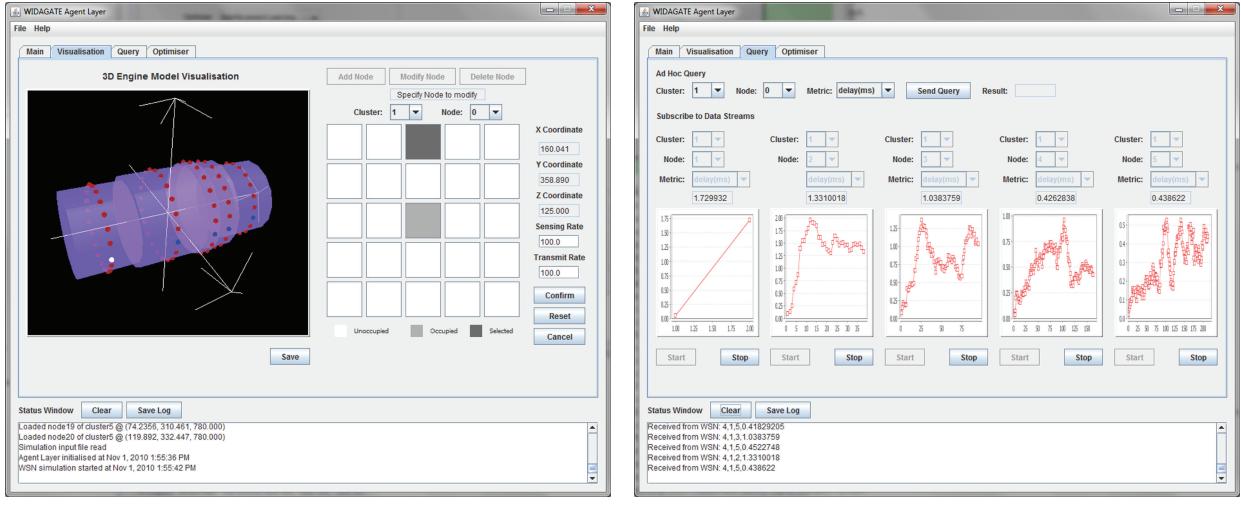


FIGURE 13: GUI screenshots of the Agent Layer.

technologies) agent messaging protocols for negotiation and decision-making, also known as ACL (Agent Communication Language). The main functions provided by the Agent-based application layer are detailed below.

6.1. Visualisation. The developed WSN core simulator is visualised in an online fashion by the multiagent system. The “Visualisation” tab of the GUI (shown in Figure 13(a)) illustrates node locations on a 3D engine model and allows users to adjust the node locations. The “Query” tab (shown in Figure 13(b)) provides capabilities for inputting user requests to monitor specific nodes and online visualisation of the vibration/pressure/temperature data collected by the sensor nodes. The user can make an ad hoc data query or subscribe to a node’s performance metric stream, which will be logged for offline data analysis.

6.2. User Interaction. The user interaction is implemented in two stages. The interaction between the user and the agent system is done by the GUI agent and the interaction between the agent system and the underlying simulator core is done by the Messaging Agent, which connects to the simulator

core through a socket connection for information exchange. The simulator core has a special node, termed as gateway, working as a server and providing a socket port. The gateway is an event scheduler of the simulator core and has full access to all other nodes (i.e., SNs, CHs). Once the simulator core is initialised, the gateway sets up a socket port and listens to the connection request from the agent system. A set of commands are defined for exchanging information between the agent and the gateway (as shown in Table 4).

At the agent side, the Messaging Agent interfaces with the gateway module via a TCP connection (as shown in Figure 14). When an agent module wants to interact with the simulator core, an ACL message is generated and sent to the Messaging Agent. The Messaging Agent interprets the ACL message into the appropriate commands and sends them over the socket connection to the simulator core. These commands are parsed at the gateway and executed by the corresponding modules in the simulator core. In the reverse manner, update messages from the simulator core are collected at the Gateway module and subsequently passed over the socket connection back to the Messaging Agent, which relays these updates to the corresponding Agent modules.

TABLE 4: Message type definition.

Value	Msg_Type	Attribute list	Description
1	SUB_START	—	Subscribe for gathering performance data from specified node
2	SUB_STOP	—	Stop subscribing to specified node
3	QUERY_ONCE	—	Single query of performance measures from specified node
4	DATA	{performance_measure_list}	List of performance measures from specified node
5	MOVE_NODE	x_coord, y_coord, z_coord	Move specified node to specified location
6	ADD_NODE	cluster_id, x_coord, y_coord, z_coord	Add a new node at specified location
7	DEL_NODE	—	Delete the specified node
8	ACK_MOVE_NODE	x_coord, y_coord, z_coord	Acknowledge the command to move specified node

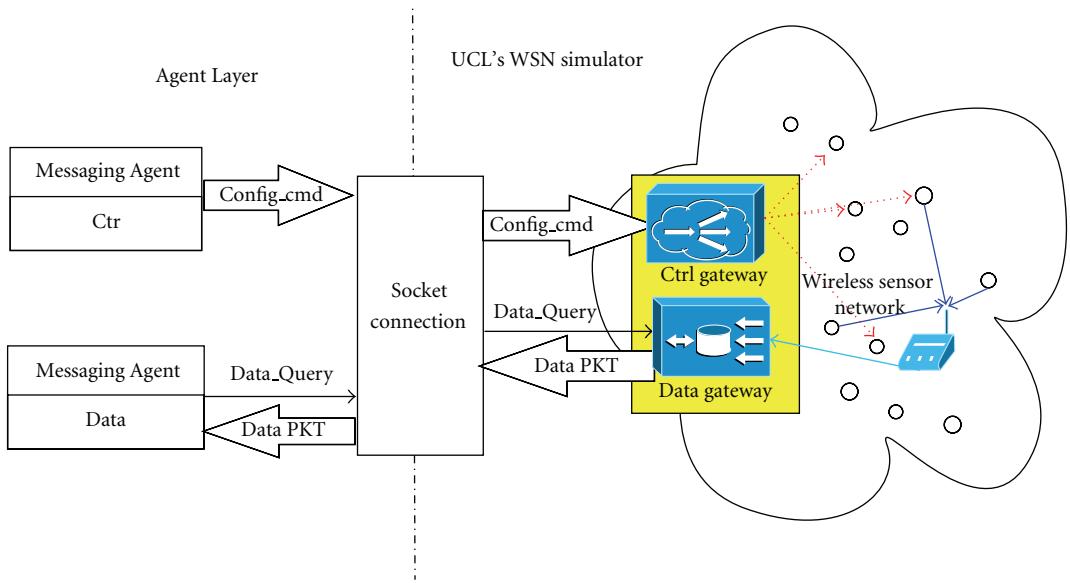


FIGURE 14: Communication mechanism between Agent Layer and WSN simulator.

6.3. Multiagent Optimisation. The ability of Agents to communicate information and make intelligent decisions about meeting objectives has been exploited in this application. In particular, we have incorporated the SN agents with an optimisation capability using which they are able to, as a group of information-sharing cooperative group, determine the most effective topology of the wireless network (with respect to a given performance metric, such as propagation delay).

6.3.1. Optimisation Formulation. The SN should be placed within a given distance constraint around the transducer. Given this requirement, the possible locations around a transducer can be formulated as a $m \times n$ grid, as shown in Figure 15, where a 5×5 grid represents the 25 possible locations for the SN. Choosing one of these grid locations as the new position of an SN is considered an “action” by the corresponding SN agent. Given the node locations (i.e., the 5×5 grid) is a discrete set of choices, the action space is also discrete. The metric of interest (e.g., packet delivery delay) is nonconvex with respect to the action. As such a discrete nonconvex search method is called upon to determine the optimal locations for the placement of sensor nodes. One such search method is the Reinforcement Learning (RL)

approach [16]. Note, the generic RL method searches for optimal “paths” where an agent tries to find the minimum cost (or, maximum utility) traversal from a start to an end location using a metric that is related to the “quality” of each intermediate action or location. However, in our case, we are interested in finding just the optimal location for an SN (with respect to the transducer location) instead of a traversal path. This simplifies our problem formulation.

The search for the optimal location of an SN can be formulated as an iterative RL problem, in which a node keeps track of the *values* of network performance (e.g., throughput, SRD) for each possible action $a \in \{0, 1, 2, \dots, 24\}$, where each entry corresponds to a square of the 25 grids and the values stored are represented by $Q(a)^2$ (the notation Q is adopted from the standard representation used in RL literature [16]). The parameter to be optimised is the action code a or the SN location. Note this is a simplified RL formulation in that it does not use information about state transition (i.e., the traversal path of the SN).

An action is drawn from the set of all actions using the ϵ -greedy method [16], such that with probability ϵ , a node will randomly choose an action from all possible actions, and with probability $(1 - \epsilon)$, the node will choose the action with

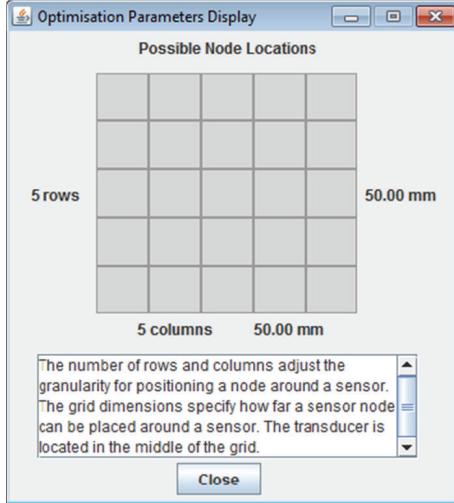


FIGURE 15: Grid squares where the SN can be placed around a transducer.

the largest Q value (i.e., the *greedy* choice). As the search for the optimal location progresses, the value of ϵ is gradually reduced to increase the probability of choosing the *greedy* action, that is, decreasing exploration and increasing exploitation.

6.3.2. Optimisation Implementation. The ϵ -greedy optimisation can be conducted independently by each SN agent (also called SN in the following discussion) to determine their optimal locations within their respective location grids. Another approach is to allow SN agents to share information with their neighbouring agents. When an SN chooses an action (i.e., it selects one of the grid locations), it queries its neighbours to rate the action and an action is chosen according to their past experiences (in this case a neighbour is another SN that is nearby to a given SN which is reachable wirelessly). SNs communicate with their neighbours wirelessly to perform query and receive ratings. A neighbour rates the action based on its effect on the local metric of interest. There could be instances in which an action that is deemed good for a node may be seen as bad by a neighbour node.

In order to reduce the number of iterations, the linear cluster topology is taken into account, where a CH agent performs optimisation for all nodes in its cluster. CH agents interact with other CH agents by exchanging action values. These interactions are shown in Figure 16. This method requires the CH agent to be aware of the states of all sensor nodes within its cluster, which can be supported by the polling protocol.

This design is modular in that the same local optimiser code can be ported to new cluster head nodes easily, which would subsequently interact with the nearest cluster head nodes to carry out the optimisation. The user initiates the optimisation from the user interface (as shown in Figure 17).

At the initial stage, the GUI Agent distributes the optimisation parameters (m, n, a , the objective function and stopping criterion) to each CH Agent and the CH Agent then

initialises the action-value and visit-count for each of the $m \times n$ grid squares. The CH Agent then triggers the optimisation process by sending a metric query via the Messaging Agent to the simulator core. While the simulator core is running, the CH Agent collects the metric of interest from the messages returned by the gateway. The average across a few collected data points is computed before a new action is chosen based on the ϵ -greedy algorithm depending on the current action values. The action values are updated accordingly by either the SARSA update rule [16]

$$Q_a \leftarrow Q_a + \alpha(r + \gamma Q_{a+1} - Q_a) \quad (9)$$

or the Q-Learning update rule [16]

$$Q_a \leftarrow Q_a + \alpha(r + \gamma \max_{a+1} Q_{a+1} - Q_a), \quad (10)$$

where α is the learning rate, r represents the immediate reward which, in this case, depends on the objective value (e.g., SRD), γ is the discount rate controlling the amount of consideration the optimiser gives to rewards of subsequent actions, and it determines how much the optimiser values future potential gains relative to immediate rewards.

The computation of new action lies in a range of integer values $\{0, 1, 2, 3, \dots, 24\}$. The integer representation of the chosen action is then converted into the corresponding row and column in the location grid. The row and column values are then converted into the corresponding 3D coordinates, using the transducer locations and the engine model. The new location is then sent to the simulator core via the Messaging Agent and takes effect immediately. The entire process is repeated until the set of locations converges, that is, it does not change across iterations. The solution is then reported to the GUI Agent, which displays the optimised node locations on the GUI. The user can then choose to try out the proposed node locations.

The RL-based optimization algorithm is tested on a simulation where the objective is to determine the location with the minimum value on a 5×5 grid. We choose propagation delay as the metric of interest in this test study (any other real-valued metric can be used instead). In this simulation, we have set the environment such that one square is set to return a mean value of 1 ms; one square is set to return a mean value of 5 ms; the rest of the squares return a mean value of 10 ms. The standard deviations for all locations are set to be 5 ms. The convergence of the search process is shown in Figure 18, where the x -axis shows number of iterations and the y -axis the delay (s).

In this simulation, it took about 260 iterations for the search to converge. This is because all alternative locations (the individual squares in the grid) had to be visited a few times to determine the mean value before confirming that a particular location returns the lowest value.

7. Validation through an Engine Testing

In the second stage of the development, validation of the simulation environment was carried out by testing the wireless sensor nodes performance on two engine platforms: the

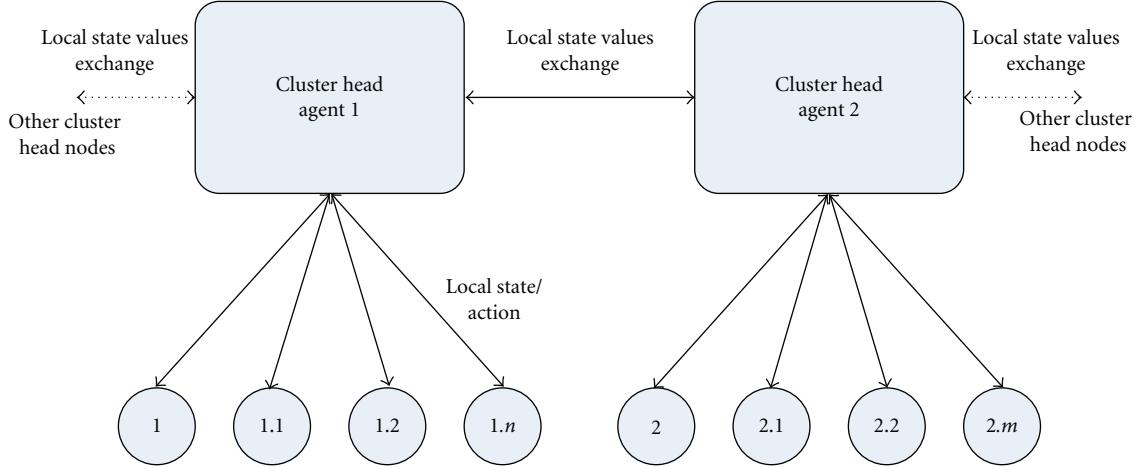


FIGURE 16: Value exchanges in distributed reinforcement learning-based formulation. Note: $x.0$ refers to the cluster head node in cluster x , whereas $x.1, \dots, x.n$ or $x.m$ refer to sensor nodes within cluster x .

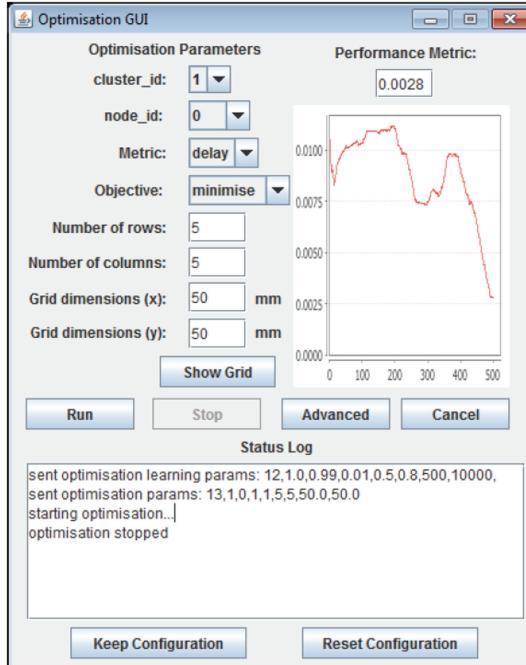


FIGURE 17: GUI to set parameters and launch optimisation.

Gnome and the Trent 900. The sensor nodes were the CISPs, provided by SELEX Galileo, UK with the tests carried out in the test facilities of Rolls-Royce, Derby, UK. The main tasks and objectives of the tests were (a) to test the radio model and interference when operating the CISPs nodes within an engine; (b) to test the network communication performance in terms of data throughput and packet loss; (c) to validate the simulation model, specifically the transmission gain model.

7.1. Gnome Engine Test Package Arrangement and Equipment. The Gnome test setup is shown in Figure 19, where the

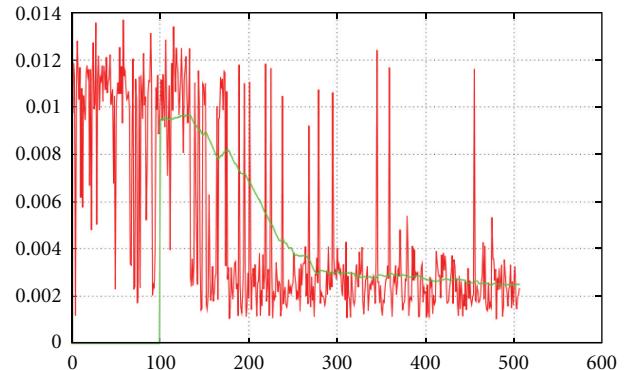


FIGURE 18: Convergence graph of the optimiser determining the optimal node location.

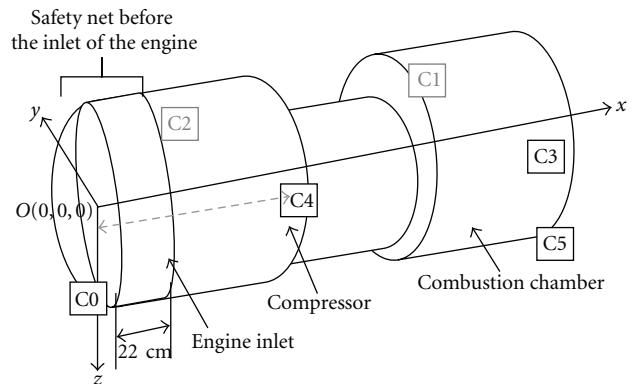


FIGURE 19: A schematic diagram showing the CISPs node locations.

engine was fixed on the ground with six CISPs nodes mounted onto the engine frame via brackets. Their locations are C2 and C4 (on either side of the compressor), C0 (at the engine inlet), C5 (at the back of the combustion chamber), and C1 (to the left of the combustion chamber) with the addition

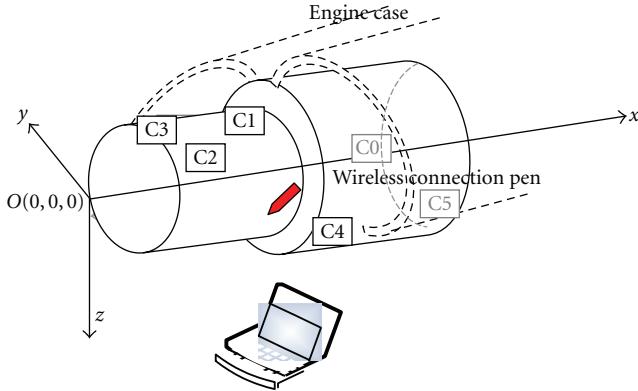


FIGURE 20: Trent 900 engine test setup and a schematic diagram of CISP node locations.

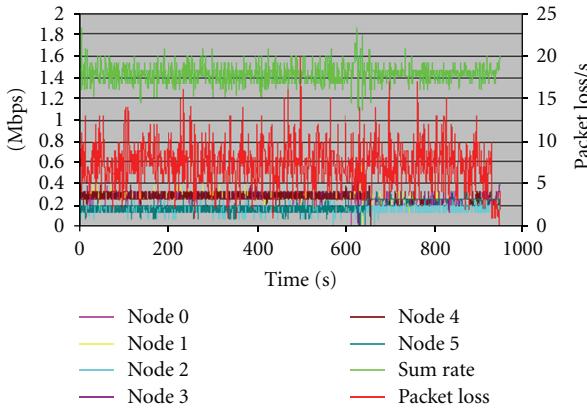


FIGURE 21: Six CISP nodes data rate and packet losses for static Gnome engine test.

of sensor node 3 which was to the right of the combustion chamber (C3). A schematic diagram of CISP node locations is shown in Figure 19. A single cluster topology was set up to test the CSMA/CA protocol to allow us to validate the simulation platform. As the CISP nodes are commercial hardware adhering to the 802.11 standard other MAC protocols could not be tested. Implemented on the CISP device all the nodes in the test were communicating back to a CH unit located in the control room of the engine test facilities. During the first set of tests, a single communication channel, Channel 6 (2.437 MHz), was used.

7.2. Trent 900 Engine Test Procedure. The second set of tests were carried out on the static Trent 900 engine in the Rolls-Royce training centre, Figure 20. The locations of the CISP nodes were C0 (Oil tank on compressor chamber), C1 (Underneath the compressor chamber), C2 (Compressor chamber exit), C3 (Turbine exit), C4 (Located at the gear box level), and C5 (Engine inlet). The CH (a laptop) was located near the external gearbox drive shaft, approximately in line with node C1. A single cluster topology was adopted and a wireless “sniffer” pen placed inside the engine in order to act as the cluster head as it would not have been practical to place the laptop itself in the case of the Trent 900.

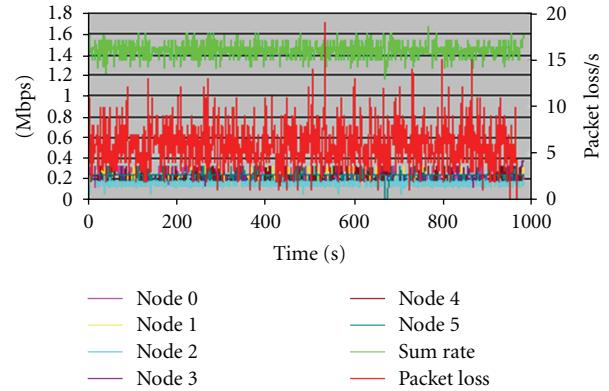


FIGURE 22: Six CISP nodes data rate and packet losses for running Gnome engine test.

7.3. Test Results. The study concentrated on benchmarking the performance of the WSN in terms of communication channels, communication protocols in the two case scenarios of (a) an unpowered and (b) running engine, given the sensor nodes were kept at the same location. The network performance, or quality of service (QoS), is measured by throughput and packet loss. In the next sections an example of the test readings for six CISP nodes on both engine types is given.

Gnome Engine Tests. The average data throughput for static and running Gnome engine are 1.435 Mbps and 1.440 Mbps, respectively, while the average packet losses are 7.245 and 5.893 (packet loss per second), respectively. The throughput and packet loss with respect to time are shown in Figure 21 for static engine and Figure 22 for a running engine. In the running test, it is possible to observe that the data throughput is more evenly distributed across the network, with nodes 0, 1, 3, 4, and 5 carrying a load of approximately 0.25 Mbps and node 2 carrying a slightly smaller data rate of approximately 0.17 Mbps.

Trent 900 Engine Tests. As shown in Figure 20, six CISP nodes were deployed around the engine core and fan case and covered by the cowling. The throughput and packet loss of the six CISP test are shown in Figure 23. It is possible to observe that again node 1 and node 2 present an analogous behaviour with an approximate data rate of ~ 0.25 Mbps. Node 0 presents a much degraded performance, with approximately 0.1 Mbps transmission and node 3 seems to follow the behaviour of node 0. Nodes 4 and 5 present the highest data rate transmission in the system.

This phenomenon is related to the MAC protocol employed in 802.11. The IEEE 802.11 employs a Carrier Sensed Multiple Access with Collision Avoidance (CSMA/CA) MAC protocol to manage the access of the media, in this case RF over the air. This protocol is an unmanaged protocol which seeks to avoid collisions (i.e., two radios transmitting at the same time) and resolves the situation by backing off for a random period. This protocol works well with channels that are operated with low link utilisation, lower

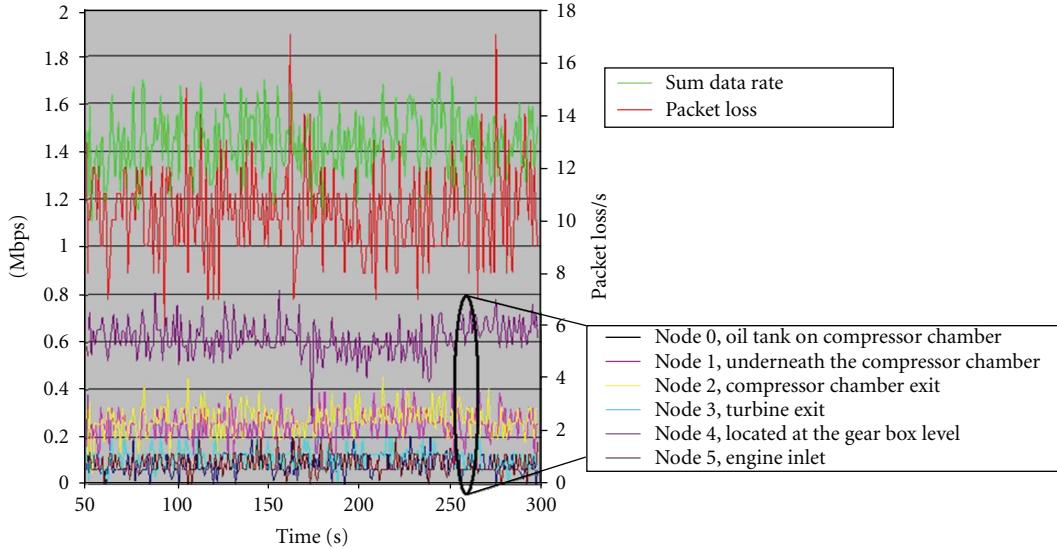


FIGURE 23: Six CISP nodes data rate and packet losses for Trent 900 engine test.

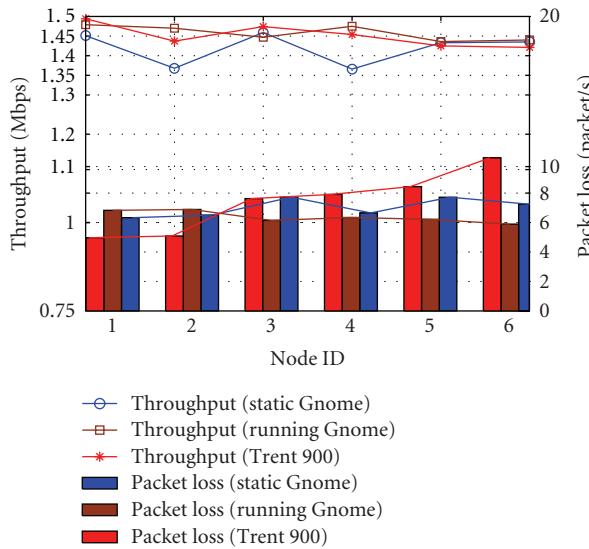


FIGURE 24: Summary of engine tests results. 6 CISP nodes throughput and packet losses.

than approximately 30%. However, in our engine tests, the wireless network works in a saturation condition where the traffic load injected into the communication network by the sampling process is about 70% of the nominal data rate. It may be that the random back off times in the MAC implementation are actually only pseudorandom, perhaps explaining the cyclic nature of some of the results.

Furthermore, the Trent 900 is a large engine with an irregular shape and surface resulting in the six CISP nodes having different transmission ranges and sensing ranges. It is possible that what we experience during the test is a “hidden node” problem. When the transmitters transmit to the same receiver at approximately the same time, they do not realise, as pointed out in [9], that their transmission collide at the

receiver. The hidden node problem in CSMA/CA networks causes unfairness. During the testing, some other RF devices (e.g., Bluetooth) and interference sources may exist and introduce interference to our system. As shown in the literature [17, 18], there is a trade-off between the total throughput and the fairness in CSMA/CA networks with multiple wireless links. The fairness may go worse in saturated networks.

A summary of the data recorded for both the Gnome and the Trent 900 Engine tests is shown in Figure 24 depicting the throughput and packet loss when the number of CISP nodes increasing from 1 to 6. It was possible to observe that as the number of nodes increased there was a slight increase in packet loss rate.

8. Conclusions

In this paper, we have described, in detail, the main achievements of the WIDGAGTE project: the development of an accurate and experimentally validated simulation model and a system demonstrator of a wireless sensor network for data gathering and health monitoring during gas turbine engine testing. The deployment of sensors within an aero engine constitutes a harsh environment with complex RF transmission characteristics and therefore a bespoke radio environment model has been developed from experimental data to inform the simulation models. Extensive simulations based on this model have shown that, in such a harsh environment, a polling-based contention-free medium access control scheme with regular duty cycles can outperform contention-based protocols in terms of both the throughput and delays.

An agent-based WSN simulation platform with a user-friendly GUI for customer interaction and optimisation has been developed to control the network simulations to allow an engine test engineer to rapidly develop an optimal network deployment to best suit their testing regime.

The developed software simulation platform and the hardware test bed not only demonstrates the usefulness of wireless technologies, but also helps the end-users build the confidence on the use of wireless technology for engine condition monitoring. The system proposed has a high innovative value, derisking wireless data acquisition in engine testing and potentially allowing in-flight condition monitoring of gas turbine engines, with extension to a wide range of potential aircraft monitoring applications.

Acknowledgments

This research was supported by the WIDAGATE (Wireless Data Acquisition in Gas Turbine Engine Testing) project sponsored by the Technology Strategy Board (TSB) project Gathering Data in Complex Environments and the UK Engineering and Physical Sciences Research Council (EPSRC) under Grants TS/G002614/1 and TS/G002681/1. The Technology Strategy Board is a business-led executive nondepartmental public body, established by the government. Its role is to promote and support research into, and development and exploitation of, technology and innovation for the benefit of UK business, in order to increase economic growth and improve the quality of life. It is sponsored by the Department for Business Innovation and Skills (BIS). For further information please visit <http://www.innovateuk.org/>. The authors wish to acknowledge the following collaborative partners and individuals for provision of the engine experimental data and technical support: Graham B. Hesketh, Armin Stranjak (Rolls-Royce plc), Douglas Friedrich, Jason Lepley, Ola Aribisala, David Lloyd, and Graham Bourdon (Selex Galileo, UK).

References

- [1] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: challenges, design principles, and technical approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, 2009.
- [2] H. A. Thompson, "Wireless sensor research at the Rolls-Royce Control and Systems University Technology Centre," in *the 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology (VITAE '09)*, pp. 571–576, May 2009.
- [3] K. Sasloglou, I. A. Glover, P. Dutta, R. Atkinson, I. Andonovic, and G. Whyte, "Empirical modelling and simulation of transmission loss between wireless sensor nodes in gas turbine engines," in *the 7th International Conference on Information, Communications and Signal Processing (ICICS '09)*, December 2009.
- [4] L. Krishnamurthy, R. Adler, P. Buonadonna, and J. Chhabra, "Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the North Sea," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2005.
- [5] H. A. Thompson, "Wireless and Internet communications technologies for monitoring and control," *Control Engineering Practice*, vol. 12, no. 6, pp. 781–791, 2004.
- [6] D. Goldsmith, E. Gaura, J. Brusey, J. Shuttleworth, R. Hazelden, and M. Langley, "Wireless sensor networks for aerospace applications—thermal monitoring for a gas turbine engine," in *the NSTI Nanotechnology Conference and Expo (NSTI-Nanotech '09)*, pp. 507–512, May 2009.
- [7] H. Bai, M. Atiquzzaman, and D. Lilja, "Wireless sensor network for aircraft health monitoring," in *Proceedings of the 1st International Conference on Broadband Networks (BroadNets '04)*, pp. 748–750, October 2004.
- [8] D. M. Benson and H. Bai, "Aircraft engine sensor network using wireless sensor communication modules," US Patent Application 20050213548, 2005.
- [9] R. K. Yedavalli and R. K. Belapurkar, "Application of wireless sensor networks to aircraft control and health management systems," *Journal of Control Theory and Applications*, vol. 9, no. 1, pp. 28–33, 2011.
- [10] K. A. Agha, G. Chalhoub, A. Guittot et al., "Cross-layering in an industrial wireless sensor network: case study of OCARI," *Journal of Networks*, vol. 4, no. 6, pp. 411–420, 2009.
- [11] H. Yang and B. Sikdar, "Performance analysis of polling based TDMA MAC protocols with sleep and wakeup cycles," in *IEEE International Conference on Communications (ICC '07)*, pp. 241–246, June 2007.
- [12] IEEE-TG15.4, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," IEEE standard for Information Technology, 2003.
- [13] Zigbee-Alliance, "ZigBee specification," 2005, <http://www.zigbee.org/>.
- [14] EUROCAE ED-14E, "A Joint EUROCAE RTCA Achievement," Tech. Rep. Sections 19–21, The European Organisation for Civil Aviation Equipment, March 2005.
- [15] G. Caire, *JADE Tutorial: JADE Programming for Beginners*, 2009.
- [16] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 1998.
- [17] Y. Jian, M. Zhang, and S. Chen, "Achieving MAC-layer fairness in CSMA/CA networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 5, pp. 1472–1484, 2011.
- [18] C. Huang, C. T. Lea, and A. K. S. Wong, "On fairness enhancement for CSMA/CA wireless networks," *IEEE Systems Journal*, vol. 4, no. 4, pp. 511–523, 2010.

Research Article

The Effects of Motion on Distributed Detection in Mobile Ad Hoc Sensor Networks

Xusheng Sun and Edward J. Coyle

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA

Correspondence should be addressed to Edward J. Coyle, ejc@gatech.edu

Received 22 July 2011; Revised 1 November 2011; Accepted 4 November 2011

Academic Editor: Frank Ehlers

Copyright © 2012 X. Sun and E. J. Coyle. 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 consider a set of mobile wireless sensors that collect observations about a brief, localized event. As they continue to move about, one of them processes its observations, decides that an event of interest occurred, and wants to determine if other sensors confirm its results. This sensor assumes the role of a Cluster Head (CH) and requests that all other sensors that collected observations at that time/location reply to it with their decisions. The motion of the sensors since the observation time determines how many wireless hops their decision must cross to reach the CH. We analyze the effect of this motion in the 1D case by modeling each sensor's motion as a Correlated Random Walk (CRW). We also account for measurement errors and communication or processing errors in each wireless hop. Quantities, such as the error probability of the final decision at the CH and the minimum energy required to collect the local decisions from all relevant sensors, can then be directly calculated as functions of time and the parameters of the CRW, the measurement noise and the channel noise. These results allow rapid characterization of the time-dependence of distributed detection algorithms that are executed in mobile sensor networks.

1. Introduction

The systems-level work reported in this paper is motivated by a security scenario that has arisen in the context of the eStadium project [1–3], which develops smartphone applications/games, wireless communication networks, and wireless sensor networks for football games and other large-scale events. One security scenario of great interest occurs in the hour immediately before or immediately after a large event. At these times, large numbers of fans, stadium personnel, and concession staff are typically walking toward or away from the stadium. This is a time when they are very vulnerable to deliberate or accidental exposure to hazardous chemical or biological agents.

Suppose that the smartphones that the fans are carrying, or possibly the tickets that they have bought, have sensors embedded in them that can detect these agents. Capabilities like this are the goal of the US Department of Homeland Security's "Cell-All" program [4]. These sensors would necessarily be small and inexpensive and might thus have high false-alarm rates or low rates of correct detection. It

is thus important that as many of their individual decisions as possible be fused with each other to ensure a correct final decision. They must therefore be able to communicate with each other or with a designated Cluster Head (CH) to accomplish this goal. These mobile, battery-powered, wireless sensors can thus be modeled as a mobile ad hoc network that is supporting an application that performs distributed detection.

Our first goal is to develop algorithms to efficiently calculate the performance of distributed detection algorithms in this dynamic and complex scenario. These numerical algorithms should account for as many factors as possible, including the motions that are typical of people in crowds, measurement errors in the sensors, errors made during wireless communication, and fusion algorithms with low enough complexity that they can be executed on very low-power processors or on processors that are shared by many different applications.

This paper is organized as follows. Section 2 describes the scenario that has motivated the work in this paper and that is analyzed as an example in Section 8. Prior work in this area is

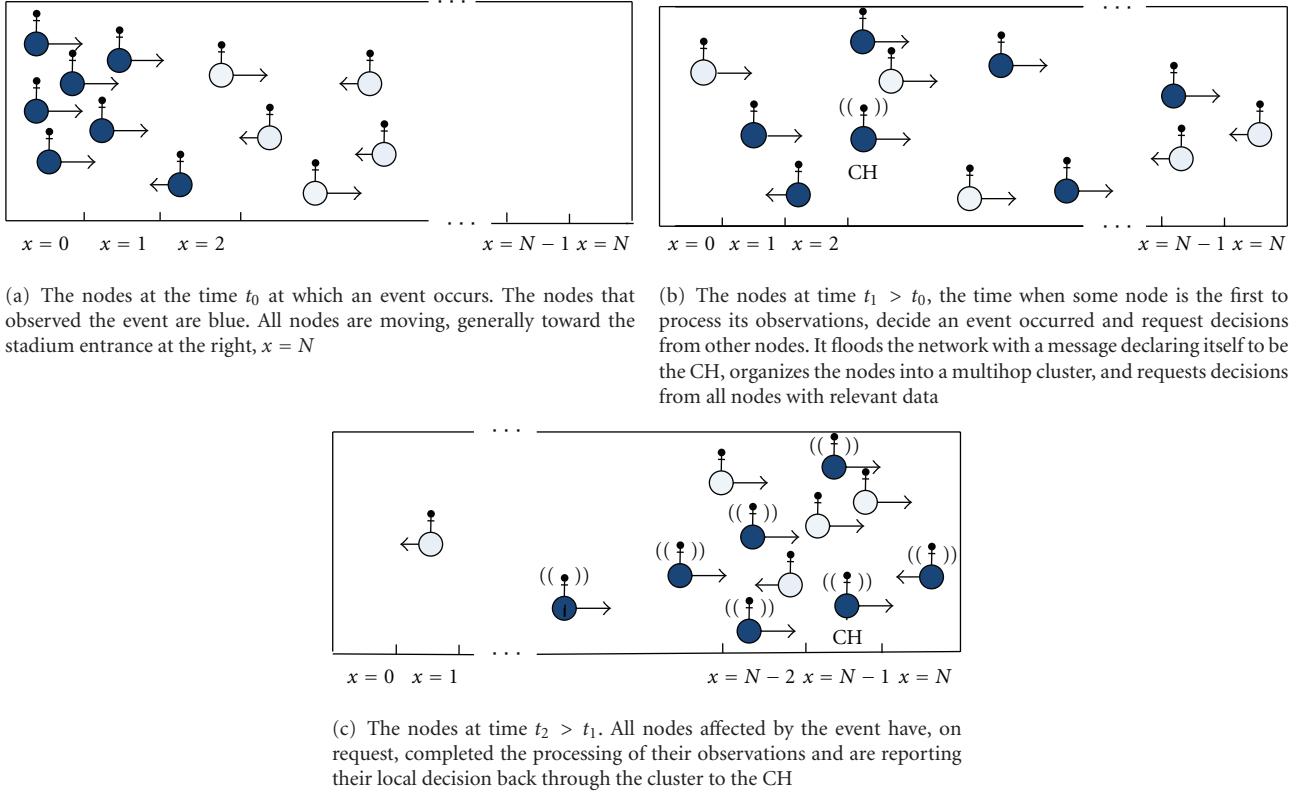


FIGURE 1: A security scenario that arises when large crowds are moving toward a stadium or other venue to participate in an event.

summarized in Section 3. In Section 4, the transient behavior of semi-infinite correlated random walks is reviewed. The solution to the finite case, which is a good model for the motion of people in the scenario of interest to us, is provided in Section 5. Its statistical properties that have the greatest impact on detection applications are analyzed in Section 6. In Section 7, the performance of detection tasks in mobile wireless sensor networks is determined. Numerical and simulation results are provided in Section 8 for the scenario described in Section 2. They demonstrate (i) the use of the mobility model in calculating the minimum energy required to collect data from each sensor participating in the data fusion process, and (ii) the decision error probability at the cluster head once the decisions from all participating sensors have been gathered. These two quantities are functions of both time and the parameters of the mobility model and we demonstrate how the results in this paper provide the ability to rapidly and accurately calculate or simulate them. Some conclusions and a discussion of future work are provided in Section 9.

2. Scenario: Distributed Detection in a Mobile Ad Hoc Network

To evaluate the tools developed in this paper, we assume a situation like that shown in Figure 1. The wireless nodes shown in the figure are carried by people or vehicles that are moving. They have been moving for some time, generally

toward the stadium entrance, which is at position $x = N$ at the right end of each subfigure. Their motion to the right is thus strongly dominant over motion to the left, which is shown by arrows to the right that are larger than arrows pointing to the left. Each node may occasionally switch its direction of motion and briefly move against the general flow, just as people do in a real crowd.

At time t_0 we assume without loss of generality that the sensors are in a configuration like that shown in Figure 1(a). We assume that the sensors that are colored blue have each collected observations/samples that have been affected by a chemical or biological agent. It takes time for the sensors to process these observations/samples and the processors managing the sensors may have to finish other tasks before processing them. During this processing delay, the people carrying the sensors continue to move toward the stadium.

At time $t_1 > t_0$ one of the sensors shown in Figure 1(b) has finished processing its observations. The results satisfy some criterion—a preliminary positive detection—that requires that sensor to seek other sensor's results so they can be fused with its own to make a highly reliable decision. This sensor declares itself to be a cluster head (CH) and floods the network of sensors—shown in Figure 1(b) as one sensor transmitting at time t_1 —with a request for other nodes to send it their results if they were at approximately the same location and time that the CH acquired its samples. This first request for data will suppress or supersede any other requests by any other sensors for data generated for this location and

time. Thus, all other sensors with data about this event will forward it to the CH. For our analysis, we will consider two situations: (i) the CH remains stationary, and (ii) the CH continues to move with the crowd.

At time $t_2 > t_1$, the CH's request for reports has reached all other sensors, including those with nothing to report, as shown in Figure 1(c). Sensors with data to report, shown in blue, begin transmitting their results to the CH. In some cases, the CH may be multiple wireless hops away, so intermediate nodes will relay these transmissions. We assume that no local fusion takes place as the nodes' data is relayed to the CH.

It is during this data collection phase that the behavior becomes quite interesting. From the moment it starts, we want to determine the best decision that the CH can make given the multi-hop network that exists at time t_2 . If t_2 is close to t_1 , the sensors would not have spread out very much, so each one may be within one wireless hop of the CH. If t_2 is significantly greater than t_1 , then the nodes may be either very spread out or, if they have all gotten close to the stadium entrance, have begun to bunch up again. In any of these cases, there will be communication errors, but more errors will occur when the nodes are spread out because additional transmissions are needed to relay the data. The algorithms we use for distributed detection account for these situations and can even determine how much energy each node should dedicate to sending its data—where the energy is measured in terms of the number of bits the node sends.

We develop analytical models of this scenario and *numerical methods* to calculate the best possible performance that can be expected at any time t . We are thus interested in the behavior of the network as it changes over time, where these changes are due to the motion of the crowd. The transient behavior of stochastic models of this motion is critical to understanding this scenario and eventually determining the optimal time at which to make a decision.

We must first model how sensors organize into a cluster, which must be done very rapidly in mobile, event-driven scenarios. We thus assume the existence of a very fast, low-complexity algorithm for clustering that is triggered by an event—in this case the request by one node for other nodes' data. The algorithm we choose is the one defined and analyzed in [5, 6]. We will only consider the single-cluster case that is created in the scenario in this paper; the more general case of hierarchical clustering is relevant if we need to collect results from *all* sensors participating in an event. In this more general case, many thousands of sensors may be involved, one for each person attending the event.

In the single-cluster scenario, the communication architecture is some variation of the 2-hop cluster shown in Figure 2. The sensor in the center of the cluster, called the CH, is the one that requested the decisions from other sensors. The sensors shown in the first ring are one wireless hop from the CH; the ones in the second ring are two hops away, and so forth. Three sensors in a sector of ring 2 are shown forwarding their decisions to one sensor in ring 1. Each sensor's observation is affected by measurement noise and all communications throughout the network will be affected by channel noise, fading, and transceiver errors.

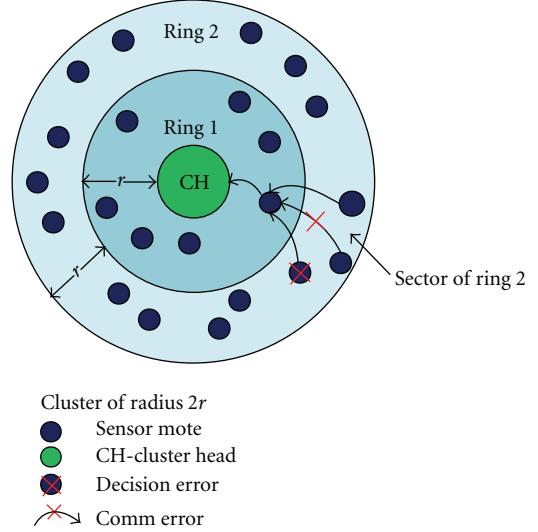


FIGURE 2: A 2-hop, 2D cluster. The mobility of the sensors produces this temporary, multi-hop cluster. Each sensor's decision may be incorrect because of measurement noise; transmitted packets may suffer bit errors because of noisy communication channels. If the number of hops in the cluster increases with time, the energy required for communication will increase and detection performance will decrease.

Because the cluster is multi-hop, decisions forwarded from the outer rings will suffer repeated exposure to the sources of communication error and more energy must be expended to get them to the CH.

The mobility model governing each sensor and the time since the event of interest will clearly affect the number of hops, and thus both the probability of detection at the CH and the energy consumed to gather the local decisions. It will also affect the network's performance in terms of coverage, maximum throughput, and throughput-delay trade-offs.

3. Prior Work on Mobility Models and Distributed Detection

Many mobility models have been proposed for analyzing the behavior of mobile ad hoc networks like the one described previously. In [7, 8], those models are categorized into four classes: random models, models with temporal dependency, models with spatial dependency, and models with geographic restrictions. For instance, random walks and random waypoint models are random models; Gauss-Markov mobility models and Smooth Random models are models with temporal dependency; group mobility models [7] are models with spatial dependency; Pathway mobility models and Obstacle mobility models are models with geographic restrictions. These models are useful as analytical or simulation tools but none of them can account for all of the constraints or features of real systems.

We thus use Correlated Random Walks (CRWs) as the model for the motion of the sensors in our security scenario. They can account for time dependency, geographical

restrictions, and nonzero drift. They are more general than random walks but are still amenable to analysis, sometimes in closed-form. The limiting distribution for a discrete-time 1D doubly-infinite CRW on the integers was derived in [9]. The probabilities of being at any lattice point at the n th step for a discrete-time 1D CRW in different cases were found in [10–13]. The absorbing probability and expected duration of a discrete-time CRW was found in [14]. The *transient behavior* of the continuous-time CRW on a semi-infinite, 1D state space, is solved, in closed-form in some cases, in [15]. The behavior of multiple sensors whose motion is modeled by discrete-time processes related to CRWs has been studied in [16]. In this paper, we use continuous-time, 1D CRWs on finite state spaces to calculate the behavior of mobile sensors in the security scenario described previously.

Ultimately, our goal is to determine—by numerical analysis instead of simulation—the effect of the mobility of the nodes on the detection problem that exists at the application layer of the mobile, ad hoc network. The performance of these applications will vary with time because of the nodes' motions. Understanding the significance of these transient effects, and being able to precisely determine such quantities as a lower bound on the probability of detection, is critical to developing a strategy for reaching a decision meeting certain criteria in the minimum amount of time.

In [17], an optimal distributed detection strategy in a single-hop network was studied. In [18], a decentralized detection problem under bandwidth-constrained communication was investigated. The noise at different sensors was assumed to be independent, but the statistics of the noise were assumed to be unknown to the CH, so it treats all received detection results equally. It is shown in [19] how the performance of detection algorithms is improved by knowledge of the channel. The limits of detection performance in a one-hop sensor cluster with nonideal channels were determined in [20].

In [21], the performance of distributed detection in a random sensor field is analyzed. The works of [22, 23] discuss the distributed detection problem for Gaussian signals under communication constraints. In [24], the authors consider the detection and localization problem of material releases with sparse sensor configurations. In [25], the sensors adopt robust binary quantizers for distributed detection. Censoring and sequential tests have also been adopted in distributed detection to achieve the same detection accuracy with less energy [26, 27], and they have been combined to further save energy [28, 29].

4. Basic Results on Transient Analysis of a Correlated Random Walk on $\{0, 1, \dots, \infty\}$

In this section, we briefly review results on the transient probability distributions of continuous-time, 1D CRWs on $\{0, 1, 2, \dots, \infty\}$ [15]. In this case, the sensor moves according to the following rules.

It takes a step in the same direction as its previous step with probability p_1 or p_2 depending on whether its previous step was in the positive or negative direction, respectively. It

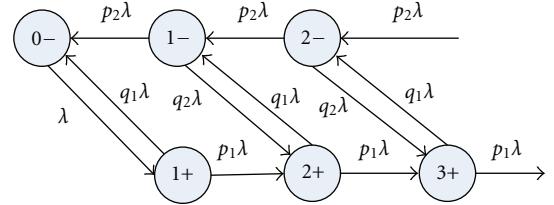


FIGURE 3: A Correlated Random Walk (CRW) on $\{0, 1, \dots, \infty\}$.

takes a step in the opposite direction of its previous step with probability $q_1 = 1 - p_1$ or $q_2 = 1 - p_2$ depending on whether its previous step was in the positive or negative direction, respectively. On reaching a (reflecting) boundary, it takes a step in the opposite direction with probability one.

The time at which the sensor takes its next step is governed by a Poisson process of intensity λ .

A CRW on $\{0, 1, \dots, \infty\}$ with a reflecting boundary at 0 can be modeled as a quasi-birth-death (QBD) process [30] with the state transition diagram shown in Figure 3. The state $n-$ at any level n is entered when the sensor moves to location n from location $n + 1$. The state $n+$ at any level n is entered when the sensor moves to location n from location $n - 1$.

Let

$$\pi(t) = [\pi_{0-}(t) \ \pi_{1+}(t) \ \pi_{1-}(t) \ \pi_{2+}(t) \ \pi_{2-}(t) \ \dots] \quad (1)$$

be the row vector of probabilities that the chain is in any of the possible states at time t , and define $\Pi(s)$ to be the Laplace transform of the vector $\pi(t)$. Letting Q denote the generator for this QBD process, then [31]

$$\Pi(s)(Q - sI) = -\pi(0), \quad \pi(0) = \pi(t)|_{t=0}. \quad (2)$$

For this QBD process,

$$Q = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ q_1\lambda & -\lambda & 0 & p_1\lambda & 0 & 0 & 0 & 0 & \dots \\ p_2\lambda & 0 & -\lambda & q_2\lambda & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & q_1\lambda & -\lambda & 0 & p_1\lambda & 0 & 0 & \dots \\ 0 & 0 & p_2\lambda & 0 & -\lambda & q_2\lambda & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & q_1\lambda & -\lambda & 0 & p_1\lambda & \dots \\ 0 & 0 & 0 & 0 & p_2\lambda & 0 & -\lambda & q_2\lambda & \dots \\ \vdots & \ddots & \end{bmatrix}. \quad (3)$$

For simplicity only, assume that the initial position of the sensor is 0; that is, $\pi(0) = [1, 0, 0, \dots]$. Define $\Pi_n(s)$ to be the transform of $\pi_n(t) = [\pi_{n+}(t), \pi_{n-}(t)]$, the row-vector of

transient probabilities for states on level n of the process. Then,

$$\begin{aligned} -(\lambda + s)\Pi_0(s) + \Pi_1(s) \begin{bmatrix} q_1\lambda \\ p_2\lambda \end{bmatrix} &= -1, \\ \lambda\Pi_0(s) + \Pi_1(s) \begin{bmatrix} -(\lambda + s) \\ 0 \end{bmatrix} &= 0, \end{aligned} \quad (4)$$

$$\Pi_n(s)B(s) + \Pi_{n+1}(s)C(s) = 0, \quad n \geq 1,$$

where

$$\begin{aligned} B(s) &= \begin{bmatrix} 0 & p_1\lambda \\ -(\lambda + s) & q_2\lambda \end{bmatrix}, \\ C(s) &= \begin{bmatrix} q_1\lambda & -(\lambda + s) \\ p_2\lambda & 0 \end{bmatrix}. \end{aligned} \quad (5)$$

Thus we get

$$\Pi_{n+1}(s) = \Pi_n(s)W(s), \quad (6)$$

where

$$W(s) = -B(s)[C(s)]^{-1} = \begin{bmatrix} \frac{p_1\lambda}{\lambda + s} & -\frac{p_1q_1\lambda}{p_2(\lambda + s)} \\ \frac{q_2\lambda}{\lambda + s} & \frac{\lambda + s}{p_2\lambda} - \frac{q_1q_2\lambda}{p_2(\lambda + s)} \end{bmatrix}. \quad (7)$$

The boundary variables must also satisfy the following:

$$\Pi_1(s)v_1(s) = 0, \quad (8)$$

where $v_1(s)$ denotes the right eigenvector corresponding to the eigenvalue of $W(s)$ whose magnitude is greater than or equal to 1 for all possible values of s . For this CRW, $W(s)$ is diagonalizable and its eigenvalues are given by

$$\gamma_k(s) = \frac{f(s) - (-1)^k \sqrt{(f(s))^2 - 4p_1p_2\lambda^2(\lambda + s)^2}}{2p_2\lambda(\lambda + s)}, \quad (9)$$

where $f(s) = (p_1 + p_2)\lambda^2 + 2s\lambda + s^2$, for $k = 1, 2$. The right eigenvectors $v_k(s)$, $k = 1, 2$, corresponding to the eigenvalues $\gamma_1(s)$ and $\gamma_2(s)$ are

$$v_k(s)$$

$$= \begin{bmatrix} f(s) - 2p_1p_2\lambda^2 + (-1)^k \sqrt{(f(s))^2 - 4p_1p_2\lambda^2(\lambda + s)^2} \\ 2p_2(p_2 - 1)\lambda^2 \end{bmatrix}. \quad (10)$$

We find numerically that, for all possible values of p_1, p_2, λ , and s , the magnitude of $\gamma_1(s)$ is greater than or equal to 1. Hence,

$$\begin{aligned} \Pi_{1+}(s) &= \frac{2\lambda(1 - p_2)}{A\lambda^2 + B(p_2)(s + \lambda)^2 + g(s)}, \\ \Pi_{1-}(s) &= \frac{f(s) - C\lambda^2 - \sqrt{(f(s))^2 - 2C\lambda^2(\lambda + s)^2}}{p_2\lambda(A\lambda^2 + B(p_2)(s + \lambda)^2 + g(s))}, \end{aligned} \quad (11)$$

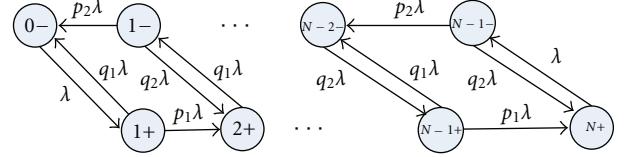


FIGURE 4: A Correlated Random Walk on $\{0, 1, \dots, N\}$.

where $g(s) = \sqrt{(s + \lambda)^4 - [B(p_1)B(p_2) + 1]\lambda^2(\lambda + s)^2 + A^2\lambda^4}$ and $A = p_1 + p_2 - 1$, $B(x) = 1 - 2x$, $C = 2p_1p_2$.

Thus by recursion,

$$\Pi_n(s) = \Pi_1(s)\widetilde{W}^{n-1}(s), \quad (12)$$

where $\widetilde{W}(s)$ is $W(s)$ after the mode whose eigenvalue is outside of the unit circle has been removed [31]. This ensures the stability of the recursion even when numerical errors occur in enforcing the orthogonality condition in (8).

5. New Results on Transient Analysis of Correlated Random Walks on $\{0, 1, \dots, N\}$

A CRW on $\{0, 1, \dots, N\}$ with reflecting boundaries at 0 and N can be modeled as a QBD process with the state transition diagram shown in Figure 4 and the following generator:

$$Q = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ q_1\lambda & -\lambda & 0 & p_1\lambda & 0 & 0 & \cdots & 0 & 0 \\ p_2\lambda & 0 & -\lambda & q_2\lambda & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & q_1\lambda & -\lambda & 0 & p_1\lambda & \cdots & 0 & 0 \\ 0 & 0 & p_2\lambda & 0 & -\lambda & q_2\lambda & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & q_1\lambda & -\lambda & 0 & p_1\lambda \\ 0 & 0 & 0 & 0 & \cdots & p_2\lambda & 0 & -\lambda & q_2\lambda \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \lambda & -\lambda \end{bmatrix}. \quad (13)$$

We assume without loss of generality that the initial position of the sensor is 0; that is, $\pi(0) = [1, 0, \dots, 0]$. Then we get

$$-(\lambda + s)\Pi_0(s) + \Pi_1(s) \begin{bmatrix} q_1\lambda \\ p_2\lambda \end{bmatrix} = -1,$$

$$\lambda\Pi_0(s) + \Pi_1(s) \begin{bmatrix} -(\lambda + s) \\ 0 \end{bmatrix} = 0,$$

$$\Pi_n(s)B(s) + \Pi_{n+1}(s)C(s) = 0, \quad 1 \leq n \leq N - 2. \quad (14)$$

$$\lambda\Pi_N(s) + \Pi_{N-1}(s) \begin{bmatrix} 0 \\ -(\lambda + s) \end{bmatrix} = 0,$$

$$-(\lambda + s)\Pi_N(s) + \Pi_{N-1}(s) \begin{bmatrix} p_1\lambda \\ q_2\lambda \end{bmatrix} = 0.$$

Thus, solving for $\Pi(s)$ in $\Pi(s)(Q - sI) = -\pi(0)$ is reduced to solving

$$[\Pi_0(s), \Pi_1(s), \Pi_{N-1}(s), \Pi_N(s)] \tilde{Q}(s) = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (15)$$

where, if we define I to be a 2×2 identity matrix, and $\bar{0} = [0, 0]$,

$$\tilde{Q}(s) = \begin{bmatrix} \begin{bmatrix} -(\lambda + s) & \lambda \\ q_1\lambda & -(\lambda + s) \\ p_2\lambda & 0 \end{bmatrix} & \bar{0} \\ W^{N-2}(s) & I \\ \bar{0} & \begin{bmatrix} 0 & p_1\lambda \\ -(\lambda + s) & q_2\lambda \\ \lambda & -(\lambda + s) \end{bmatrix} \end{bmatrix} \quad (16)$$

and the 2×2 matrix $W^{N-2}(s)$ can be computed, via the Cayley-Hamilton theorem, as a linear combination of $W(s)$ and I . It is then straightforward to calculate the transient distributions of the process via stable algorithms for transform inversion [32–34].

As a check on the aforementioned results for the transient behavior of the CRW, we can use the previous results to find its limiting distribution. Since $\pi(\infty) = \lim_{s \rightarrow 0} s\Pi(s)$, it is easy to show that the limiting probability of the CWR on $\{0, 1, \dots, N\}$ being at any location n is [13]

$$\begin{aligned} \bar{\pi}_0 &= \frac{1}{2} \frac{(p_1/p_2) - 1}{(p_1/p_2)^N - 1}, \\ \bar{\pi}_n &= \bar{\pi}_0 \left(1 + \frac{p_1}{p_2}\right) \left(\frac{p_1}{p_2}\right)^{N-1}, \quad n = 1, 2, \dots, (N-1), \\ \bar{\pi}_N &= \bar{\pi}_0 \left(\frac{p_1}{p_2}\right)^{N-1}. \end{aligned} \quad (17)$$

For the special case of $p_1 = p_2 = p$, $\bar{\pi}_0 = \bar{\pi}_N = 1/(2N)$, and $\bar{\pi}_n = 1/N$, for $n = 1, 2, \dots, (N-1)$. Note, though, that the transient behavior is of interest to us in the rest of this paper.

Note that this approach to analyzing the transient behavior of a simple CRW can be extended to model more complex motion. The number of states in each level of the QBD—two states in the case of the CRW in Figure 4—can be increased. With four to five states per level, the walker can pause, can walk at different speeds at different times, and so forth. The only difference in the algorithms for calculating the transient distribution is that the eigenvalues and eigenvectors of $W(s)$ may not be available in closed form. These techniques can also be extended to some classes of 2D CRWs [15].

6. Statistical Properties of Distances of Nodes from the CH

Suppose that the CH is motionless and there are a total of K sensors in the cluster. As described in the introduction, they

start at the same location as the CH ($n = 0$) and then move independently based on identical but independent CRW models. Define $x_i(t)$ as the location of the i th sensor at time t . Further define $d_{\max}(t)$ and $d_{\min}(t)$ as the maximum distance and the minimum distance between any sensor in the cluster and the CH, respectively. Then,

$$\begin{aligned} P(d_{\max}(t) = n) &= P(d_{\max}(t) \leq n) - P(d_{\max}(t) \leq n-1) \\ &= P(x_1(t) \leq n)^K - P(x_1(t) \leq n-1)^K \\ &= \left(\sum_{l=0}^n \pi_l(t) \right)^K - \left(\sum_{l=0}^{n-1} \pi_l(t) \right)^K, \\ P(d_{\min}(t) = n) &= P(d_{\min}(t) \geq n) - P(d_{\min}(t) \geq n+1) \\ &= P(x_1(t) \geq n)^K - P(x_1(t) \geq n+1)^K \\ &= \left(\sum_{l=n}^N \pi_l(t) \right)^K - \left(\sum_{l=n+1}^N \pi_l(t) \right)^K. \end{aligned} \quad (18)$$

Now assume that the CH is also mobile, that its position is governed by the same CRW as any sensor, and, as before, their motions are all independent. Then $d_{\max}(t)$ and $d_{\min}(t)$ become the maximum distance and the minimum distance between a specified sensor and any other sensor in the cluster. The probabilities remain the same whether the sensor is designated as the CH before or after motion of the sensors begins. Denote the position of the CH by $x_0(t)$. We have

$$\begin{aligned} P(d_{\max}(t) = n) &= \sum_{m=0}^N P(x_0(t) = m) P(d_{\max}(t) = n \mid x_0 = m) \\ &= \sum_{m=0}^N P(x_0(t) = m) [P(d_{\max}(t) \leq n \mid x_0 = m) \\ &\quad - P(d_{\max}(t) \leq n-1 \mid x_0 = m)] \\ &= \sum_{m=0}^N P(x_0(t) = m) \\ &\quad \times [P(m-n \leq x_i(t) \leq m+n, 1 \leq i \leq K) \\ &\quad - P(m-n+1 \leq x_i(t) \leq m+n-1, 1 \leq i \leq K)] \\ &= \sum_{m=0}^N \pi_m(t) \left[\left(\sum_{l=m-n}^{m+n} \pi_l(t) \right)^K - \left(\sum_{l=m-n+1}^{m+n-1} \pi_l(t) \right)^K \right], \\ P(d_{\min}(t) = n) &= \sum_{m=0}^N P(x_0(t) = m) [P(d_{\min}(t) \geq n \mid x_0 = m) \\ &\quad - P(d_{\min}(t) \geq n+1 \mid x_0 = m)] \\ &= \sum_{m=0}^N \pi_m(t) \left(\sum_{l=0}^{m-n} \pi_l(t) + \sum_{l=m+n}^N \pi_l(t) \right)^K \\ &\quad - \sum_{m=0}^N \pi_m(t) \left(\sum_{l=0}^{m-n-1} \pi_l(t) + \sum_{l=m+n+1}^N \pi_l(t) \right)^K. \end{aligned} \quad (19)$$

The probability distribution of the distance between any two sensors $d(t)$ is, for $n = 0$, $P(d(t) = 0) = \sum_{m=0}^N \pi_m(t)^2$, and, for $n \neq 0$,

$$P(d(t) = n) = \sum_{m=0}^N \pi_m(t)(\pi_{m+n}(t) + \pi_{m-n}(t)). \quad (20)$$

Define $D_{\max}(t)$ and $D_{\min}(t)$ as the maximum distance and the minimum distance between any two sensors in the cluster. Then,

$$P(D_{\max}(t) = 0) = \sum_{m=0}^N P(x_i(t) = m, 1 \leq i \leq K) = \sum_{m=0}^N \pi_m(t)^K, \quad (21)$$

and for $n \neq 0$,

$$\begin{aligned} & P(D_{\max}(t) = n) \\ &= \sum_{m=0}^{N-n} P(\min\{x_i(t)\} = m, \max\{x_i(t)\} = m+n, 1 \leq i \leq K) \\ &= \sum_{m=0}^{N-n} \sum_{l=1}^{K-1} \sum_{r=1}^{K-l} \binom{K}{l} \binom{K-l}{r} \pi_m(t)^l \pi_{m+n}(t)^r \\ &\quad \times \left(\sum_{s=m+1}^{m+n-1} \pi_s(t) \right)^{K-l-r}, \end{aligned} \quad (22)$$

where l sensors are at the location m and r sensors are at the location $m+n$, while all the others are in between. The calculation of the probability distribution of $D_{\min}(t)$ is much more complicated and, when $K > N$, it is always zero.

Thus, for both the motionless- and mobile-CH cases, computing $D_{\max}(t)$, $D_{\min}(t)$, $d_{\max}(t)$, and $d_{\min}(t)$ has been reduced to computing the transient distributions of CRWs. The computational techniques in Sections 2 and 3 thus enable fast and stable computation of statistics of interest in the structure of a cluster of mobile nodes as it evolves over time.

7. Detection in Wireless Sensor Networks

Here we assume a multi-hop sensor network in which each wireless hop is modeled as a Binary Symmetric Channel (BSC). The BSC cross-over probability captures the effect of channel errors on each individual decision bit and is easier to estimate than the full characteristics of the channel. The assumptions of symmetry and the same cross-over probability for each mote-to-mote channel are easily relaxed—they are used only to simplify the analysis so that important trends can be discerned. This paper also assumes that the error probabilities of the individual-received detection results are learned over time by the CH. Hence, the CH is assumed to know the optimal weights for the weighted median in the MAP detector [35] it uses to fuse the received detection results.

In [35], a MAP approach to the distributed detection problem in a multi-hop cluster in a sensor network was developed. It considered a cluster with K rings and N_k motes in the k th ring. Each mote makes a decision between two hypotheses, $s_0 = 0$ and $s_1 = 1$, where “1” denotes that an event has occurred and “0” that it has not. The detection results by different motes are assumed to be i.i.d. Bernoulli random variables, each with a detection error probability $p_m < 1/2$. The noise processes in different wireless channels are assumed to be independent and white. Each hop in the network is modeled as a BSC with cross-over probability $p_c < 1/2$. The decisions made by motes in the outer rings are relayed by the motes in inner rings to the CH. Let the error probability of the detection results received by the CH from the k th ring be $p_{e,k}$. Then, for example, $p_{e,1} = p_c(1 - p_m) + p_m(1 - p_c)$. Denote the detection result received by the CH from the i th mote in the k th ring by $r_{k,i}$ and arrange the detection results in the same ring in a vector $\bar{r}_k = (r_{k,1}, r_{k,2}, \dots, r_{k,N_k})$, $k = 1, 2, \dots, K$. Let E denote the event that a decision error happens at the CH.

In a *one-hop* cluster, suppose that the correct decision should be s . Assume the prior probability $p(s = s_0) = p < 1/2$ and find a real number χ such that $\ln((1 - p)/p) = \chi \ln((1 - p_{e,1})/p_{e,1})$. Define $W = \lfloor N_1/2 + \chi/2 \rfloor$. The MAP-based decision bit at the CH is $\hat{r} = (\bar{r}_1)_W = W^{\text{th}}$ order statistic of $(r_1, r_2, \dots, r_{N_1})$. The decision error probability at the CH is

$$\begin{aligned} P(E) &= (1 - p) \left(\sum_{i=W}^{N_1} \binom{N_1}{i} (p_{e,1})^i (1 - p_{e,1})^{N_1-i} \right) \\ &\quad + p \left(\sum_{i=N_1-W}^{N_1} \binom{N_1}{i} (p_{e,1})^i (1 - p_{e,1})^{N_1-i} \right). \end{aligned} \quad (23)$$

It is shown in [35] that, as an estimate of the true decision bit, this weighted order statistic is biased but asymptotically unbiased. When the prior probability is $p = 1/2$, it simplifies into a majority logic operator.

Now we summarize key results in [35] for the *multi-hop* case. For a mote that is k hops away from the CH, the detection result received by the CH after being relayed over these hops has error probability:

$$p_{e,k} = \frac{1}{2} - \frac{1}{2} (1 - 2p_m)(1 - 2p_c)^k, \quad k \geq 1. \quad (24)$$

We use the notation $W \diamondsuit x$, which means that x should be duplicated W times. For simplicity, suppose that the prior probabilities are $p(s = s_0) = p(s = s_1) = 1/2$.

Theorem 1. Define $\chi_k = \ln((1 - p_{e,k})/p_{e,k})$, and assume that these χ_k 's can be scaled so that $\chi_1 : \chi_2 : \dots : \chi_K = W_1 : W_2 : \dots : W_K$, where the W_k 's are positive integers with $\gcd(W_1, W_2, \dots, W_K) = 1$. The MAP-based decision bit is then given by $\hat{r} = \text{Median}(W_1 \diamondsuit \bar{r}_1, W_2 \diamondsuit \bar{r}_2, \dots, W_K \diamondsuit \bar{r}_K)$. The decision error probability at the CH is given by

$$P(E) = \sum_{\sum W_k(2c_k - N_k) > 0} \prod_{k=1}^K \binom{N_k}{c_k} (p_{e,k})^{c_k} (1 - p_{e,k})^{N_k - c_k}, \quad (25)$$

where c_k is the number of occurrences of $1 - s$ in the vector $\bar{r}_k, k = 1, 2, \dots, K$.

Because the weighted median is the MAP detector, we assume—even for sector-level fusion algorithms—that the CH calculates this weighted median after receiving inputs from all motes in the cluster. To understand what happens in these more complex cases, the asymptotic behavior of the weighted median must be determined for the single- and multi-hop cases; otherwise, it is too difficult to determine the probability of a clusterwide decision error at the CH.

The asymptotic behavior of the median filter in one-hop clusters has thus been studied using large deviation techniques in [36]:

$$\lim_{N_1 \rightarrow \infty} -\frac{1}{N_1} P(E) = C(p_{e,1}), \quad (26)$$

$$C(p_{e,1}) = -\ln(2) - \frac{1}{2} \ln(p_{e,1}(1 - p_{e,1})).$$

We now find the error exponent for the decision error probability for the multi-hop case.

Theorem 2. In multi-hop sensor networks in which $p_{e,k}$ is the error probability of the individual detection results received by the CH from nodes in ring k , the error probability of the MAP detector is upper bounded by [37]

$$\ln(P(E)) \leq \sum_{k=1}^K N_k \left[\ln(2) + \frac{1}{2} \ln(p_{e,k}(1 - p_{e,k})) \right]. \quad (27)$$

Proof. In the context of the weighted median filter, the decision error probability at the CH is

$$P(E) = P \left(\sum_{k=1}^K W_k \sum_{i=1}^{N_k} r_{k,i} \geq \sum_{k=1}^K \frac{W_k N_k}{2} \right) \quad (28)$$

$$= P \left(\sum_{k=1}^K \sum_{i=1}^{N_k} W_k \left(r_{k,i} - \frac{1}{2} \right) \geq 0 \right).$$

Since $P(Z \geq 0) \leq E[e^{Zt}]$, let $r'_{k,i} = r_{k,i} - 1/2$, and $E[r'_{k,i}] = p_{e,k} - 1/2 < 0$. We have

$$P(E) \leq E \left[e^{t \sum_{k=1}^K \sum_{i=1}^{N_k} W_k r'_{k,i}} \right] = \prod_{k=1}^K \prod_{i=1}^{N_k} E \left[e^{t W_k r'_{k,i}} \right]. \quad (29)$$

For each term in the product, $E[e^{t W_k r'_{k,i}}] = p_{e,k} e^{(1/2)t W_k} + (1 - p_{e,k}) e^{(-1/2)t W_k}$. Setting $\partial E[e^{t W_k r'_{k,i}}]/\partial t = 0$, we find

$$\frac{1}{2} W_k p_{e,k} e^{(1/2)t W_k} = \frac{1}{2} W_k (1 - p_{e,k}) e^{(-1/2)t W_k}, \quad (30)$$

$$t = 1. \quad \square$$

This error exponent is accurate and the bound is tight when the motes are densely deployed; that is, the number of sensors is large. Minimizing $P(E)$ when the number of motes is finite is a very difficult combinatorial problem,

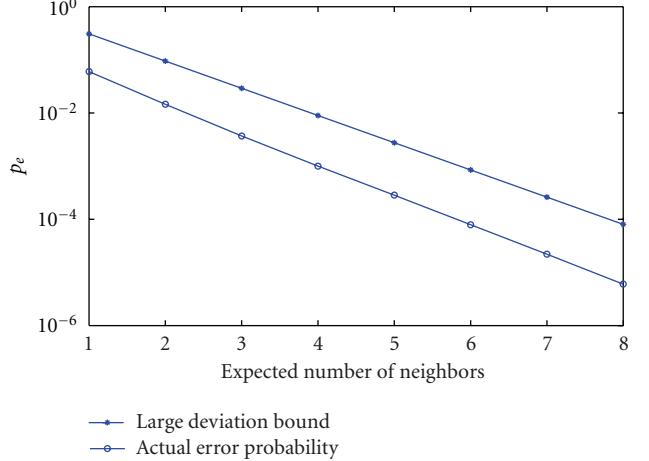


FIGURE 5: Comparison for a 3-ring cluster of the error probability obtained via simulation and its large deviations bound. The error exponent is clearly correct and the difference approaches zero asymptotically. Comparing the performance of complex strategies via their error exponents will thus correctly show which strategy is best when motes have large numbers of 1-hop neighbors. Direct calculations and simulations are used to confirm those results for small to moderate numbers of neighbors.

so we can minimize this upper bound instead. Also note that the effect of each ring of motes on the decision error probability at the CH is apparent in this bound. It may thus be used to simplify many optimization problems in distributed detection in multi-hop scenarios.

Figure 5 compares the large deviation bound on the error probability with the error probability from simulations for a 3-ring cluster. It shows the desired result that the large deviation error exponent is accurate—the bound is parallel with the simulation result over the entire range of spatial densities. Of course, the bound is high by a multiplicative factor, which is often the case with large deviation techniques.

If the CH is static, then we have

$$P(E) \leq E \left[\prod_{i=1}^K 2 \sqrt{p_{e,i}(1 - p_{e,i})} \right] \quad (31)$$

$$= \prod_{i=1}^K E \left[2 \sqrt{p_{e,i}(1 - p_{e,i})} \right]$$

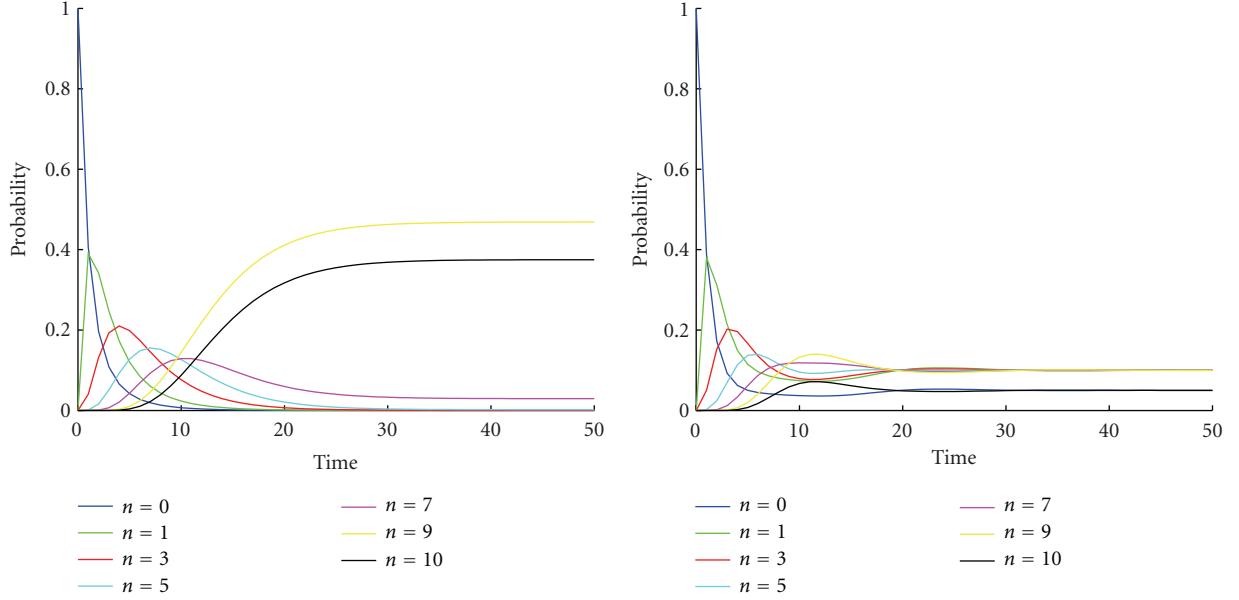
$$= \left[E \left[2 \sqrt{p_{e,1}(1 - p_{e,1})} \right] \right]^K,$$

where

$$p_{e,1} = \frac{1}{2} - \frac{1}{2} (1 - 2p_m)(1 - 2p_c)^{\lceil x_1(t)/r \rceil}. \quad (32)$$

The expected energy consumption for collecting the data is then

$$C = K \left\lceil \frac{E[x_1(t)]}{r} \right\rceil. \quad (33)$$



(a) An asymmetric CRW with $p_1 = 0.8$ and $p_2 = 0.2$. The walker starts at 0 and then drifts toward and tends to stay near 10

(b) A symmetric CRW with $p_1 = 0.9$ and $p_2 = 0.9$. The walker starts at 0 and then it is eventually equally likely to be in each position except a boundary state

FIGURE 6: Transient probability distribution calculated for two CRWs on $[0, 10]$. The curve for a given n shows the probability that the walker is at position n at time t .

If the CH is also mobile and its position is governed by a CRW, then we have

$$\begin{aligned} P(E) &\leq \sum_{m=0}^N P(x_0(t) = m) P(E | x_0(t) = m) \\ &= \sum_{m=0}^N P(x_0(t) = m) \prod_{i=1}^K E\left[2\sqrt{p_{e,i}(1-p_{e,i})} | x_0(t) = m\right] \\ &= \sum_{m=0}^N P(x_0(t) = m) \left[E\left[2\sqrt{p_{e,1}(1-p_{e,1})} | x_0(t) = m\right] \right]^K, \end{aligned} \quad (34)$$

where

$$p_{e,1} = \frac{1}{2} - \frac{1}{2}(1-2p_m)(1-2p_c)^{\lceil d(t)/r \rceil}. \quad (35)$$

The expected energy consumption for collection of the data is

$$C = K \left\lceil \frac{E[d(t)]}{r} \right\rceil. \quad (36)$$

8. Numerical Results

Figure 6 shows the transient probability distribution of two CRWs on $[0, 10]$ with different parameters. One CRW is asymmetric, which means that the walker tends, in the case shown, to prefer motion to the right. The other CRW is symmetric, which means that the walker has no preference

for one direction or the other—but its motion is still correlated, unlike in a standard random walk.

Figure 7 shows the transient probability distribution of the distance from a static clusterhead to the furthest of five mobile sensors for each point in time. The static clusterhead is assumed to be at location 0. This distance d is important for calculating the number of wireless hops between each sensor and the CH that is trying to gather data. If the transmission radius of each sensor is r , then the number of hops to the CH is the ceiling of d/r .

Figure 8 shows the transient probability distribution of the maximum distance between any of the five sensors and the CH when the mobile CH moves according to the same mobility model as the sensors. This is probably the most realistic case for mobile networks. It leads to an interesting phenomenon when the motion is on a finite grid and the CRWs are all asymmetric in the same direction. The sensors start at 0 when they gathered their observations. With time, they spread out but they all tend to move to the right. As they get close to the other barrier, they then tend to bunch up again. This effect is similar to a crowd moving in a given direction and then gathering around that destination. The distance here, when divided by the transmission radius of the sensors, tells the maximum size, in wireless hops, of the cluster.

Figure 9 shows the expected energy consumed—as a function of the time at which data collection starts—for a static or mobile CH to collect one packet of data from five mobile sensors. The six sensors all collected measurements at the same time/place but then continued to move. By the time a request to send in all data has been received, they are at locations in the state space with probabilities determined

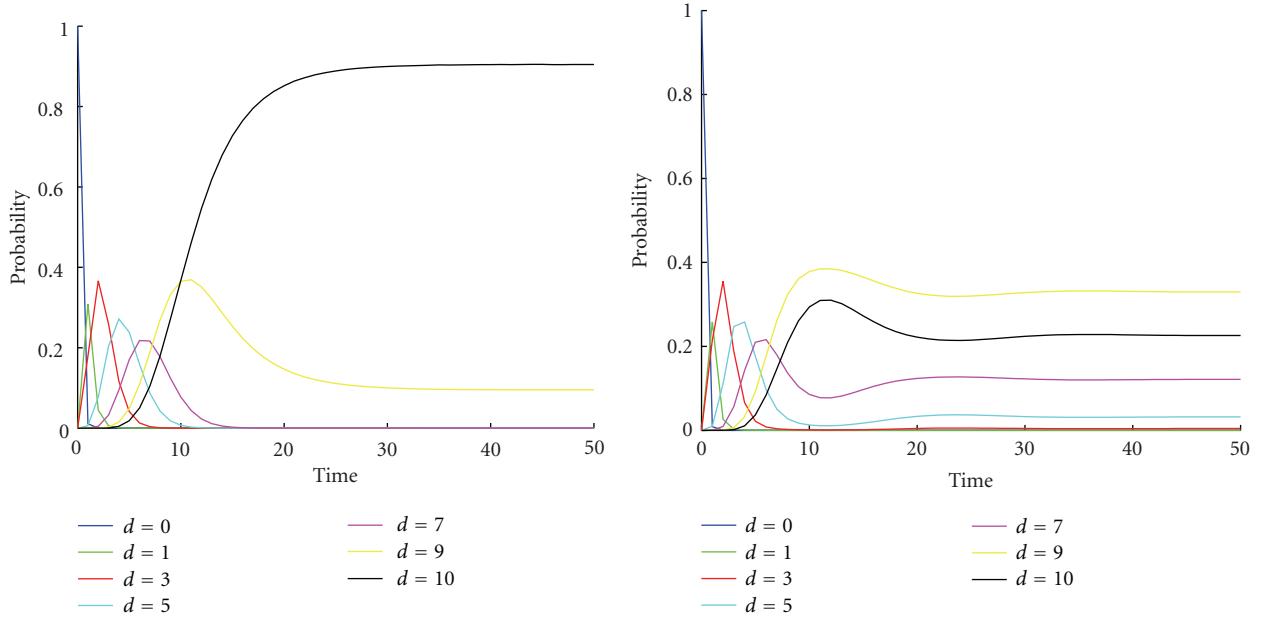


FIGURE 7: Calculations of the transient probability distribution of the distance d from a static CH to the furthest of five sensors that are moving according to independent CRWs on $[0, 10]$.

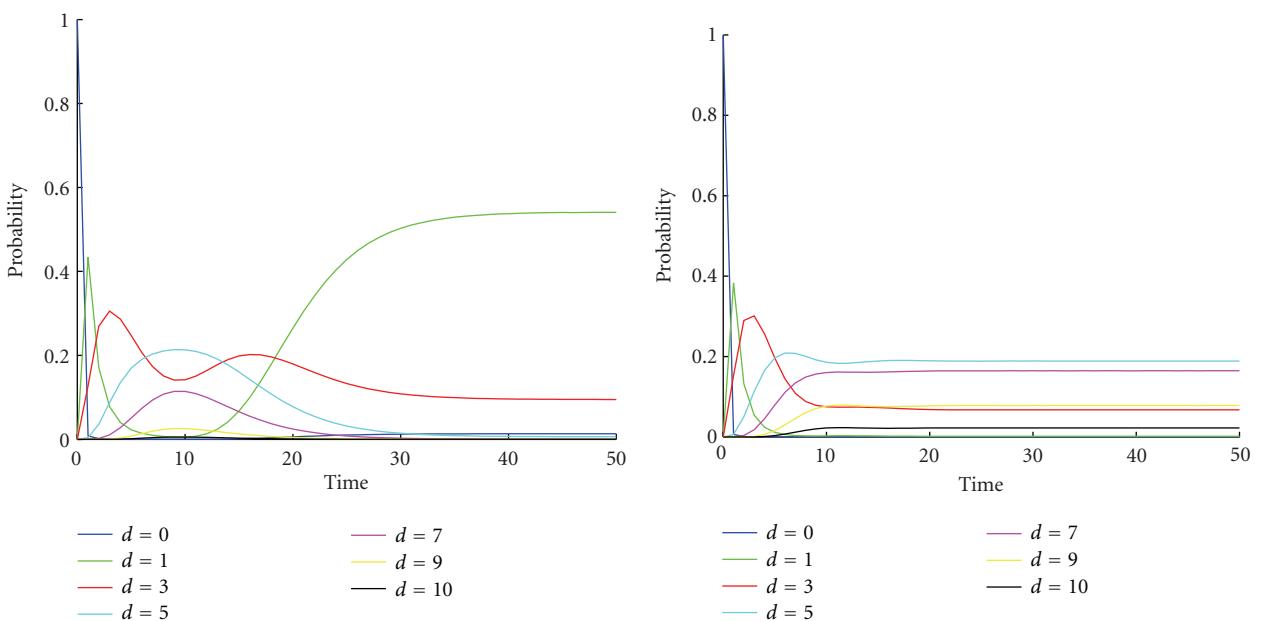
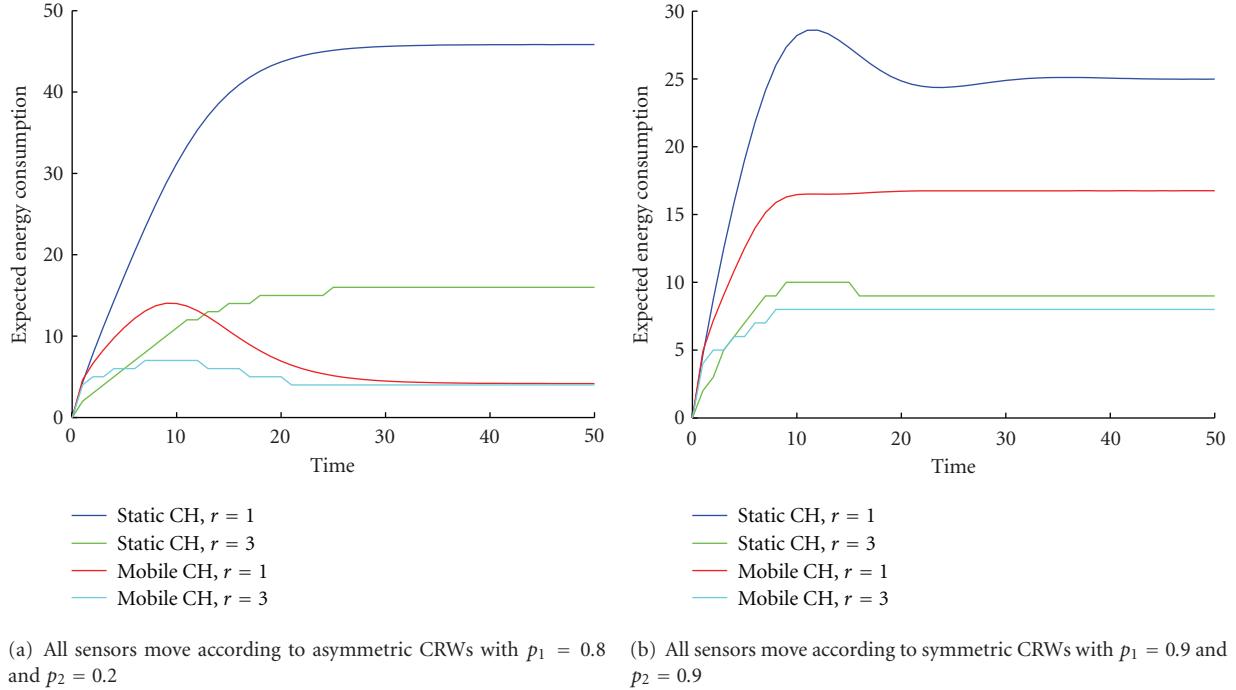
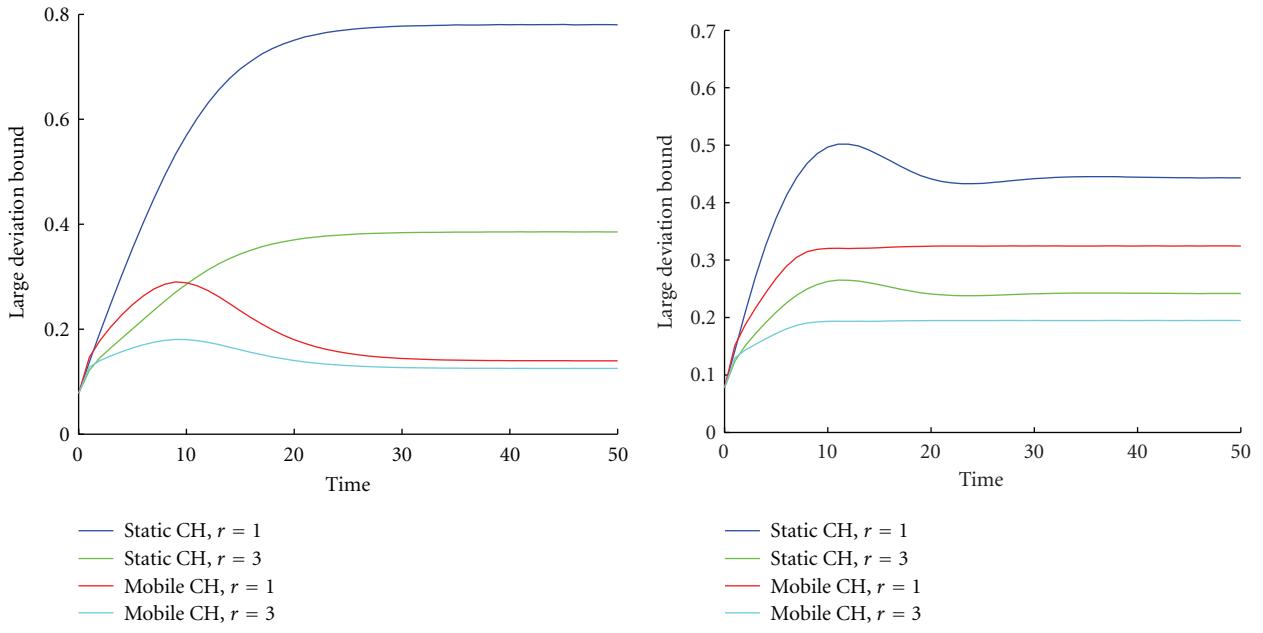


FIGURE 8: Calculations of the transient probability distribution of the distance d from a mobile CH to the furthest of five sensors for d from 0 to 10. The CH and the sensors are all moving independently and according to CRWs on $[0, 10]$.



(a) All sensors move according to asymmetric CRWs with $p_1 = 0.8$ and $p_2 = 0.2$ (b) All sensors move according to symmetric CRWs with $p_1 = 0.9$ and $p_2 = 0.9$

FIGURE 9: Calculation of the expected energy consumed by the network when the collection of data from the five mobile sensors begins at time t . The cases considered are when the transmission radius of each sensor is $r = 1$ or $r = 3$ and all sensors move according to symmetric or asymmetric CRWs.



(a) All sensors move according to asymmetric CRWs with $p_1 = 0.8$ and $p_2 = 0.2$ (b) All sensors move according to symmetric CRWs with $p_1 = 0.9$ and $p_2 = 0.9$

FIGURE 10: Calculation of the large deviation bound of the error probability when the collection of data from the five mobile sensors begins at time t . The cases considered are when the transmission radius of each sensor is $r = 1$ or $r = 3$ with $p_m = 0.10$ and $p_c = 0.05$ and all sensors move according to symmetric or asymmetric CRWs.

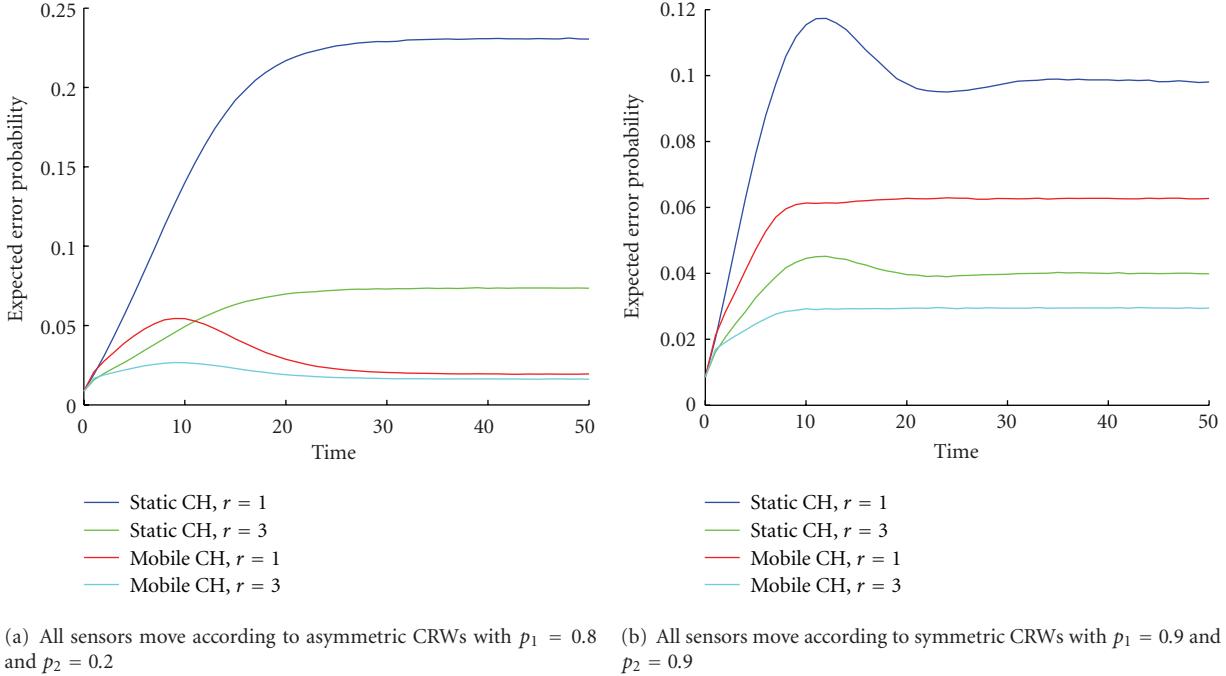


FIGURE 11: Expected error probability when the collection of data from the five mobile sensors begins at time t . The cases considered are when the transmission radius of each sensor is $r = 1$ or $r = 3$ with $p_m = 0.10$ and $p_c = 0.05$ and all sensors move according to symmetric or asymmetric CRWs. The curves in this figure were generated from 5 million runs of a simulation of the detection process for each setting of the mobility parameters. The values of the probabilities of the nodes locations for the detection process were obtained by simulations of the CRW.

by their CRWs. The data collected and decisions made by the sensors that have wandered the furthest require the most energy to collect—that data must travel over the largest number of hops to reach the CH.

Note that some curves in Figure 9 have a stepped appearance. This is due to the existence of the rings in the network that result from the fixed transmission radius for each node. When a node in ring i moves further from the cluster head, it may cross from ring i to ring $i + 1$. Its packets must then be relayed over one additional hop to reach the CH. This extra relay results in a significant jump in the energy that must be expended by the network.

Also note that the mobility of the CH has a significant impact on the energy consumed to collect data. If it is mobile and all sensors are moving according to a CRW that is asymmetric, then the energy required for collection first increases and then decreases. In this case, it is better to collect data either very quickly after the event, or, if that is not possible because of the time to process measurements, to wait until the CH and the other sensors bunch up again at their destination.

Figure 10 shows the large deviation bound of the error probability as a function of time for a static or mobile CH to collect one packet of data from five mobile sensors. It shows good agreement with the results in Figure 9, which shows the expected energy consumption as a function of time for a static or mobile CH to collect one packet of data from five mobile sensors.

Figure 11 shows the expected error probability as a function of time for a static or mobile CH to collect one packet of data from five mobile sensors. The same comments made previously about the effects of the mobility of the CH, and the effects of symmetric and asymmetric CRWs, apply here. The values of the probabilities of the nodes locations used to calculate these expected error probabilities were obtained in this case by simulations of the CRW.

9. Conclusions

In this paper, the solution to the finite state space CRW was provided and its statistical behavior was studied both analytically and numerically. As an illustration of the application-specific performance measures it can help address, we studied the impact of motion on the error probability of the final decision at the CH and the energy required to collect the decisions from all relevant sensors. Thus the temporary cluster head can decide the most appropriate time to call for data reports to help it make decisions within the allowed time frame and energy budget. In future research, we will study the impact of motion on the energy required to gather data and the error probability of the final decision in complex mobile wireless sensor networks.

References

- [1] X. Zhong, H. H. Chan, T. J. Rogers, C. P. Rosenberg, and E. J. Coyle, “The development and eStadium testbeds for research

- and development of wireless services for large-scale sports venues,” in *Proceedings of the 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM ’06)*, pp. 340–348, Barcelona, Spain, March 2006.
- [2] X. Zhong and E. J. Coyle, “eStadium: a wireless “living lab” for safety and infotainment applications,” in *Proceedings of the 1st International Conference on Communications and Networking in China (ChinaCom ’06)*, pp. 1–6, Beijing, China, October 2006.
 - [3] A. Ault, J. V. Krogmeier, S. R. Dunlop, and E. J. Coyle, “Stadium: the mobile wireless football experience,” in *Proceedings of the 3rd International Conference on Internet and Web Applications and Services (ICIW ’08)*, Athens, Greece, June 2008.
 - [4] “US DHS cell-all program,” http://www.dhs.gov/files/programs/gc_1268073038372.shtml.
 - [5] S. Bandyopadhyay, Q. Tian, and E. J. Coyle, “Spatio-temporal sampling rates and energy efficiency in wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 13, no. 6, pp. 1339–1352, 2005.
 - [6] S. Bandyopadhyay and E. J. Coyle, “Minimizing communication costs in hierarchically-clustered networks of wireless sensors,” *Computer Networks*, vol. 44, no. 1, pp. 1–16, 2004.
 - [7] F. Bai and A. Helmy, “A survey of mobility modeling and analysis in wireless ad hoc networks,” in *Wireless Ad Hoc and Sensor Networks*, Kluwer Academic Publishers, Boston, Mass, USA, 2004.
 - [8] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.
 - [9] S. Goldstein, “On diffusion by discontinuous movements, and on the telegraph equation,” *Quarterly Journal of Mechanics and Applied Mathematics*, vol. 4, no. 2, pp. 129–156, 1951.
 - [10] E. Renshaw and R. Henderson, “The correlated random walk,” *Journal of Applied Probability*, vol. 18, pp. 403–414, 1981.
 - [11] W. Böhm, “The correlated random walk with boundaries: a combinatorial solution,” *Journal of Applied Probability*, vol. 37, no. 2, pp. 470–479, 2000.
 - [12] A. D. Proudfoot and D. G. Lampard, “A random walk problem with correlation,” *Journal of Applied Probability*, vol. 9, pp. 436–440, 1972.
 - [13] R. Lal and U.N. Bhat, “Some explicit results for correlated random walks,” *Journal of Applied Probability*, vol. 27, pp. 757–766, 1989.
 - [14] Y. L. Zhang, “Some problems on a one-dimensional correlated random walk with various types of barriers,” *Journal of Applied Probability*, vol. 29, pp. 196–201, 1992.
 - [15] S. Bandyopadhyay, E. J. Coyle, and T. Falck, “Stochastic properties of mobility models in mobile ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 11, pp. 1218–1229, 2007.
 - [16] H. Cai and D. Y. Eun, “Toward stochastic anatomy of inter-meeting time distribution under general mobility models,” in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’08)*, pp. 273–282, Hong Kong, China, May 2008.
 - [17] K. Liu and A.M. Sayed, “Optimal distributed detection strategies for wireless sensor networks,” in *Proceedings of the 42nd Annual Allerton Conference on Communication, Control and Computing*, pp. 1651–1661, Monticello, Ill, USA, September 2004.
 - [18] Z. Q. Luo, “Universal decentralized detection in a bandwidth constrained sensor network,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2617–2624, 2005.
 - [19] B. Chen and P. K. Willett, “On the optimality of the likelihood-ratio test for local sensor decision rules in the presence of nonideal channels,” *IEEE Transactions on Information Theory*, vol. 51, no. 2, pp. 693–699, 2005.
 - [20] B. Chen, L. Tong, and P. K. Varshney, “Channel-aware distributed detection in wireless sensor networks,” *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 16–26, 2006.
 - [21] R. Niu and P. K. Varshney, “Performance analysis of distributed detection in a random sensor field,” *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 339–349, 2008.
 - [22] W. Li and H. Dai, “Distributed detection in wireless sensor networks using a multiple access channel,” *IEEE Transactions on Signal Processing*, vol. 55, no. 3, pp. 822–833, 2007.
 - [23] S. K. Jayaweera, “Bayesian fusion performance and system optimization for distributed stochastic Gaussian signal detection under communication constraints,” *IEEE Transactions on Signal Processing*, vol. 55, no. 4, pp. 1238–1250, 2007.
 - [24] E. B. Fox, J. W. Fisher, and A. S. Willsky, “Detection and localization of material releases with sparse sensor configurations,” *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 2172–2181, 2007.
 - [25] Y. Lin, B. Chen, and B. Suter, “Robust binary quantizers for distributed detection,” *IEEE Transactions on Wireless Communications*, vol. 6, no. 6, pp. 2172–2181, 2007.
 - [26] S. Appadwedula, V. V. Veeravalli, and D. L. Jones, “Decentralized detection with censoring sensors,” *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1362–1373, 2008.
 - [27] Y. Mei, “Asymptotic optimality theory for decentralized sequential hypothesis testing in sensor networks,” *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2072–2089, 2008.
 - [28] P. Addesso, S. Marano, and V. Matta, “Sequential sampling in sensor networks for detection with censoring nodes,” *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5497–5505, 2007.
 - [29] Y. Yao, “Group-ordered SPRT for distributed detection,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’08)*, pp. 2525–2528, Las Vegas, Nev, USA, April 2008.
 - [30] M. F. Neuts, *Matrix Geometric Solutions in Stochastic Models*, Johns Hopkins University Press, London, UK, 1981.
 - [31] J. Zhang and E. J. Coyle, “Transient analysis of quasi-birth-death processes,” *Communications in Statistics*, vol. 5, pp. 459–496, 1989.
 - [32] J. Abate, G. L. Choudhury, and W. Whitt, “An introduction to numerical transform inversion and its application to probability models,” in *Computational Probability*, W. Grassman, Ed., pp. 257–323, Kluwer, Boston, Mass, USA, 1999.
 - [33] J. Abate and W. Whitt, “The Fourier-series method for inverting transforms of probability distributions,” *Queueing Systems*, vol. 10, no. 1-2, pp. 5–87, 1992.
 - [34] J. Abate and P. P. Valko, “Multi-precision Laplace transform inversion,” *International Journal for Numerical Methods in Engineering*, vol. 60, no. 5, pp. 979–993, 2004.
 - [35] Q. Tian and E. J. Coyle, “Optimal distributed detection in clustered wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3892–3904, 2007.
 - [36] J. F. Chamberland and V. V. Veeravalli, “Asymptotic results for decentralized detection in power constrained wireless sensor

- networks,” *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1007–1015, 2004.
- [37] X. Sun and E. J. Coyle, “Low-complexity algorithms for event detection in wireless sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, Article ID 5555912, pp. 1138–1148, 2010.

Research Article

Modelling and Performance Analysis of a Network of Chemical Sensors with Dynamic Collaboration

Alex Skvortsov¹ and Branko Ristic²

¹HPP Division, Defence Science and Technology Organisation, 506 Lorimer Street, Fishermans Bend, VIC 3207, Australia

²ISR Division, Defence Science and Technology Organisation, 506 Lorimer Street, Fishermans Bend, VIC 3207, Australia

Correspondence should be addressed to Alex Skvortsov, alex.skvortsov@dsto.defence.gov.au

Received 14 July 2011; Revised 4 November 2011; Accepted 4 November 2011

Academic Editor: Frank Ehlers

Copyright © 2012 A. Skvortsov and B. Ristic. 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.

The problem of environmental monitoring using a wireless network of chemical sensors with a limited energy supply is considered. Since the conventional chemical sensors in active mode consume vast amounts of energy, an optimisation problem arises in the context of a balance between the energy consumption and the detection capabilities of such a network. A protocol based on “dynamic sensor collaboration” is employed: in the absence of any pollutant, the majority of sensors are in the sleep (passive) mode; a sensor is invoked (activated) by wake-up messages from its neighbors only when more information is required. The paper proposes a mathematical model of a network of chemical sensors using this protocol. The model provides valuable insights into the network behavior and near optimal capacity design (energy consumption against detection). An analytical model of the environment, using turbulent mixing to capture chaotic fluctuations, intermittency, and nonhomogeneity of the pollutant distribution, is employed in the study. A binary model of a chemical sensor is assumed (a device with threshold detection). The outcome of the study is a set of simple analytical tools for sensor network design, optimisation, and performance analysis.

1. Introduction

Development of wireless sensor network (WSN) for a particular operation scenario is a complex scientific and technical problem [1, 2]. Very often this complexity resides in establishing a balance between the peak performances of the WSN prescribed by the operational requirements (e.g., minimal detection threshold, size of surveillance region, detection time, rate of false negatives, etc.) and various resource constraints (e.g., limited energy supply, limited number of sensors, limited communication range, fixed detection threshold of individual sensors, limited budget for the cost of hardware, maintenance, etc.). The issue of resource constraints becomes even more relevant for a network of chemical sensors that are used for the continuous environmental monitoring (air and water pollution, hazardous releases, smoke, etc.). The reason is that a modern chemical sensor is usually equipped with a sampling unit (a fan for air and a pump for water), which turns on when the sensor is active.

The sampling unit usually requires a significant amount of energy to operate as well as frequent replacement of some consumable items (i.e., cartridges, filters). This leads to the critical requirement in the design of a WSN to reduce the active (i.e., sampling) time of its individual sensors.

One attractive way to achieve an optimal balance between the peak performance of the WSN and its constraints in resources mentioned above is to exploit the idea of dynamic sensor collaboration (DSC) [3, 4]. The DSC implies that a sensor in the network should be invoked (or activated) only when the network will gain information by its activation [4]. For each individual sensor, this information gain can be evaluated against other performance criteria of the sensor system, such as the detection delay or detection threshold, to find an optimal solution in the given circumstances.

While the DSC-based approach is a convenient framework for the development of algorithms for optimal scheduling of constrained sensing resources, the DSC-based algorithms involve continuous estimation of the state of each

sensor in the network and usually require extensive computer simulations [3, 4]. These simulations may become unpractical as the number of sensors in the network increases (e.g., “smart dust” sensors). Even when feasible, the simulations can provide only the numerical values for optimal network parameters, which are specific for an analysed scenario, but without any analytical framework for their consistent interpretation and generalisation. For instance, the scaling properties of a network (the functional relationship between the network parameters) still remain undetermined, which prevents any comprehensive optimisation study.

This motivates the development of another, perhaps less rigorous, but certainly simpler approach to the problem of network analysis and design. The main idea is to phenomenologically employ the so-called bioinspired (epidemiology, population dynamics) or physics-inspired (percolation and graph theory) models of DSC in the sensor network in order to describe the dynamics of collaboration as a single entity [5–10]. Since the theoretical framework for the bio- or physics-inspired models is already well established, we are in the position to make significant progress in the analytical treatment of these models of DSC (including their optimisation). From a formal point of view, the derived equations are ones of the “mean-field” theory, meaning that instead of working with dynamic equations for each individual sensor we only have a small number of equations for the “averaged” sensor state (i.e., passive, active, faulty, etc.), *regardless of the number of the sensors in the system*. A revealing example of the efficiency of this approach is the celebrated SIR model in epidemiology [11]. For any size of population, the SIR model describes the spread of an infection by using only three equations, corresponding to three “infectious” classes of the population: susceptible, infectious, and recovered.

The analytic or “equation-based” approach often leads to valuable insights into the performance of the proposed sensor network system by providing simple analytical expressions to calculate the vital network parameters, such as detection threshold, robustness, responsiveness, and stability and their functional relationships.

In the current paper, we develop a simple model of a wireless network of chemical sensors, where dynamic sensor collaboration is driven by the level of concentration of a pollutant (referred to as the “external challenge”) at each individual sensor. Our approach is based on the known analogy [10] between the information spread in a sensor network and the epidemics propagation across a population. In this analogy, the infection transmission process corresponds to message passing among the sensors. A chain reaction in transmission of an infection is called the epidemic. In the context of a sensor network, a chain reaction will trigger the network (as a whole) to move from the “no pollutant” state to the “pollutant present” state, which will indicate the presence of an external challenge.

The paper shows that the adopted epidemics or population-inspired approach can provide a reliable description of the dynamics of such a sensor network. The simple analytical formulas (scaling laws) derived from the model express the relationships between the parameters of the network (e.g., number of sensors, their density, sensing time, etc.), the

network performance (probability of detection, response time of a network), and the parameters of the external challenge (environment, pollutant). As an example of application of the proposed framework, we performed a simple optimisation study. Numerical simulations are carried out and presented in the paper in support of analytic expressions.

Although the model presented in this paper is specific to a network of chemical sensors, the underlying analytical approach can be easily adapted to other applications and other types of networks by a simple change of the model of environment and sensor.

2. The Model of Environment

The external challenges are modeled by a random time series which mimics the turbulent fluctuation of concentration at each sensor of the network. In this approach, the fluctuations in concentration C are modeled by the probability density function (pdf) of C with the mean C_0 as a parameter (i.e., C_0 is a mean concentration of the tracer in the area) [12]:

$$f(C | C_0) = (1 - \omega)\delta(C) + \frac{\omega^2}{C_0} \frac{(\gamma - 1)}{(\gamma - 2)} \left(1 + \frac{\omega}{(\gamma - 2)} \frac{C}{C_0}\right)^{-\gamma}. \quad (1)$$

Here, the value $\gamma = 26/3$ can be chosen to make it compliant with the theory of tracer dispersion in Kolmogorov turbulence (see [12]), but it may vary with the meteorological conditions. The parameter ω , which models the tracer intermittency in the turbulent flow, can be in the range $[0, 1]$, with $\omega = 1$ corresponding to the nonintermittent case. In general, it also depends on a sensor position within a chemical plume; thus, ω is in the range 0.95–0.98 near the plume centroid and may drop to 0.3–0.5 near the plume edge. For $\omega \neq 0$, the pdf f of (1) has a delta impulse in zero, meaning that the measured concentration in the presence of intermittency can be zero on some occasions. It can be easily shown that the pdf of (1) integrates to unity, so it is appropriately normalized.

The measured concentration time series can be generated by drawing random samples from the probability density function given in (1) at each time step. The random number generator is implemented using the *inverse transform* method based on the following steps [13]:

- (1) draw a sample u from the standard uniform distribution: $u \sim U[0, 1]$;
- (2) compute the value of C that satisfies $F(C) = u$, where $F(\cdot)$ is the cumulative distribution function (cdf) of the distribution of interest;
- (3) the value of C computed in the previous step is a random sample drawn from the desired probability distribution.

The cdf $F(\cdot)$ needed for inverse transform sampling is obtained by integrating the pdf in (1) and is given by

$$F(C | C_0) = 1 - \omega \left[1 + \left(\frac{2}{\gamma - 2}\right) \frac{C}{C_0}\right]^{1-\gamma}. \quad (2)$$

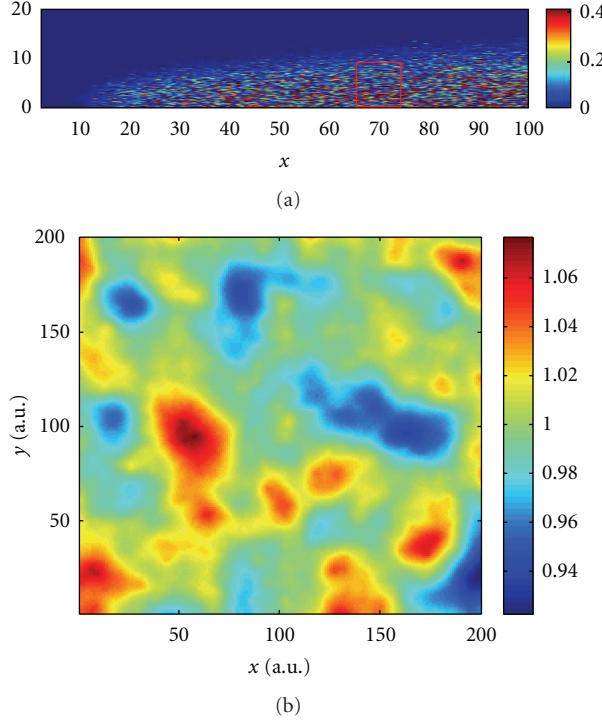


FIGURE 1: Example for concentration realisation for a plume-like flow (a) and the selected area of the flow with higher resolution (b).

The use of this cdf in the inverse transform sampling procedure generates the value of concentration

$$C = \begin{cases} C_0 \left(\frac{\gamma - 2}{\omega} \right) \left[\left(\frac{1-u}{\omega} \right)^{-1/(\gamma-1)} - 1 \right], & u \geq 1-\omega, \\ 0, & u < 1-\omega, \end{cases} \quad (3)$$

where u is again the standard uniform distribution $u \sim U[0, 1]$.

In order to produce spatial correlations that comply with the well-known scaling properties of turbulent dispersion, a special “swapping” algorithm was implemented. This recursive algorithm mimics the chaotic fluctuations occurring in the real turbulent flows (for details, see [14]).

The proposed framework allows to implement a reasonably realistic model of the contaminated environment (i.e., to generate the concentration realisation at each sensor over time), see Figure 1. Due to a universal nature of turbulence, it can be used to simulate performance of WSN in detection of either airborne and waterborne releases. The parameters γ and ω are typically estimated from geophysical observation (meteorological and oceanographical) and will be assumed known.

The geometrical complexity of the turbulent flow can be incorporated in the theoretical framework (2) by assuming a temporal and spatial variability of the mean concentration field $C_0 \equiv C_0(\mathbf{r}, t)$. This way we can simulate various morphologies of the flow (jet, wake, boundary layer, compartment flow, etc.) as well as various scenarios of hazardous

release (plume, puff), for details see [13, 15]. For the sake of simplicity in the current paper, we consider only case $C_0 = \text{const}$. This assumption corresponds to the approximation when the size of WSN is less than the width of hazardous plume (see Figure 1), or to an important practical case of a “highly distributed” source of pollutant (traffic, extended industrial site, or urban area [16]).

3. The Model of a Chemical Sensor

We adopt a simple binary (or “threshold”) model of a sensor, with the sensor reading V given by

$$V = \begin{cases} 1, & C \geq C_*, \\ 0, & C < C_*. \end{cases} \quad (4)$$

We emphasize that threshold C_* is an internal characteristic of the sensor, unrelated to C_0 in (1). This threshold is another important parameter of our model. A chemical sensor with bar readings, which includes many subsequent levels for concentration thresholds mapped into a discrete sensor output, is an evident generalisation of (4).

Using (3) and (4), it is straightforward to derive the probability of detection for an individual sensor embedded in the environment characterised by (2)

$$p = 1 - F(C_* | C_0). \quad (5)$$

This aggregated parameter links the characteristics of a specific sensor C_* , the parameter of the external challenge C_0 , and the environment $(F(\cdot), \gamma, \omega)$.

4. Modeling and Analysis of Network Performance

Our focus is a wireless network of chemical sensors with dynamic collaboration. We assume that N identical sensors (i.e., with the same detection threshold C_* and sampling time τ_*) are uniformly distributed over the surveillance domain of area S with density $\rho = N/S$.

We will model the following network protocol for dynamic collaboration. Each sensor can be only in one of the two states: *active* or *passive*. The sensor can be activated only by a message it receives from another sensor. Once activated, the sensor remains in the active state during an interval of time τ_* ; then it returns to the passive (sleep) state. While being in the active state, the sensor senses the environment, and if the chemical tracer is detected (binary detection), it broadcasts a (single) message. If a sensor receives an activation message while it is in the active state, it will ignore this message. The broadcast capability of the sensor is characterized by its communication range r_* , which is another important parameter of the model. The described protocol assumes that certain sensors of the network are permanently active. The number of permanently active sensors in the network is fixed, but the actual permanently active sensors vary over time in order to equally distribute the energy consumption of individual sensors.

The WSN following this protocol can be considered as a system of agents, interacting with each other (by means of message exchange) and with the stochastic environment (by means of sampling and probing). The interactions can change the state of agents (active and passive). From this perspective, this WSN is similar to the epidemic SIS (susceptible-infected-susceptible) model [11], in which an individual can be in only two states (susceptible or infected), and the change of state is a result of interaction (mixing) between the individuals (which corresponds to the exchange of messages in our case). Thus, a dynamic (population) model for our system [11] is as follows:

$$\frac{dN_+}{dt} = \alpha N_+ N_- - \frac{N_+}{\tau_*}, \quad (6)$$

$$\frac{dN_-}{dt} = -\alpha N_+ N_- + \frac{N_+}{\tau_*}, \quad (7)$$

where N_+ , N_- denote the number of active and passive sensors, respectively. The nonlinear terms on the RHS of (6) and (7) are responsible for the interaction between individuals (i.e., sensors), with the parameter α being a measure of this interaction. The population size (i.e., the number of sensors) is conserved, that is, $N_+ + N_- = N = \text{const}$.

The next step is to express α in terms of the parameters of our system by invoking physics-based arguments used in population dynamics [11]. It is well known that parameter α in (6) describes the intensity (contact rate) of social interaction between individuals in the community, so we can propose (see [11, 17])

$$\alpha \propto \frac{mp}{N\tau_*}, \quad (8)$$

where m is the number of contacts made by an “infected” sensor during the infectious period τ_* (i.e., the number of sensors receiving a message from an alerting sensor). In our case, we have $m = \pi r_*^2 \rho$. Then using $N = Sp$, we can write

$$\alpha = G \frac{\pi r_*^2}{\tau_* S} p, \quad (9)$$

where G is a constant calibration factor, being of order unity (it must be estimated during the network calibration); p was defined by (5). In order to simplify notation, from now on, we will assume that G is absorbed in the definition of r_* .

It is worth noting that by introducing nondimensional variables $n_+ = N_+/N$, $n_- = N_-/N$, and $\tau = t/\tau_*$, the system (6)-(7) can be rewritten in a compact nondimensional form

$$\frac{dn_+}{d\tau} = R_0 n_+ n_- - n_+, \quad n_- = 1 - n_+, \quad (10)$$

with only one nondimensional parameter

$$R_0 = \alpha \tau_* N. \quad (11)$$

The parameter R_0 is well known in epidemiology where it has the meaning of a *basic reproductive number* [11].

The system (6)-(7) combined with the condition $N_+ + N_- = N$ can be reduced to one equation for $y = N_+$,

$$\frac{dy}{dt} = \alpha y(N - y) - \frac{y}{\tau_*} = y(b - \alpha y), \quad (12)$$

where

$$b = \alpha N - \frac{1}{\tau_*} = \frac{(R_0 - 1)}{\tau_*}. \quad (13)$$

By simple change of variables $z = \alpha y/b$, this equation can be reduced to the standard logistic equation

$$\frac{dz}{dt} = bz(1 - z), \quad (14)$$

which has the well-known solution

$$z(t) = \frac{z_0}{(1 - z_0) \exp(-bt) + z_0}, \quad (15)$$

where $z_0 = z(0)$.

We can see that if $b < 0$, then $z \rightarrow 0$ as $t \rightarrow \infty$ for any z_0 , so any individual sensor activation in the network will “die out,” that is, the network will not be able to detect the external challenge. The same is valid for $b = 0$ when $z = z_0 = \text{const}$ (no response to external challenges). Only if the condition $b > 0$ is satisfied, then $z \rightarrow 1$ as $t \rightarrow \infty$ (independently of z_0). In this case, after a certain transition interval, the network will reach a new steady state with

$$\frac{N_+}{N} = 1 - \theta, \quad \frac{N_-}{N} = \theta, \quad \theta = \frac{1}{\alpha \tau_* N} \equiv \frac{1}{R_0}. \quad (16)$$

A fraction of active sensors N_+ at this new state is a measure of the network (positive) response to the event of chemical contamination. From (15), it is clear that the time scale for the network to reach the new state can be estimated from the condition $e^{-bt} \ll 1$, so

$$\tau \geq \frac{1}{b} = \frac{\tau_*}{R_0 - 1}. \quad (17)$$

This equation provides the relationship between the scale of activation time and parameter R_0 . One can see that this scale decreases as R_0 increases.

From (14), (17), it follows that an “epidemic threshold” for the sensor network is simply $\alpha \tau_* N > 1$ or in terms of the “basic reproductive number” (11),

$$R_0 = \alpha \tau_* N = p N \frac{\pi r_*^2}{S} > 1. \quad (18)$$

Observe that sensor sampling time τ_* has disappeared from the expression for R_0 . This means that it is possible to create an information epidemic (i.e., detect a chemical pollutant) for any value of τ_* , provided this time is long enough for a sensor to detect the chemical tracer. But according to (17), the responsiveness of the whole network to the external challenges (i.e., the time constant of detection) is, indeed, strongly dependent on the sensor sampling time $\tau = \tau_*/(R_0 - 1)$.

The expressions (16), (17), and (18) are the main analytical results of the paper. For a given level of external challenges (i.e., C_0) and meteorological conditions (i.e., γ , ω), these expressions provide a simple yet rigorous way to estimate how a change in the network and sensor parameters (i.e., N , C_* , τ_*) will affect the network performance (i.e., N_+ , τ).

We can also see that for a given external challenge the network of chemical sensors will respond in the most effective way when its parameters are selected in the combination which meets the criterion for “information epidemic” (18).

The final analytical expressions enable us to maximize the network information gain and optimize other parameters. For example, from (16), we can readily infer the important scaling properties of the network performance:

$$\frac{N_-}{N} \sim \frac{1}{r_*^2}, \quad \frac{N_-}{N} \sim \frac{1}{N}, \quad \frac{N_-}{N} \sim \frac{1}{p}. \quad (19)$$

For instance, if we double the communication range of an individual sensor r_* , the fraction of inactive sensors in the network will drop four times. Likewise, if we need to reach a specified fraction of active sensors ($1 - N_-/N$) to be able to reliably detect a given level of pollutant concentration, these formulas describe all possible ways of changing the parameters of the model in order to achieve this goal.

5. Information Gain of Collaboration

We have explained earlier that the concept of DSC is important for a network with limited energy/material resources. But the question remains will a network with DSC be inferior (in terms of detection performance) in comparison with a benchmark network where all sensors operate independently of each other and only report their (positive) detections of chemical pollution to the central processor for decision making? Clearly, such a benchmark network would be very expensive to run (all sensors would have to be active all the time), but could provide excellent detection performance.

In this section, we show that, under a certain condition, the network with DSC can provide superior detection performance compared to the benchmark network. Let us assume that we have δN sensors continuously operating ($0 \leq \delta \leq 1$). For a benchmark network, on average, we have $p\delta N$ sensors detecting pollutant. For the network with DSC, the same quantity can be estimated as $p(1 - \theta)N$ (as we have seen the saturation level of N_+ does not depend on initial conditions). From here, we can then deduce that the network with DSC will provide more information (for detection of chemical pollution) than the benchmark network if the following condition is satisfied:

$$\theta = \frac{1}{\alpha\tau_*N} \leq (1 - \delta), \quad (20)$$

which is eventually reduced to the condition of “epidemic threshold” (18) for the small value of δ .

The value of the parameter δ can be also estimated based on the following arguments. Let us assume that our aim is to detect a level concentration C_0 associated with a hazardous release within the time T (the constraint on time is driven by the requirement to mitigate the toxic effect of the release). Then, we can write a simple condition for the information “epidemic” in the WSN to occur during time T ,

$$\frac{\delta pNT}{\tau_*} \geq 1, \quad (21)$$

where p is given by (5), that is, $p = 1 - F(C_* | C_0)$. Evidently, for information epidemic to be observable, the number of continuously active sensors should be less than the number of sensors activated due to the hazardous release. Thus, from (20), we can write the following “consistency” condition for the minimum value of δ :

$$\delta_{\min} \approx \frac{\tau_*}{pNT} \leq \left(1 - \frac{1}{\alpha\tau_*N}\right), \quad (22)$$

or by rewriting it in terms of R_0 , see (16),

$$\delta_{\min} \approx \frac{\tau_*}{pNT} \leq \left(1 - \frac{1}{R_0}\right). \quad (23)$$

It can be seen that with other conditions being equal, the fraction of “stand-by” sensors δ_{\min} can be made however small (since $R_0 \geq 1$). It implies that only a small fraction of WSN will be active most of the time and is a clear demonstration of the energy consumption gain associated with the “epidemic” protocol.

Another important criteria for epidemic protocol can be derived by comparison of amplitude of “detectable events” for the *same number of sensors* in the network with DSC with the system of N -independent sensors. For the network with DSC, it is $(1 - \theta)N$ (since we use N_+ to retrieve information about the environment), and for the system of the *same independent sensors*, it is still pN (since N_+ is simply equal to N). Then instead of (20), we can write

$$\theta < (1 - p). \quad (24)$$

Under this condition, more detectable events will occur in the presence of chemical pollution by the described network with DSC (activation messages) than in a network of stand alone sensors (signals of positive detection). This leads to the interesting threshold condition on the number of sensors in the network

$$N > \frac{S}{\pi r_*^2} \frac{1}{p(1 - p)}. \quad (25)$$

The last term in RHS $(p(1 - p))^{-1}$ has an obvious minimum 4 corresponding to $p = 1/2$, so finally, we arrive at the simple universal condition

$$N > N_* = \frac{4}{\pi} \frac{S}{r_*^2}. \quad (26)$$

This condition reads that if the number of sensors in the system is greater than N_* , then networking with DSC *can* provide an information gain over the benchmark network. Under this condition, the network with DSC is not only desirable from the aspect of energy conservation, but also provides better detection performance through the information gain.

The condition $p = 1/2$ minimizing RHS of (25) can be considered as a criterion for an “optimal” sensor for a given network with DSC and for a given concentration of pollutant to be detected. Namely, from the equation $F(C_* | C_0) = 1/2$ and using (2), we can write

$$C_* = C_0 \left(\frac{\gamma - 2}{2} \right) \left[\left(\frac{1}{2\omega} \right)^{1/(1-\gamma)} - 1 \right]. \quad (27)$$

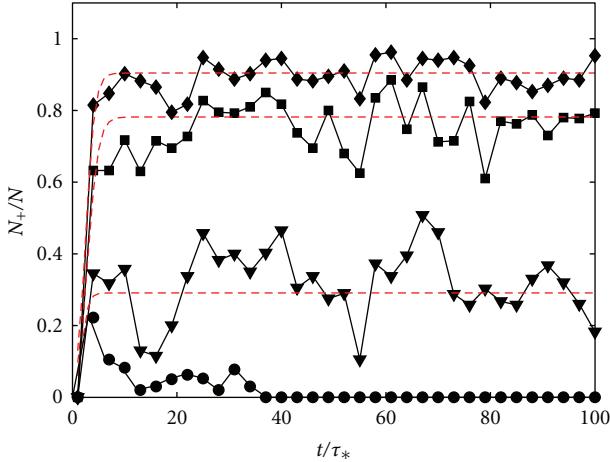


FIGURE 2: Results of numerical simulations: the fraction of active sensors in the network over time for the different communication range r_* : $\diamond—r_* = 40 \text{ m}$, $\blacksquare—r_* = 30 \text{ m}$, $\blacktriangledown—r_* = 27 \text{ m}$, and $\bullet—r_* = 20 \text{ m}$; $C_*/C_0 = 1.03$, $N_+(t=0) = 10$. The dashed red line corresponds to the analytical predictions (15). It is clearly seen that in the case $r_* = 20 \text{ m}$, the information epidemic in WSN dies off.

Given environmental parameters (γ, ω) and given the level of concentration to be detected (C_0), formula (27) also specifies a simple condition on detection threshold for an individual sensor to maximize an information gain by being networked.

6. Numerical Simulations

In support of analytical derivations presented above, a network of chemical sensors operating according to the adopted protocol for dynamic collaboration was implemented in MATLAB. A comprehensive report with numerical simulations result will be published elsewhere; here, we present only some illustrative examples.

For consistency, a $1000 \text{ m} \times 1000 \text{ m}$ surveillance region populated by $N = 400$ sensors with a uniformly random placement was assumed in all tests. In each run, chemical pollution with concentration $C_0 = 150$ is applied, and the simulation starts when a single randomly selected sensor (which has detected the presence of chemical contamination in its vicinity) starts broadcasting. Due to this random initiation and the fact that the probability of detection of individual sensors is less than unity ($p < 1$), each run of the computer program results in a slightly different outcome. Figures 2 and 3 show the average evolution of the ratio N_+/N in the network over time. The curves were obtained by using the following parameters: $\omega = 0.98$, $\gamma = 26/3$. Figure 2 demonstrates the changes in dynamics of the WSN for different values of communication range r_* , and Figure 3 depicts the similar plots for changes of the detection threshold of individual sensor C_* . For all plots in Figures 2 and 3, the initial number of active sensors is $N_+(t = 0) = 10$.

Overall, we found that the simulation output is much more sensitive to the changes of communication range than to the threshold of an individual sensor (see range of parameters depicted in Figures 2 and 3). In all cases,

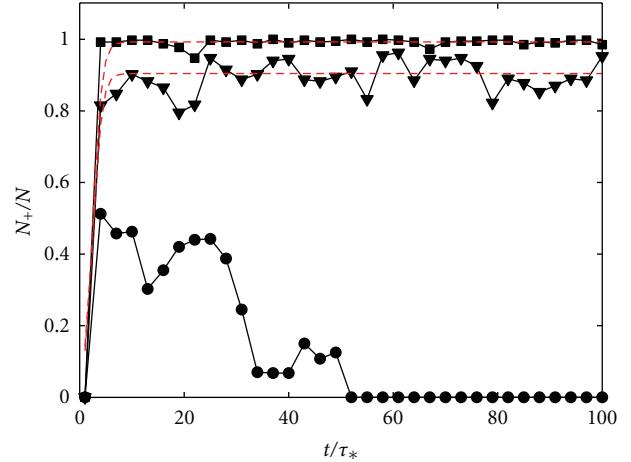


FIGURE 3: Results of numerical simulations: the fraction of active sensors in the network over time for the different threshold of individual sensor C_* : $\blacksquare—C_*/C_0 = 1.05$, $\blacktriangledown—C_*/C_0 = 1.02$, and $\bullet—C_*/C_0 = 1.00$; $r_* = 40 \text{ m}$, $N_+(t = 0) = 10$. The dashed red line corresponds to the analytical predictions (15). It is clearly seen that in the case $C_*/C_0 = 1.00$, the information epidemic in WSN dies off.

we observed the transition of N_+ from the initial steady state (where N_+ is very small indicating the absence of the pollutant) to the new steady state (high value of N_+), so information “epidemic” in the network of chemical sensors does occur. By direct substitution into (18), it was also validated that in all cases presented in Figures 2 and 3 the condition for an information “epidemic” was satisfied. In general, the saturation value of N_+ derived from these plots was in an agreement with theoretical prediction (16), but the estimated standard deviation of N_+ (not shown in Figure 2) could be very high (up to 30%) for some combination of parameters. The relative standard deviation (normalized by mean value N_+) usually gradually decreased over time and quite rapidly decays with the increase of communication range r_* . The occasional high variability of the output of the sensor network is undesirable and motivates further analysis. We also used the data from the plots in Figures 2 and 3 to calibrate our model. The calibration was performed by extracting the steady-state (or saturation) values of N_+ from the plots and by adjusting the “free” constant G in the analytical expressions (16) to achieve the best match between the analytical predictions and simulations. The value $G \approx 0.7$ seems to provide an optimal agreement with the presented simulations.

In order to validate our simple model for parameter α , we performed the following study. For each simulation, we derived the value of α_s from (16) and then compared it with the value of α_t calculated from the theoretical expression (9) using the calibration value $G \approx 0.7$. The results of this study are presented in Figure 4. The red dashed line corresponds to the perfect agreement between the theory and simulations. Considering the high variability of N_+ and a rather simple model for α , the agreement between the theory and simulations is acceptable.

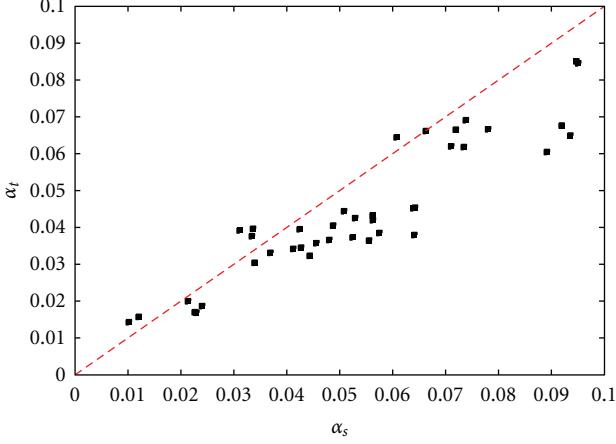


FIGURE 4: Simulation and theoretical predictions of parameter α : α_t is the theoretical value (9), α_s is the results of simulations. The red dashed line corresponds to the perfect agreement.

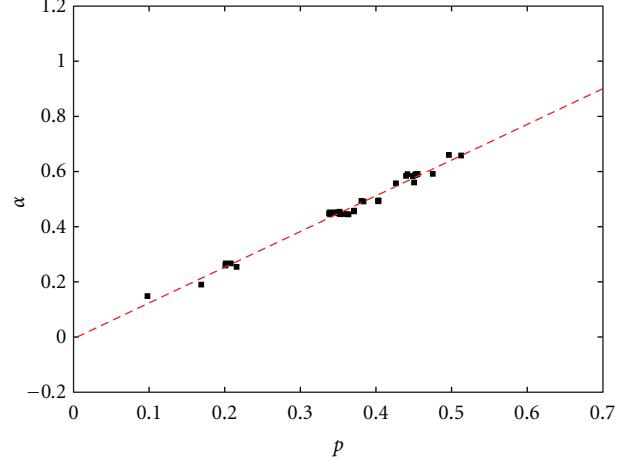


FIGURE 5: Parameter α as a function of p extracted from numerical simulations in log-log scale. The dashed line corresponds to the power-law fit $\alpha \propto p^q$, $q = 1.27$, and theoretical prediction corresponds to $q = 1$, see (8).

To validate further the alignment between the computer simulations and the proposed mathematical model, we numerically estimated some scaling properties of the network system (i.e., (9), (19)). Firstly, we derived the scaling properties from computer simulations and then compared them to the theoretical predictions. In general, we found that all trends of the derived scaling do agree with theoretical expressions in (19), but the quantitative agreement may significantly vary from case to case. As an illustration, in Figure 5, we present the plot of dependency of α against p in log-log scale. The extracted exponent corresponds to $\alpha \propto p^q$, where $q = 1.27$, while the theoretical value according to (8) is $q = 1$. This indicates that while our analytical model is very simple and fast to compute, for higher accuracy it may need further refinements as discussed below.

The results of numerical simulations presented above serve to verify that the “information epidemic” does occur in the wireless network of chemical sensors. This also implies that the proposed theoretical framework may lead to a gain in the energy consumption that may result in the significant advantages in operational deployment of such systems. More detailed analysis of the optimal values of parameters satisfying threshold conditions (18), (23), and (26) and lead to the optimal performance of WSN will be reported in separate publications.

7. Refinements of the Model

The disagreement described above is due to the implicit assumption of “homogeneous mixing” which we made in equations (6)-(7). The homogeneous mixing manifests itself in the bilinear form of the interaction terms on the RHS of (6)-(7). This bilinearity means that the number of new “infected” sensors is proportional to the product of the number which is currently “infected” and the number which is currently “susceptible.” Effectively it means that all passive sensors are equally likely to be activated. This assumption

holds only if the majority of activated (“infected”) sensors are far away from each other (i.e., at the distances $\gg r_*$). At some stage of the sensor “epidemic,” this assumption can be violated, because the secondary “infected” sensors will be at the shorter distances from the “infectious” parents (see Figure 6). The broadcasted messages in overlapping areas become duplicated and the rate of new “infections” will be no longer proportional to the number of their parents. The fraction of “infected” sensors in the overlapping areas will depend on the new equilibrium state of the sensor system (i.e., N_+/N as $t \rightarrow \infty$) and may not be small for some scenarios. To overcome this restriction, we again invoke an approach successfully implemented in epidemiology (see [17]). Instead of (6)-(7), we now write

$$\frac{dN_+}{dt} = \alpha N_+^\nu N_- - \frac{N_+}{\tau_*}, \quad \frac{dN_-}{dt} = -\alpha N_+^\nu N_- + \frac{N_+}{\tau_*}, \quad (28)$$

where a new parameter $0 \leq \nu \leq 1$ depends on the packing density of “infected” sensors (or on the ratio N_+/N). For a “sparse” network configuration, we have $\nu \approx 1$ (no overlapping areas), and for an extremely “dense” network, $\nu \approx 0$ (all sensors are located around the same point), see Figure 6. In general, ν can be used as a fitting parameter of the model [8] or estimated based on the mathematical theory of packing. For a specific network configuration, a value $\nu = 1/2$ was derived in [7] based on some simplified assumptions. By employing new parameter ν , we can significantly improve agreement between analytical model and simulation at the initial stage of information epidemic, since here we can assume that $N_- \approx N = \text{const}$, so $dN_+/dt \propto N_+^\nu$. An example of improved fitting is presented in Figure 7.

Similarly to the epidemiological models (see [11]), incorporation of the spatial inhomogeneity can be achieved by

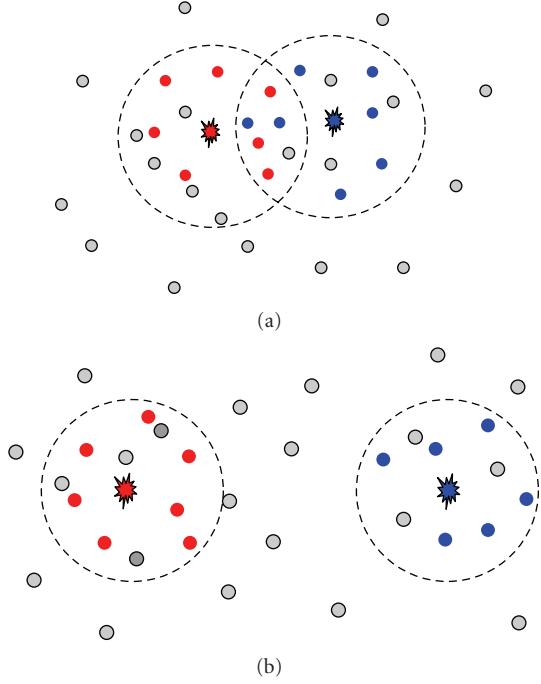


FIGURE 6: Examples of dense (a) and sparse (b) wireless sensor networks.

adding the appropriate diffusion terms on the LHS of (6) and (7):

$$\begin{aligned} \frac{\partial N_+}{\partial t} - D\Delta N_+ &= \alpha N_+ N_- - \frac{N_+}{\tau_*}, \\ \frac{\partial N_-}{\partial t} - D\Delta N_- &= -\alpha N_+ N_- + \frac{N_+}{\tau_*}, \end{aligned} \quad (29)$$

where D is diffusivity in the sensor system which can be estimated as $D \approx r_*^2/\tau_*$ and Δ is the Laplace operator. At the same time, the inhomogeneity of pollutant distribution can be easily incorporated in $\alpha(\mathbf{r})$ with nonuniform $C_0(\mathbf{r})$ (see (2), (5), (8)).

An important property of the system (29) is the existence of analytical solutions in the form of traveling waves, propagating with the velocity $v_0 \sim \sqrt{\alpha D}$ [11]. In our case, these waves correspond to the switching fronts between active and passive sensors. If pollutant is advected by the wind flow with a characteristic velocity v_* , then a simple synchronisation condition $v_0 \geq v_*$ or $\alpha \geq v_*^2 \tau_*/r_*^2$ provides an important criteria for network optimisation.

Another interesting extension of the proposed model is the introduction of the concept of a *faulty* sensor, a sensor which is no longer available for sensing and networking. This state of a sensor would correspond to the *removed* population segment in the epidemiological framework and can be attributed to any kind of faults (flat battery, software malfunction, hardware defects, etc.). As in the celebrated SIR epidemiological model [11], a new state results in the third equation for N_0 in the system (6)-(7) with a new temporal parameter—an average operational time (the lifespan) of a sensor. The total number of sensors will be still conserved:

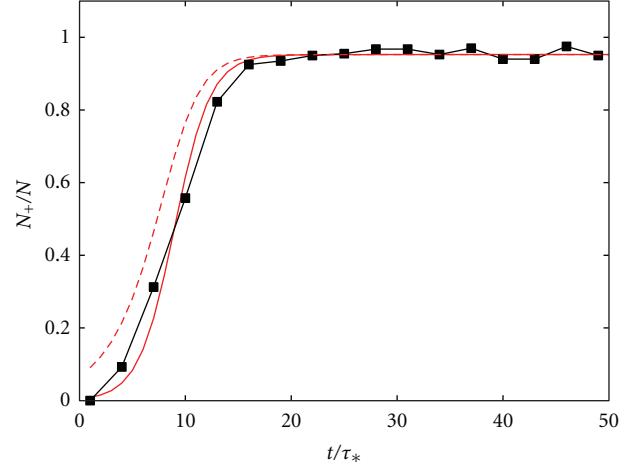


FIGURE 7: Effect of parameter ν in the model (28) on the simulation data fit: dashed line: $\nu = 1$, solid line: $\nu = 0.7$.

$N = N_+ + N_- + N_0 = \text{const}$. This model provides a more realistic representation of an operational sensor systems and allows us to estimate such important parameters as the operational lifetime of the network and the reliability of the network.

8. Conclusions

We developed a “bioinspired” model of a network of chemical sensors with dynamic collaboration for the purpose of energy conservation and information gain. The proposed model leverages on the existing theoretical discoveries from epidemiology resulting in a simple analytical model for the analysis of network dynamics. The analytical model enabled us to formulate analytically the conditions for the network performance. Thus, we found an optimal configuration which, within the underlying assumptions, yields a balance between the number of sensors, detected concentration, the sampling time, and the communication range. The findings are partly supported by numerical simulations. Further work is required to address the model refinements and generalisations.

Acknowledgments

The authors would like to thank Ralph Gailis, Ajith Gunatilaka, and Chris Woodruff for helpful technical discussions and Champake Mendis for his assistance in the software implementation of the proposed model.

References

- [1] C. S. Raghavendra, K. M. Sivalingam, and T. Znati, *Wireless Sensor Networks*, Springer, New York, NY, USA, 2005.
- [2] P. E. Bieringer, A. Wyszogrodzki, J. Weil, and G. Bieberbach, “An evaluation of propylene sampler grid designs for the FFT07 field program,” Tech. Rep., National Center for Atmospheric Research, Boulder, Colo, USA, 2006.

- [3] E. Ertin, J. W. Fisher, and L. C. Potter, "Maximum mutual information principle for dynamic sensor query problems," in *Information Processing in Sensor Networks*, vol. 2634 of *Lecture Notes in Computer Science*, pp. 91–104, 2003.
- [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] J. Mathieu, G. Hwang, and J. Dunyak, "The state of the art and the state of the practice: transferring insights from complex biological systems to the exploitation of netted sensors in command and control enterprises," MITRE Technichal Papers, MITRE Corporation, Bedford, Mass, USA, 2006.
- [6] A. Khelil, C. Becker, J. Tia, and K. Rothermel, "An epidemic model for information diffusion in MANETs," in *Proceedings of the 5th ACM international workshop on Modeling, Analysis and Simulation of wireless and mobile systems (MSWiM '02)*, pp. 54–60, Atlanta, Ga, USA, 2002.
- [7] P. De, Y. Liu, and S. K. Das, "An epidemic theoretic framework for evaluating broadcast protocolsin wireless sensor networks," in *Proceedings of the Proceedings of IEEE Internatonal Conferenceon Mobile Adhoc and Sensor Systems (MASS '07)*, pp. 1–9, Pisa, Italy, 2007.
- [8] B. Ristic, A. Skvortsov, and M. Morelande, "Predicting the progress and the peak of an epidemic," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09)*, pp. 513–516, Taiwan, April 2009.
- [9] A. Dekker and A. Skvortsov, "Topological issues in sensor networks," in *Proceedings of the Modelling and Simulation Society of Australia and New Zealand International Congress on Modelling and Simulation (MODSIM '09)*, Cairns, Australia, 2009.
- [10] S. Eubank, V. S. A. Kumar, and M. Marathe, "Epidemiology and wireless communication: tight analogy or loose metaphor?" in *Bio-Inspired Computing and Communication*, vol. 5151 of *Lecture Notes in Computer Science*, pp. 91–104, Springer, New York, NY, USA, 2008.
- [11] J. D. Murray, *Mathematical Biology*, vol. 1-2, Springer, New York, NY, USA, 2002.
- [12] V. Bisignanesi and M. S. Borgas, "Models for integrated pest management with chemicals in atmospheric surface layers," *Ecological Modelling*, vol. 201, no. 1, pp. 2–10, 2007.
- [13] A. Gunatilaka, B. Ristic, A. Skvortsov, and M. Morelande, "Parameter estimation of a continuous chemical plume source," in *Proceedings of the 11th International Conference on Information Fusion (FUSION '08)*, pp. 1–8, Cologne, Germany, July 2008.
- [14] A. Gunatilaka, A. Skvortsov, and R. Gailis, "Progress in DSTO CBR simulation environment development," in *Proceedings of the Land Warfare Conference (LWC '08)*, pp. 62–68, Brisbane, Australia, 2008.
- [15] A. Skvortsov and E. Yee, "Scaling laws of peripheral mixing of passive scalar in a wall-shear layer," *Physical Review E*, vol. 83, no. 3, part 2, Article ID 036303, 8 pages, 2011.
- [16] M. Jamriska, T. C. DuBois, and A. Skvortsov, "Statistical characterisation of bio-aerosol background in an urban environment," *Atmospheric Environment*. In press, <http://arxiv.org/abs/1110.4184>.
- [17] A. T. Skvortsov, R. B. Connell, P. D. Dawson, and R. M. Gailis, "Epidemic spread modeling: alignmentof agent-based simulation with a simple mathematical model," in *Proceedings of the International Conference on Bioinformatics and Computational Biology (BIOCOMP '07)*, vol. 2, pp. 487–490, CSREA Press, Las Vegas, Nev,USA, 2007.

Research Article

Energy-Balanced Density Control to Avoid Energy Hole for Wireless Sensor Networks

Jie Jia,¹ Jian Chen,² Xingwei Wang,¹ and Linliang Zhao¹

¹ School of Information Science & Engineering, Northeastern University, Shenyang 110819, China

² Research Institute, Northeastern University, Shenyang 110819, China

Correspondence should be addressed to Jie Jia, jiajie@ise.neu.edu.cn

Received 2 June 2011; Revised 1 September 2011; Accepted 2 October 2011

Academic Editor: Mandar Chitre

Copyright © 2012 Jie Jia 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.

Density control is of great relevance for wireless sensor networks monitoring hazardous applications where sensors are deployed with high density. Due to the multihop relay communication and many-to-one traffic characters in wireless sensor networks, the nodes closer to the sink tend to die faster, causing a bottleneck for improving the network lifetime. In this paper, the theoretical aspects of the network load and the node density are investigated systematically. And then, the accessibility condition to satisfy that all the working sensors exhaust their energy with the same ratio is proved. By introducing the concept of the equivalent sensing radius, a novel algorithm for density control to achieve balanced energy consumption per node is thus proposed. Different from other methods in the literature, a new pixel-based transmission mechanism is adopted, to reduce the duplication of the same messages. Combined with the accessibility condition, nodes on different energy layers are activated with a nonuniform distribution, so as to balance the energy depletion and enhance the survival of the network effectively. Extensive simulation results are presented to demonstrate the effectiveness of our algorithm.

1. Introduction

With the help of technological advances in MEMS, a mass production of tiny and economical sensors becomes possible. A wireless sensor network consists of a large number of sensor nodes deployed in region of interest to collect related information and communicate the results to the users [1]. The network can be embedded in our physical environment and have many potential applications, such as battlefield surveillance, environment monitoring, and fire detection.

Since the microsensors are usually supported by battery, they are thus limited in resources and vulnerable in nature. When sensor nodes are deployed to monitor hazardous applications like over a battlefield, an important question is to guarantee that the target area is covered and the detection probability is high. If a small number of nodes are deployed, blind spots or sensing holes might be left, which may reduce the accuracy of the results obtained. In order to enhance the reliability of the network, sensor nodes are usually deployed with high density, up to 20 nodes/m^3 . However, as all of the nodes share common sensing tasks, if those sensors operate

in the active mode simultaneously, data collected in such a high-density network would be highly correlated and redundant, consuming an excessive amount of energy. To illustrate the point, imagine the scenario that when a certain triggering event occurs, a large number of nodes will send packets at the same time, making such a network less responsive and less energy efficient. As a result, deploying such a sensor network to monitor hazardous applications and maintaining its sensing coverage could be a daunting task.

In general, density control is an effective method to solve the above problem. Recently, a class of work has appeared to find the optimal subset of sensor nodes for densely deployed wireless sensor network while these working nodes can completely cover the monitored area [2–8]. In addition, the problem is proved to be NP-complete [3]. However, most of the current works do not consider the issue of uneven energy depletion with distance to a predetermined sink. They all aimed to achieve a uniform hexagonal distribution to preserve area coverage with the fewest sensors. When using such a uniform distribution in many-to-one sensor network applications, the sensor nodes around the sink should forward

more data and deplete their energy faster. Consequently, an energy imbalance problem manifests itself, as an energy hole is created around the sink node. If this happens, no more data can be transmitted to the sink. Moreover, the network lifetime ends soon and more energy of the nodes would be wasted. Experimental results in [9] show that when the network lifetime is over, up to 90% of the total initial energy of the nodes is left unused if the nodes are distributed uniformly in the network. It becomes a major concern for network designers to maintain the balance of power consumption so that the lifetime of sensor network is prolonged.

In this paper, we formulate the energy imbalance problem and present a nonuniform distribution of sensor nodes to analyze the maximum network lifetime for many-to-one wireless sensor networks. In contrast to constant data acquisition rate, we import the pixel-based transmission mechanism to avoid sending needless duplication of the same sensing data. Furthermore, a density control algorithm is proposed to achieve balance of energy depletion by introducing the concept of the equivalent sensing radius. The rest of this paper is organized as follows. In Section 2, we review the related work in the literature. In Section 3, we theoretically analyze the nonuniform node distribution strategy. And after that, an energy-balanced density control algorithm is proposed in Section 4. Section 5 describes the simulation results of the proposed algorithm. Finally, the paper is concluded in Section 6.

2. Related Work

As one of the most fundamental issues in wireless sensor networks, the density control problem has attracted significant research attention. Therefore, coverage together with sensor management has been a strong research focus for the last few years. In [3], the authors provide a method for finding the maximum number of disjoint cover sets that are working successively in a WSN. In each cover set, a sufficient number of sensor nodes necessary to cover the targets are active, while the remainder of the nodes are put to sleep. However, their approach is based on a centralized solution. A distributed approach named PEAS is proposed in [4], in which the nodes use a simple rule to decide about their activity. If a node cannot find any active node in the probing range, it becomes active. Otherwise, it returns to the sleeping mode. Although this approach eliminates the complexity of getting neighbours' status, it does not require location information and cannot guarantee full sensing coverage for the target area. Similarly, the authors in [5] propose a scheduling scheme that enables each node to enter active or sleeping state based on the coverage relationship with its neighbours. In their approach, in order to avoid the "blind point" caused by two neighbours simultaneously turning off, a random back-off time is introduced before the node makes a decision about its status. However, these algorithms cannot achieve full sensing coverage for the target area. Our previous work reported in [6] attempted to find the best cover to maintain a full coverage of the network with the least number of working nodes, and a NSGA-II based approach was proposed. The coverage problem is also explored in [7], and a distributed,

localized algorithm, called OGDC (Optimal Geographical Density Control), is proposed to maintain coverage as well as connectivity. They prove that if the communication range is at least twice the sensing range, complete coverage implies connectivity. In particular, the jointly coverage and connectivity problem is studied in [8], and a sleep-aware scheduling scheme is proposed for energy conservation and surveillance quality provisioning. In [10], the coverage maintenance protocol named as PCP is proposed, and the simulation results show that it can significantly save the number of activated sensors by using probabilistic sensing model.

These algorithms all focused on finding a uniform distribution, thus to reduce the number of working nodes. However, as the sensors closer to the sink tend to carry more traffic loads and thus would consume more energy, the uniform deployment will cause the network lifetime descended by the sensors at the first-hop from the sink. This is also known as the "energy hole" problem, which is characterized by a mathematical model in [11]. Apparently, it cannot prolong the system lifetime under a uniform distribution by simply increasing the number of nodes. The authors in [12–14] have also investigated several approaches to mitigate this problem. In [12], the authors present a mathematical model and aim to investigate some approaches towards mitigating this energy hole problem. However, the uneven energy depletion still exists, even by using their mere system design and the associated routing strategy. The authors in [13] investigated the energy hole problem and designed guidelines for maximizing lifetime and avoiding energy holes in sensor networks with nonuniform distribution. In [14], the authors proposed a nonuniform deployment scheme based on a general sensor application model. They derived a formula to determine the number of nodes as a function of the distance from the sink. Simulation results show that their method can enhance the network lifetime. Since each sensor was also assumed to report the data to the sink with the same acquisition rate, it cannot achieve the energy balance completely in the entire network.

3. Preliminaries

3.1. Assumptions and Network Model. In this section, we present our network model and basic assumptions. Assume that a set of N heterogeneous sensors are deployed in a circular area with radius d in order to monitor some physical phenomenon. We refer to the complete set of sensors that has been deployed as $S = \{s_1, s_2, \dots, s_N\}$. Each sensor node has an ID, a fixed transmission range R_c , and a fixed sensing range R_s . Note that the location awareness is impractical in the highly dense network. In recent years, many research efforts have been made to address the localization problem [15–18]. However, this requirement can be relaxed slightly in our work if each node is aware of its relative location to the neighbours. The only sink node is located at the centre of the circle, as shown in Figure 1. We divide the area into n adjacent coronas with the same width of R_c and denote the i th corona by C_i . Obviously, the corona C_i is composed of nodes whose distances to the sink are between $(i - 1) * R_c$ and $i * R_c$.

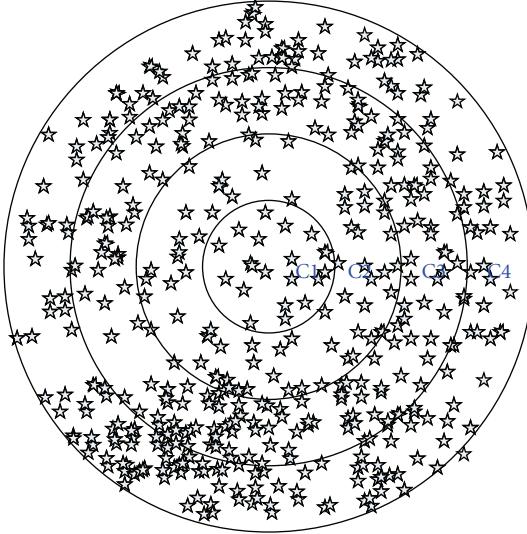


FIGURE 1: A circular target area with four coronas.

The network works in rounds, and each round is further divided into two phases: the first phase of node selection and the second phase of stability monitoring. In the first phase, the suitable sensor nodes are selected to work and the rest of the nodes are set to sleep state thus to save energy. During the second phase, each working node should send their sensing messages to the sink node per unit monitoring cycle ΔT . In order to avoid retransmitting the same messages in cross-covered areas, each sensor needs to check its own Voronoi polygon through the establishment of Voronoi graph with its neighbours before sending any data. In our work, this mechanism is called as the pixel-based transmission mechanism, which can ensure that the information data for any pixel in the target area is sent only once.

We use a simplified power consumption model and do not consider the MAC layer and physical layer issues. In our model, the energy consumption is only dominated by communication costs, as opposed to sensing and processing costs. The initial energy of each sensor is $\varepsilon > 0$, and the sink has no energy limitation. A node consumes e_1 units of energy when sending one bit, while it depletes e_2 units of energy when receiving one bit, where $e_1 > e_2 > 0$.

3.2. Nonuniform Node Distribution. Based on the network model, nodes belonging to corona $\{C_i | i \neq n\}$ will forward both the data generated by themselves and the data generated by coronas $\{C_j | (i+1) \leq j \leq n\}$ while the nodes in the outermost corona C_n need not forward any data. Assume that the sensors in each corona are distributed uniformly and there is no data aggregation at any forwarding nodes. Define the number of nodes deployed in corona C_i to be N_i and the number of pixels in corona C_i to be A_i . Based on the pixel-based transmission mechanism, the number of messages for corona C_i to receive and forward is $(A_{i+1} + A_{i+2} + \dots + A_n)$ and $(A_i + A_{i+1} + \dots + A_n)$. As the sensing messages are transmitted

per monitoring cycle ΔT , the average energy consumption for sensors in corona C_i during ΔT is

$$\bar{E}_i = \frac{[\sum_{k=i+1}^n A_k (e_1 + e_2) + A_i e_1]}{N_i}, \quad 1 \leq i \leq n-1. \quad (1)$$

Note that (1) can be simplified as

$$\bar{E}_i = \frac{e_1}{\rho_i} + \frac{A_{i+1} + A_{i+2} + \dots + A_n}{A_i \cdot \rho_i} (e_1 + e_2), \quad 1 \leq i \leq n-1, \quad (2)$$

where ρ_i is the node density of corona C_i .

Sensors in corona C_n only need to send their own sensing messages; so the energy depletion of sensors in corona C_n is

$$\bar{E}_n = \frac{A_n \cdot e_1}{N_n} = \frac{e_1}{\rho_n}. \quad (3)$$

Thus, we can formulate \bar{E}_i as follows:

$$\bar{E}_i = \begin{cases} \frac{e_1}{\rho_n}, & i = n, \\ \frac{e_1}{\rho_i} + \frac{\sum_{k=i+1}^n A_k}{A_i \cdot \rho_i} (e_1 + e_2), & 1 \leq i \leq n-1. \end{cases} \quad (4)$$

Ideally, when all the nodes deplete their energy with the same ratio, the network lifetime is prolonged and the energy efficiency is improved. In particular, there is no energy wasted and the network lifetime can be given by

$$\frac{\varepsilon}{\bar{E}_1} = \frac{\varepsilon}{\bar{E}_2} = \dots = \frac{\varepsilon}{\bar{E}_i} = \dots = \frac{\varepsilon}{\bar{E}_n}. \quad (5)$$

Theorem 1. *Maximum energy efficiency is possible, in the sense that all the working nodes take the pixel-based transmission mechanism, and the node distribution density ρ_i in corona C_i satisfies*

$$\rho_i = \rho_n \cdot \left[1 + \frac{(n^2 - i^2) \cdot (e_1 + e_2)}{(2i-1) \cdot e_1} \right], \quad \rho_1 \geq \rho_2 \geq \dots \geq \rho_n. \quad (6)$$

Proof. To use the deductive method, suppose that (6) is true, and thus (2) can be described as follows:

$$\begin{aligned} \bar{E}_i &= \frac{A_i \cdot e_1 + \sum_{k=i+1}^n A_k \cdot (e_1 + e_2)}{A_i \cdot \rho_i} \\ &= \frac{[A_i \cdot e_1 + \sum_{k=i+1}^n A_k \cdot (e_1 + e_2)] \cdot (2i-1) \cdot e_1}{A_i \cdot \rho_n \cdot [(2i-1) \cdot e_1 + (n^2 - i^2) \cdot (e_1 + e_2)]}. \end{aligned} \quad (7)$$

Owing to $A_i = \pi R_c^2 \cdot (2i-1)$, after basic transformations, we have

$$\begin{aligned} \bar{E}_i &= \frac{[(2i-1) \cdot e_1 + (n^2 - i^2) \cdot (e_1 + e_2)] \cdot (2i-1) \cdot e_1}{(2i-1) \cdot \rho_n \cdot [(2i-1) \cdot e_1 + (n^2 - i^2) \cdot (e_1 + e_2)]} \\ &= \frac{e_1}{\rho_n} = \bar{E}_n. \end{aligned} \quad (8)$$

Since $d\rho_i/di = -2i/(2i-1) - 2(n^2 - i^2)/(2i-1)^2 < 0$ is a permanent establishment, we can get the following conclusion, $\rho_1 \geq \rho_2 \geq \dots \geq \rho_n$. This completes the proof of Theorem 1. \square

Theorem 1 shows that in a circular monitored area, based on the pixel data transmission mechanism, if the sensors in each corona obey a nonuniform distribution and the distribution density meets a certain condition, the energy-balanced depletion of the whole network can be achieved. Besides, the node density ρ_i of corona C_i only relates to ρ_n of corona C_n and the corona number i .

Further we will analyze the lifetime enhancement of the nonuniform distribution strategy to the traditional one. Suppose that the node density in nonuniform distribution satisfies (6) and the initial conditions are the same. In the uniform distribution, the density ρ_i is equal to ρ_n . As the innermost corona C_1 needs to forward all of the sensing messages in the whole network, it consumes the most energy. Thus the maximum lifetime of network in uniform distribution is determined by the survival time C_1 . The network lifetime can be calculated as

$$\begin{aligned}\frac{\varepsilon}{\bar{E}'_1} &= \frac{\varepsilon}{e_1/\rho_n + \left(\sum_{j=2}^n A_j/A_1 \cdot \rho_n\right)(e_1 + e_2)} \\ &= \frac{\rho_n \cdot \varepsilon}{e_1 + (n^2 - 1) \cdot (e_1 + e_2)},\end{aligned}\quad (9)$$

where \bar{E}'_1 is the average energy depletion of C_1 per unit time in uniform distribution. Using (8), we can get the average energy depletion in C_1 under energy-balanced conditions as

$$\bar{E}_1 = \bar{E}_2 = \dots = \bar{E}_i = \dots = \bar{E}_n = \frac{e_1}{\rho_n}. \quad (10)$$

Thus the lifetime enhancement is

$$\frac{\varepsilon/\bar{E}_1}{\varepsilon/\bar{E}'_1} = \frac{(e_1 + (n^2 - 1) \cdot (e_1 + e_2))/\rho_n}{e_1/\rho_n} = \frac{\rho_1}{\rho_n} > 1. \quad (11)$$

Therefore, the network lifetime of nonuniform distribution can be extended ρ_1/ρ_n times effectively compared with the traditional uniform distribution strategy.

4. Energy-Balanced Density Control

4.1. Problem Formulation. The problem of Energy-Balanced Density Control (EBDC) is formalized as follows. Given a set of N potential sensors, $S = \{s_1, s_2, \dots, s_N\}$, find a subset $\text{cov} \subset S$, which achieves a nonuniform sensor distribution satisfying (6), and the number of sensors $K_{\text{cov}} = |\text{cov}|$ is minimized with a full coverage. The subset cov is named as the energy balance working cover for the target area.

4.2. Density Control Based on Equivalent Sensing Radius. The proposed algorithm, called EBDC, is inspired by the algorithm introduced in [7]. As a contribution, we made major modifications with the purpose of selecting sensors at variable densities according to (6).

Definition 2 (Equivalent sensing radius). It is defined as the sensing radius when the given distribution density ρ_i is the lowest one to maintain network coverage.

As the hexagonal distribution is the optimal sensor distribution to fully cover the target area with the fewest sensors,

define $\text{Hex}(i)$ to be the hexagonal area covered by sensor i with the sensing radius $R_i \cdot \text{Hex}(i)$ can be calculated as

$$\text{Hex}(i) = \frac{3\sqrt{3}}{2} R_i^2. \quad (12)$$

And the minimum distribution density ρ_i to fully cover the area is

$$\rho_i = \frac{1}{\text{Hex}(i)}. \quad (13)$$

Thus, the relationship of the equivalent sensing radius and the distribution density ρ_i is

$$R_i = \sqrt{\frac{2}{\sqrt{27} \cdot \rho_i}}. \quad (14)$$

Theorem 3. If the sensor selection algorithm uses the equivalent sensing radius R_i according to the density ρ_i , the network can achieve balanced energy depletion, where R_i satisfies

$$R_i = R_s \sqrt{\frac{(2i-1) \cdot e_1}{(2i-1) \cdot e_1 + (n^2 - i^2) \cdot (e_1 + e_2)}}, \quad (15)$$

$$R_1 \leq R_2 \leq \dots \leq R_n.$$

Proof. According to the definition of equivalent sensing radius, we can combine it with the energy-balanced condition in (6). Thus we have

$$\begin{aligned}\rho_i &= \frac{2}{\sqrt{27} \cdot R_i^2} \\ &= \frac{2}{\sqrt{27} \cdot R_s^2} \cdot \left[1 + \frac{(n^2 - i^2) \cdot (e_1 + e_2)}{(2i-1) \cdot e_1} \right].\end{aligned}\quad (16)$$

After transformation, we have

$$R_i = R_s \cdot \sqrt{\frac{(2i-1) \cdot e_1}{(2i-1) \cdot e_1 + (n^2 - i^2) \cdot (e_1 + e_2)}}. \quad (17)$$

This concludes the proof of Theorem 3. \square

Therefore, by introducing the concept of equivalent sensing radius, the problem of EBDC can be transformed into a uniform density control problem with different sensing radius, which gives the chance of using the existing schemes to solve it. In this paper, the density control algorithm is combined with OGDC approach, which only needs relative location during node selection.

In order to make sure that the node selected in each corona satisfies hexagonal distribution with its equivalent sensing radius, first, each sensor needs to know which corona is located. The calculation mechanism of corona number is presented in Section 4.2.1. And then the optimal principle for sensor selection is adopted [7]: anytime when a sensor in corona C_i is active, the next active node with the distance of $\sqrt{3}R_i$ away from the first one will be selected, and a similar selection method is used for the third node. Ideally, the centres of the three sensors should form an equilateral triangle with edge $\sqrt{3}R_i$.

4.2.1. Calculation of Corona Number. Since the sensors are deployed with high density, there are challenges to calculating each sensor's location and measuring the distance between sensor nodes accurately. Moreover, it seems to be impossible for sensor s_i to calculate corona number based on the distance $\text{dist}(s_i, \text{sink})$. On the other hand, as the corona number is equal to the minimum hop count from each sensor to sink, the corona number for each sensor can be calculated simply on the basis of its minimum hop count through routing. In our paper, this minimum hop count is calculated by using DV-hop localization algorithm [16].

The calculation of minimum hop count in DV-hop localization algorithm is similar to classical distance vector routing. At first, the sink node broadcasts a beacon to be flooded throughout the network containing its position with a hop-count parameter, which is initialized to be one. Then, each receiving node maintains the minimum counter value per anchor of all the beacons received and ignores those with higher hop-count values. At every intermediate hop, beacons are flooded outward with hop-count values incremented. Through this mechanism, all of the sensor nodes can get the minimum hops to the sink. The calculation of corona number based on hop count is shown in Figure 2.

4.2.2. Selection of the Starting Node. After all the sensors calculated their corona number and the corresponding equivalent sensing radius, they are powered on with undecided status. Then the node volunteer whose energy exceeds a predetermined power threshold P_t will become a starting node with probability p , where P_t is related to the length of the round. In general, it is set to a value so as to ensure that the sensor can remain powered on until the end of the round with high probability. And then, a back-off timer of τ_1 seconds is set, where τ_1 is distributed uniformly in $[0, T_d]$. When the timer expires, the node turns into the "ON" state and broadcasts a power-on message meanwhile. The power-on message is a quaternary array of $\langle \text{location}, R_i, \text{Corona_Num}, \alpha \rangle$, indicating the location of sensor, the equivalent sensing radius, the serial corona number, and the angle of next node. Note that α is used to determine the direction along which the second working node should be located. It is uniformly distributed in $[0, \alpha_d]$, where α_d represents the direction range of the next selecting nodes and is related to the location of the first selecting node. In the next section, we will give a detailed description of how to calculate α_d . A message-driven mechanism is used in the wake-up process. If the initial candidate node receives *power-on* messages before the back-off time finishes, its timer is cancelled and this node cannot become a starting node. This method helps to avoid many neighbours to become the starting node at the same time effectively. If the node does not volunteer itself to be a starting node, a timer of T_s seconds will be set to a sufficiently large value, such that there is at least one node whose power level qualifying to be a starting node and the selection of working nodes can be completed in an early stage of each round.

4.2.3. Actions Taken When Receiving a Power-on Message. When a node receives a power-on message, it first checks

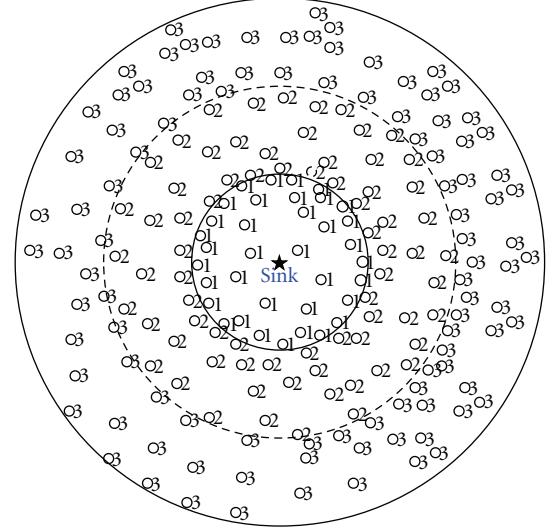


FIGURE 2: The calculation of corona number.

whether the *Corona_Num* are equal. If this message comes from an adjacent corona, and the receiving node is not "ON", or there are not any uncovered crossings, it will omit this message and sets itself to "OFF" state. Otherwise, it will become a starting node of its corona and transmits a new power-on message with new *Corona_Num* and a new equivalent sensing radius. If the power-on message comes from the same corona, the subsequent actions taken are to ensure that the working sensors selected form a hexagon distribution.

Similar to OGDC, T_{c1} , T_{c2} , and T_{c3} are back-off timers indicating the different retreated actions in different cases. In any of the above three cases, when the back-off timer expires, the node sets its state to "ON" and broadcasts a power-on message with a new direction field α set to -1 (indicating a message generated by a nonstarting node). The whole procedure of a node receiving a power-on message is shown in Figure 3.

4.2.4. Direction Range α_d in Power-on Message. The parameter α in power-on message indicates the direction along which next working node hoping to be activated. In terms of large coverage area, α is distributed uniformly in $[0, 2\pi]$. In order to distinguish the power-on message from starting nodes or nonstarting nodes, we set the latter α to -1. In terms of the node selection in different coronas with different equivalent sensing radius, the ratio of corona width to equivalent sensing radius R_c/R_i cannot be ignored. Therefore, when the OGDC scheme is implemented in each corona with R_i , the boundary effect must be considered. Furthermore, in order to speed the dissemination of power-on message in the same corona, it is also need to control the direction range α_d .

Define the coordinate of candidate starting sensor as $A(x_a, y_a)$, and the sink as (x_0, y_0) . Firstly, the distance between A and sink is calculated, and then the corona serial number as well as the equivalent sensing radius R_i is determined according to $\text{dist}(A, \text{sink})$. Based on this point, we can calculate the direction range α_d as follows: if the sensing disc

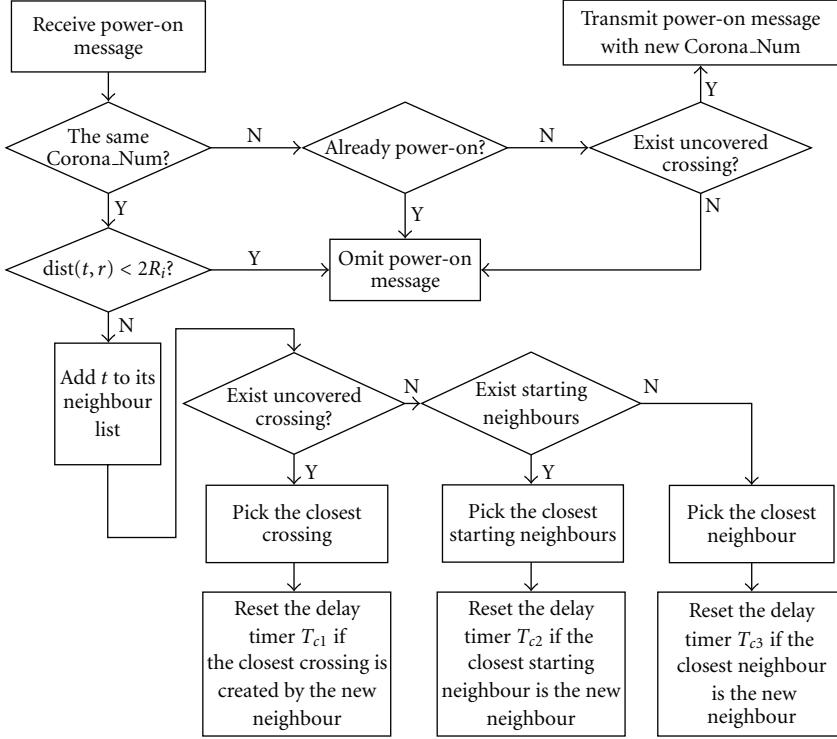


FIGURE 3: Flowchart of the actions taken after receiving a power-on message.

centred at A with radius $\sqrt{3}R_i$ has at most one crossing point with the inner or outer boundary of corona C_i , α_d is set to 2π , as shown in Figure 4(a). Otherwise, we have one of the three cases, as depicted from Figure 4(b) to Figure 4(d): (i) there exists two crossing points between the sensing disc and the inner border of the outer adjacent corona; (ii) there exists two crossing points between the sensing disc and the outer border of the inner adjacent corona; (iii) there exists four crossing points between the sensing disc and the border of both inner and outer adjacent coronas.

If case (i) satisfies, we have $\sqrt{(x_a - x_0)^2 + (y_a - y_0)^2} + \sqrt{3}R_i > i \cdot R_c$ and $\sqrt{(x_a - x_0)^2 + (y_a - y_0)^2} - \sqrt{3}R_i > (i-1) \cdot R_c$, as shown in Figure 4(b). The intersection coordinates can be obtained by the following formulation:

$$\begin{aligned} (x - x_a)^2 + (y - y_a)^2 &= 3R_i^2, \\ (x - x_0)^2 + (y - y_0)^2 &= (i \cdot R_c)^2. \end{aligned} \quad (18)$$

Assuming the calculated crossing points as $A_1(x_1, y_1)$ and $A_2(x_2, y_2)$, the direction range α_d is given by $\tan^{-1}((y_1 - y_a)/(x_1 - x_a)) \sim \pi + \tan^{-1}((y_2 - y_a)/(x_2 - x_a))$.

If case (ii) satisfies, we have $\sqrt{(x_a - x_0)^2 + (y_a - y_0)^2} + \sqrt{3}R_i < i \cdot R_c$ and $\sqrt{(x_a - x_0)^2 + (y_a - y_0)^2} - \sqrt{3}R_i < (i-1) \cdot R_c$, as shown in Figure 4(c). The intersection coordinates can be obtained by the following formulation:

$$\begin{aligned} (x - x_a)^2 + (y - y_a)^2 &= 3R_i^2, \\ (x - x_0)^2 + (y - y_0)^2 &= [(i-1) \cdot R_c]^2. \end{aligned} \quad (19)$$

Assuming the calculated crossing points as $A_1(x_1, y_1)$ and $A_2(x_2, y_2)$, the direction range α_d is given by $(0 \sim \tan^{-1}((y_1 - y_a)/(x_1 - x_a))) \cup (\pi + \tan^{-1}((y_2 - y_a)/(x_2 - x_a)) \sim 2\pi)$.

If case (iii) satisfies, we have $\sqrt{(x_a - x_0)^2 + (y_a - y_0)^2} + \sqrt{3}R_i > i \cdot R_c$ and $\sqrt{(x_a - x_0)^2 + (y_a - y_0)^2} - \sqrt{3}R_i < (i-1) \cdot R_c$, as shown in Figure 4(d). The intersection coordinates can be obtained by the following formulation:

$$\begin{aligned} c(x - x_a)^2 + (y - y_a)^2 &= 3R_i^2, \\ (x - x_0)^2 + (y - y_0)^2 &= (i \cdot R_c)^2, \\ c(x - x_a)^2 + (y - y_a)^2 &= 3R_i^2, \\ (x - x_0)^2 + (y - y_0)^2 &= [(i-1) \cdot R_c]^2. \end{aligned} \quad (20)$$

Using $A_1(x_1, y_1)$, $A_2(x_2, y_2)$, $A_3(x_3, y_3)$, and $A_4(x_4, y_4)$ to represent the calculated crossing points, we can get the direction range α_d :

$$\begin{aligned} \left(\tan^{-1}\left(\frac{y_1 - y_a}{x_1 - x_a}\right) \sim \tan^{-1}\left(\frac{y_2 - y_a}{x_2 - x_a}\right) \right) \\ \cup \left(\pi + \tan^{-1}\left(\frac{y_3 - y_a}{x_3 - x_a}\right) \sim \pi + \tan^{-1}\left(\frac{y_4 - y_a}{x_4 - x_a}\right) \right). \end{aligned} \quad (21)$$

After the direction range α_d is set, the nodes along α_d will be selected first, like nodes D and C in Figure 4(d). When node A becomes the power-on node, D and C will be the following active nodes to form a hexagonal distribution. We should note that although the node selection along range will

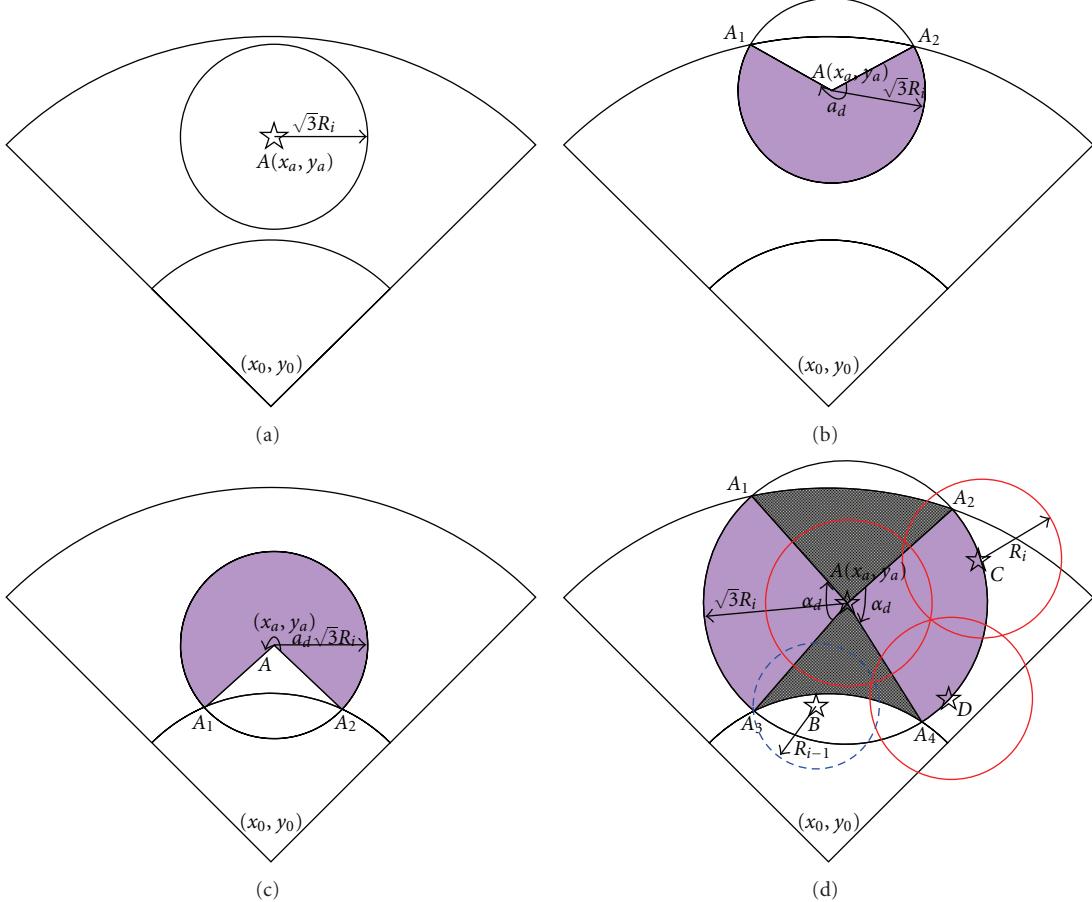


FIGURE 4: The possible intersection between sensing disc and the corona borders. (a) At most one crossing point. (b) Two crossing points between the sensing disc and the inner border of the outer adjacent corona. (c) Two crossing points between the sensing disc and the outer border of the inner adjacent corona. (d) Four crossing points between the sensing disc and the border of both inner and outer adjacent coronas.

result in the region out of α_d being uncovered originally, with the dissemination of the power-on message in the adjacent coronas, finally, it will be fully covered by the node in its adjacent coronas. Take node B in Figure 4(d) as an example. When node B receives a power-on message from node A , it will become the new starting node in corona C_{i-1} according to Figure 2, thus to cover the shadow area in corona C_i . To summarize, by calculating the direction range, we can achieve a rapid selection of the working nodes in the same corona and reduce the number of invalid power-on messages effectively.

5. Simulations Results

In this section, we evaluate the performance of the proposed density control algorithm. The basic simulation parameters are listed in Table 1.

Initially, in order to deploy more nodes close to the sink node, the deployment model of two-dimensional Gaussian distribution is adopted. Given the coordinate of sink as (x_0, y_0) , the node deployment density follows:

$$f(x, y) = f(x - x_0, y - y_0) = \frac{1}{2\pi\sigma^2} e^{-[(x-x_0)^2 + (y-y_0)^2]/2\sigma^2}, \quad (22)$$

where σ is the standard deviation of coordinate (x, y) , and it is equal to the communication radius R_c in our simulation.

As the finally selected nodes obey approximate uniform distribution in the corona in each round, the sensing data forwarding strategy is similar to [13]. Any node in corona C_i can communicate with almost q nodes in the ring C_{i-1} directly, where $q_i = \rho_{i-1} \cdot A_{i-1}/\rho_i \cdot A_i$. Among these q_i candidate forwarding nodes, the node with most residual energy will be selected as the forwarding node.

There are 1000 potential sensors randomly distributed in the circular area of radius 60 using Gaussian distribution deployment model, as shown in Figure 5(a). The target area is divided into three coronas denoted by C_1 , C_2 , and C_3 . From (17), we can calculate the equivalent sensing radius from C_1 to C_3 to be 2.8, 5.28, and 10.

Figure 5(b) shows the working sensors selected after running NSGA-II 500 generations. The number of working nodes selected from corona C_1 to C_3 is 24, 53, and 59. Further, those working sensors in Figure 5(b) are renamed as 1, 2, 3, ..., 136, where the sensors with the larger IDs belong to outer coronas and those with smaller IDs are closer to sink node.

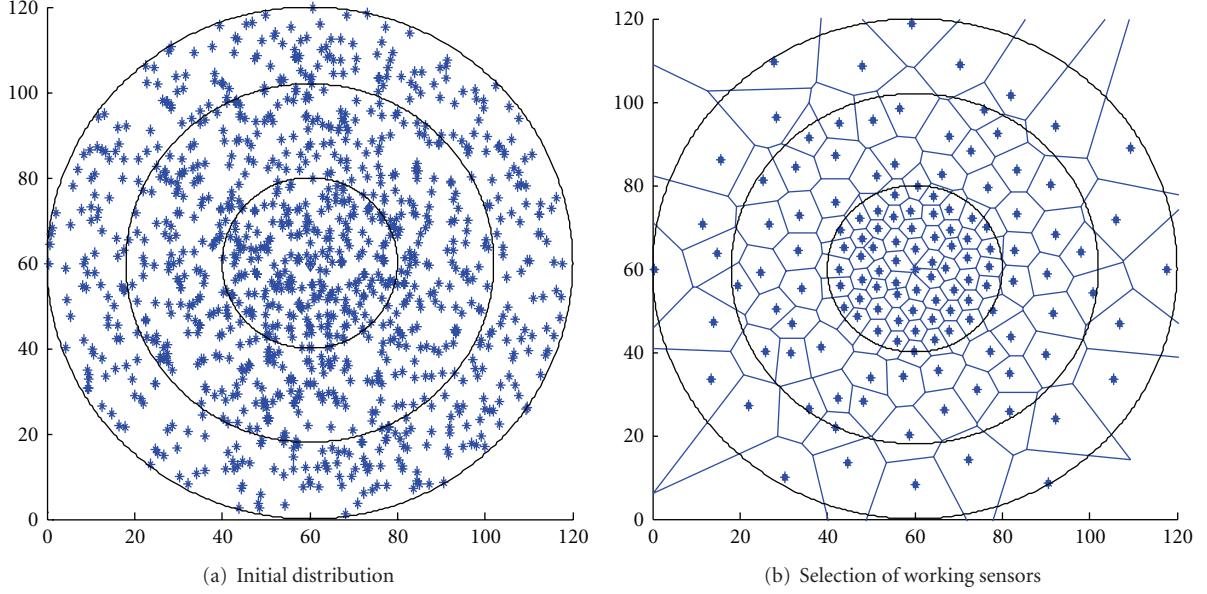


FIGURE 5: Distribution of wireless sensor network.

TABLE 1: Simulation parameters.

Parameters	value
Initial energy of sensor node (ε)	1000 J
Energy consumed to send one bit of data (e_1)	0.5/10 ³ J
Energy consumed to receive one bit of data (e_2)	0.25/10 ³ J
The basic unit of data length (L)	1000 Bit
Monitoring area radius (R)	60
The number of nodes deployed	1000–3000
Communication radius (R_c)	20
Sensing radius (R_s)	10
Recombination rate	0.9
Working round T	1000 s
Monitoring cycle ΔT	3 s
The threshold of working node energy (used to maintain the energy level of dormancy) (P_t)	900 s
Time of the initial node from timer (T_d)	10 ms
Time of the initial node selection process (T_s)	1 s
Time of receiving the power-on message from the first noninitial node and the working area of node not covered by its neighbours (T_{c3})	200 ms
Power-on message sending time	6.8 ms
Time of calculation T_{c1} and T_{c2} required constant C	10/rs ²

In order to verify that the working sensors selected by our algorithm can balance energy consumption, the energy depletion of this working set in one round is investigated especially. In our simulation, the working round is set as 1000 s, and the monitoring cycle is 3 s. The energy depletion of those nodes in one working round is shown in Figure 6.

From Figure 6, we can see that although the nodes in corona C_1 and C_2 behave as both data originator and router,

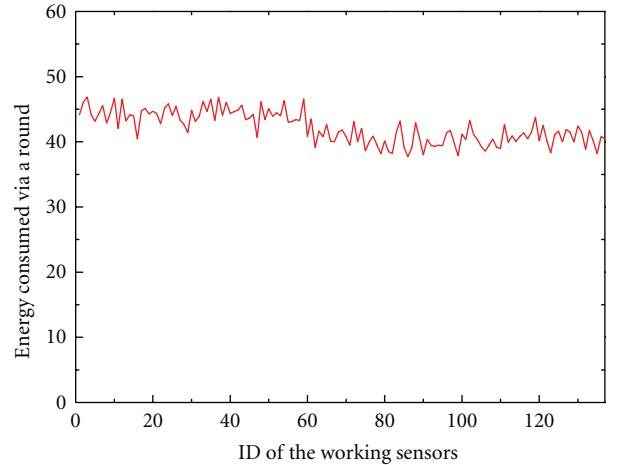


FIGURE 6: Energy consumption of the working nodes in one round.

the energy consumption of the whole working set is almost equal, thus to enhance the power efficiency of sensors in outer coronas. This is mainly because the inner sensors' sensing pixels are much smaller than those of the outer sensors by adopting nonuniform sensor distribution and pixel-based transmission mechanism.

Figure 7 shows the relationship between the total energy left and working rounds. From Figure 7, we can see that the total energy left with working rounds has an approximate linear relation. When the network runs to 150 working rounds, the remaining energy is 102300. Continuing to run algorithm, we can see that the energy attenuation with the working cycles becomes more flat. That is because the remaining survival nodes can no longer establish communication with the sink node, and the energy consumption is mainly caused by network sensing with little data forwarding. Although

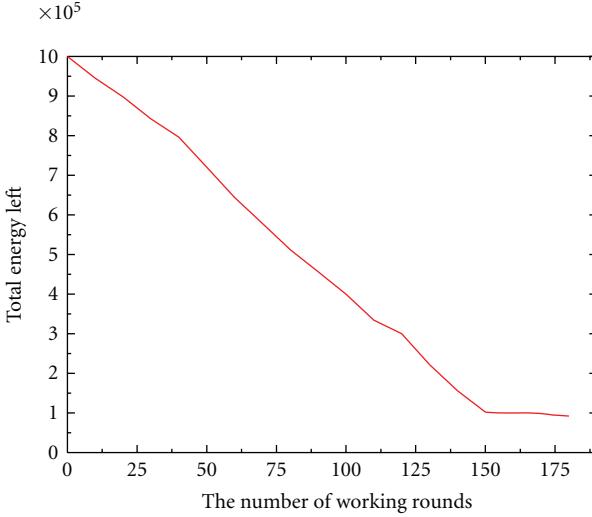


FIGURE 7: Changes of energy left in different working rounds.

there is about 10% of the residual energy, due to the uneven distribution of these final survival nodes, they cannot meet the needs of the network coverage and connectivity any more.

We further compare the performance of EBDC with OGDC, PCP, and the nonuniform distribution in [13]. In those series of simulations, we vary the deployed nodes density from 1000 to 3000 nodes in the circular area with radius 60. The round length is set as 1000 s and the monitoring cycle is 3 s.

Figure 8 shows the network lifetime comparison with different node deployments. Although OGDC focuses on how to select the optimal cover set, it does not consider the imbalance consumption of energy near the sink. This mainly causes the nodes near the sink node to forward data more frequently and finally gets a much shorter network lifetime. As PCP uses a probabilistic detection model, it needs fewer active sensors to cover the target area completely and thus has more working rounds than OGDC. However, the problem of imbalanced energy depletion is not solved effectively in PCP. Although literature [13] adopts a nonuniform node distribution, the energy imbalance still exists because of constant data acquisition, which inevitably leads inner sensors consume more energy than outer sensors. As the pixel-based transmission mechanism is imported in our scheme, the total transmitted messages in each round are much smaller. Our algorithm can achieve the energy balance by selecting suitable nodes to work and thus has a much longer network lifetime.

Figure 9 shows the comparison of the energy unused ratio with different deployed nodes, which refers to the ratio of the residual energy to the total energy at the end of the network lifetime. With the increase of deployed nodes in the network, the energy unused ratio shows a downward tendency using our algorithm. This is because that the energy consumption of each node selected by our algorithm in each working cycle is almost equal. The energy unused is mainly caused by the initially uneven distribution of nodes. The

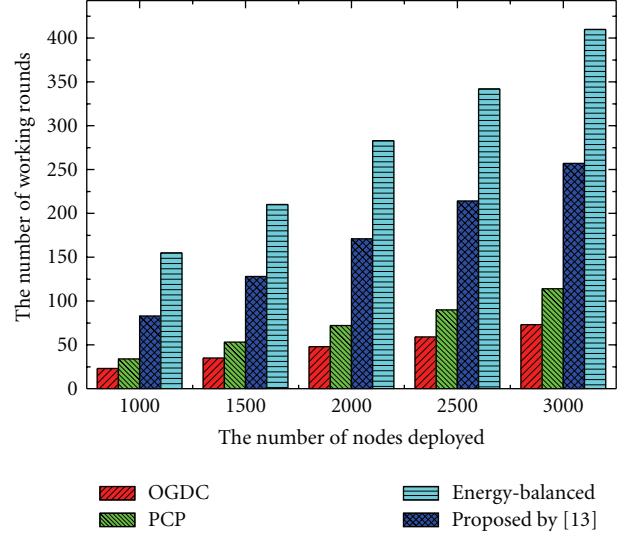


FIGURE 8: The number of nodes deployed versus working cycles.

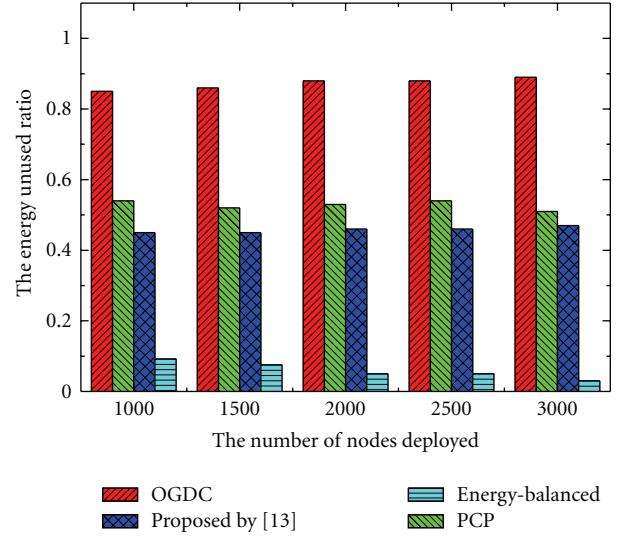


FIGURE 9: Comparison of energy unused ratio with different deployed nodes.

more nodes deployed, the more chance can be got for the survival nodes connecting to the sink and thus reduce the energy unused ratio. Since OGDC and PCP adopt a uniform node selection strategy and do not consider the phenomenon of energy imbalance consumption, both of them have a large energy unused ratio. Actually, as the nonuniform distribution strategy does not take the pixel-based data transmission mechanism, the energy imbalance cannot be avoided. Compared with the above methods, our density control algorithm based on energy balance has the higher-energy efficiency, which verifies the effectiveness of the algorithm.

6. Conclusion

In this paper, we have investigated the density control problem to select the energy-balanced working nodes for sensor

networks. We analyze energy attenuation in nonuniform distribution strategy theoretically and prove that when the pixel-based transmission mechanism is adopted, a full energy balance can be achieved through the rational node distribution density. Contributively, a distributed nonuniform density control algorithm with the concept of equivalent sensing radius is proposed. Simulation results show that our algorithm has a better performance than the existing algorithms and can prolong the network lifetime effectively.

In the future, as our work requires that each node knows its relative locations, we plan to investigate more deeply the impact of location on the performance of the proposed approach. We also intend to extend EBDC to the probabilistic sensing models and investigate some potential applications of EBDC such as topology control, distributed storage, and network health monitoring.

Acknowledgment

The authors would like to thank the reviewers giving valuable comments on the earlier version of this paper. This work is supported by the National Natural Science Foundation of China under Grant nos. 60903159, 61173153, 61070162, 71071028, and 70931001; China Postdoctoral Science Foundation funded project under Grant no. 20110491508; the Specialized Research Fund for the Doctoral Program of Higher Education under Grant no. 20070145017; and the Fundamental Research Funds for the Central Universities under Grant nos. N090504003 and N090504006.

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] J. Jiang and W. Dou, "A coverage-preserving density control algorithm for wireless sensor networks," in *Proceedings of the 3rd International Conference on Ad Hoc Networks and Wireless*, vol. 3158, pp. 42–55, LNCS, 2004.
- [3] S. Sljepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications*, (ICC '01), pp. 472–476, IEEE, June 2000.
- [4] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: a robust energy conserving protocol for long-lived sensor networks," in *Proceedings of the 23th IEEE International Conference on Distributed Computing Systems*, (ICDCS '03), pp. 28–37, IEEE, May 2003.
- [5] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications*, (WSNA '02), pp. 32–41, ACM, September 2002.
- [6] J. Jia, J. Chen, G. R. Chang, and Y. Y. Wen, "Efficient cover set selection in wireless sensor networks," *Acta Automatica Sinica*, vol. 34, no. 9, pp. 1157–1162, 2008.
- [7] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Adhoc and Sensor Wireless Networks*, vol. 1, no. 1, pp. 89–124, 2005.
- [8] 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, ACM, May 2006.
- [9] A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, and K. Jones, "Training a wireless sensor network," *Mobile Networks and Applications*, vol. 10, no. 1, pp. 151–168, 2005.
- [10] M. Hefeeda and H. Ahmadi, "Energy-efficient protocol for deterministic and probabilistic coverage in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 5, pp. 579–593, 2010.
- [11] J. Li and P. Mohapatra, "An analytical model for the energy hole problem in many-to-one sensor networks," in *Proceedings of the 62nd Vehicular Technology Conference*, (VTC '05), pp. 2721–2725, IEEE, September 2005.
- [12] S. Olariu and I. Stojmenovic, "Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting," in *Proceedings of the 25th IEEE International Conference on Computer Communications*, (INFOCOM '06), pp. 1–12, IEEE, April 2006.
- [13] X. B. Wu and G. H. Chen, "The energy hole problem of non-uniform node distribution in wireless sensor networks," *Chinese Journal of Computers*, vol. 31, no. 2, pp. 253–261, 2008.
- [14] M. Cardei, Y. Yang, and J. Wu, "Non-uniform sensor deployment in mobile wireless sensor networks," in *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks*, (WoWMoM '08), pp. 1–8, June 2008.
- [15] A. Savvides, C. C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the seventh annual international conference on Mobile computing and networking*, (Mobicom '01), pp. 166–179, July 2001.
- [16] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1–4, pp. 267–280, 2003.
- [17] R. Sugihara and R. K. Gupta, "Sensor localization with deterministic accuracy guarantee," in *Proceedings of the 30th International Conference on Computer Communications*, (INFOCOM '11), pp. 1772–1780, June 2011.
- [18] M. Jin, S. Xia, H. Wu, and X. Gu, "Scalable and fully distributed localization with mere connectivity," in *Proceedings of the 30th International Conference on Computer Communications*, (INFOCOM '11), pp. 3164–3172, June 2011.

Research Article

Linear Decision Fusion under the Control of Constrained PSO for WSNs

Sisi Jiang,¹ Zhiwen Zhao,^{1,2} Sheng Mou,¹ Zushun Wu,¹ and Yi Luo¹

¹College of Information Science and Technology, Beijing Normal University, Beijing 100875, China

²Key Laboratory of Inertial Technology for National Defense, Beihang University, Beijing 100191, China

Correspondence should be addressed to Zhiwen Zhao, zhaozw126@126.com

Received 31 May 2011; Revised 8 September 2011; Accepted 26 September 2011

Academic Editor: Don Sofge

Copyright © 2012 Sisi Jiang 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 major application of a distributed WSN (wireless sensor network) is to monitor a specific area for detecting some events such as disasters and enemies. In order to achieve this objective, each sensor in the network is required to collect local observations which are probably corrupted by noise, make a local decision regarding the presence or absence of an event, and then send its local decision to a fusion center. After that, the fusion center makes the final decision depending on these local decisions and a decision fusion rule, so an efficient decision fusion rule is extremely critical. It is obvious that the decision-making capability of each node is different owing to the dissimilar signal noise ratios and some other factors, so it is easy to understand that a specific sensor contribution to the global decision should be constrained by this sensor decision-making capability, and, based on this idea, we establish a novel linear decision fusion model for WSNs. Moreover, the constrained particle swarm optimization (constrained PSO) algorithm is creatively employed to control the parameters of this model in this paper and we also apply the typical penalty function to solve the constrained PSO problem. The emulation results indicate that our design is capable of achieving very high accuracy.

1. Introduction

Owing to the low cost and ease of operation, wireless sensor networks are ideal for a wide variety of applications such as environmental monitoring, smart factory instrumentation, intelligent transportation, and remote surveillance [1–3]. When a WSN is used to monitor an area, each sensor node in the network should collect local observations and send a summary (compressed or partially processed data) to the fusion center. Then the fusion center uses these summaries and a specific decision fusion rule to make the final global decision, as shown in Figure 1.

Actually, WSN issues such as node deployment, localization, energy-aware clustering, and data aggregation are always considered as optimization problems. An optimization method that requires moderate memory and computational resources and yet produces good results is desirable, especially for implementation on an individual sensor node. Bio-inspired optimization methods are computationally efficient alternatives to analytical methods. Particle swarm

optimization (PSO) [4] is a popular multidimensional optimization technique. Ease of implementation, high quality of solutions, computational efficiency, and speed of convergence are the advantages of the PSO. In literatures [5–9], PSO is used to discover the optimal WSN deployment. Static deployment is a one-time process in which solution quality is more important than fast convergence. PSO suits centralized deployment. Fast PSO variants are necessarily of dynamic deployment, so PSO limits network scalability. Literatures [10–14] apply PSO for WSN localization. Owing to energy issues, the distributed localization is desirable. Though PSO is appropriate for distributed localization, the choice is influenced by availability of memory on the nodes. Besides, PSO is also used for energy-aware clustering in literatures [15–17], where clustering is a centralized optimization carried out in a resource-rich base station and optimal clustering has a strong influence on the performance of WSNs. Recently, some researchers have begun to use PSO for WSN data aggregation. As discussed in literature [18], PSO has provided optimization in several aspects of data aggregation as follows.

- (a) In [19], Wimalajeewa and Jayaweera address the problem of optimal power allocation through constrained PSO. Their algorithm *PSO-Opt-Alloc* uses PSO to determine optimal-power allocation in the cases of both independent and correlated observations. The objective is to minimize the energy expenditure while keeping the fusion-error probability under a required threshold.
- (b) In [20], Veeramachaneni and Osadciw present a hybrid of ant-based control and PSO (*ABC-PSO*) for hierarchy and threshold management for decentralized serial sensor networks. PSO is used to determine the optimal thresholds and fusion rules for the sensors while the ant colony optimization algorithm determines the hierarchy of sensor decision communication.
- (c) In [21], Veeramachaneni and Osadciw present a binary multiobjective PSO *BMPSO* for optimal configuration for multiple sensor networks. PSO is modified to optimize two objectives: accuracy and time. The output of the algorithm is the choice of sensors, individual sensor threshold, and the optimal decision fusion rule.

As mentioned above, data fusion is a distributed repetitive process which is moderately suitable for PSO. Effective data fusion influences overall WSN performance and demands quick-convergence optimization techniques that assure high-quality solutions. Therefore, it is reasonable for us to choose PSO to control the parameters of our fusion rule.

In this paper, we design a linear decision fusion rule. In this rule, a linear equation is used to compute the integrated contribution of all the local decisions, and this equation is made up with local decision weight k_i and all local decisions. Local decision weight k_i can reflect a sensor decision-making capability, and the result of this equation is the summation of all the local sensor contributions. By comparing the result of the linear equation with a threshold λ_g , the final decision can be made in the fusion center. In order to get the smallest error probability, constrained PSO is employed to find out the optimal local decision weight k_i and the threshold λ_g .

The rest of the paper is organized as follows. Section 2 discusses the constrained-PSO algorithm briefly. Section 3 describes our linear decision fusion rule in detail. The performance of our fusion rule is evaluated in Section 4, and Section 5 concludes this paper.

2. Brief Introduction of Constrained PSO Algorithm

The particle swarm optimization (PSO) algorithm is a population-based evolutionary computation originally developed by Kennedy and Eberhart [4]. This algorithm does very well in searching throughout a highly multiple modal search space resulting in an optimized solution to an n -dimensional problem. PSO, originally introduced in terms

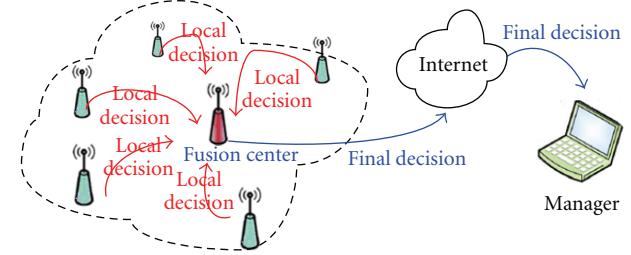


FIGURE 1: A typical structure of a decentralized decision fusion system.

of social and cognitive behavior, is widely used as a bottom-up problem solving method in engineering and computer science.

In the basic PSO algorithm, a swarm consists of m particles flying around in an n -dimensional search space. The position of the i th particle at the t th iterations is used to evaluate the particle and represent the candidate solution for the optimization problem. It can be presented as $X_i^t = [x_{i1}^t, x_{i2}^t, \dots, x_{in}^t]$, where x_{ij}^t is the position value of the i th particle with respect to the j th dimension ($j = 1, 2, \dots, n$). During the searching process, the position of a particle is affected by two factors: the best position visited by itself (p_{best}) denoted as $P_i^t = [p_{i1}^t, p_{i2}^t, \dots, p_{in}^t]$ and the best position of the swarm found so far (g_{best}) denoted as $G^t = [g_1^t, g_2^t, \dots, g_n^t]$. The new velocity (denoted as $V_i^t = [v_{i1}^t, v_{i2}^t, \dots, v_{in}^t]$) and the position of particle i at the next iteration are calculated as

$$\begin{aligned} v_{ij}^{t+1} &= w \cdot v_{ij}^t + c_1 r_1 \cdot (p_{ij}^t - x_{ij}^t) + c_2 r_2 \cdot (g_{ij}^t - x_{ij}^t), \\ x_{ij}^{t+1} &= x_{ij}^t + v_{ij}^{t+1}, \end{aligned} \quad (1)$$

where w is called the inertia parameter, c_1 and c_2 are, respectively, cognitive and social learning parameters, and r_1, r_2 are random numbers between (0,1). Relying on expressions (1), the particles can fly throughout search space toward p_{best} and g_{best} in a navigated way while still exploring new areas by the stochastic mechanism to escape from local optimal. The process for a particle to update its location is as shown in Figure 2.

Further, a constrained PSO problem can be described as follows:

$$\begin{aligned} \min_x f(x) \quad (x \in R^n), \\ g_j(x) \geq 0 \quad (j = 1, 2, \dots, q), \\ h_p(x) = 0 \quad (p = 1, 2, \dots, m), \end{aligned} \quad (2)$$

where $f(x)$ is the objective function, $g_j(x) \geq 0$ is the j th nonlinear constraint, and $h_p(x) = 0$ is the p th linear constraint. In this paper, we just discuss the situation with the linear constraint. The most common approach for solving constrained PSO problems is to take advantage of a penalty function. The constrained problem can be transformed into an unconstrained one by penalizing the constraints and building a single objective function, which in turn is minimized by using an unconstrained optimization algorithm [22–24].

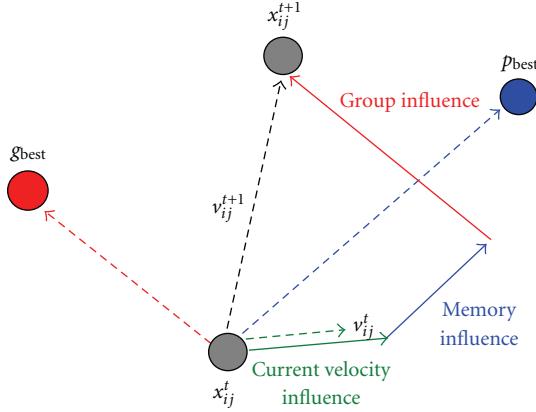


FIGURE 2: The process of updating location for each iteration in PSO.

Generally, penalty function can be classified into two main classes: stationary and nonstationary. Stationary penalty functions use fixed penalty values throughout the minimization, while in contrast, in nonstationary penalty functions, the penalty values are dynamically modified. In this paper, we just use the stationary penalty function as expression (3), which is generally defined as [25]

$$F(x) = f(x) + MH(x), \quad x \in R^n, \quad (3)$$

where $f(x)$ is the original objective function of the constrained PSO problem, M is a fixed penalty value, and $H(x)$ is a penalty factor. In this way, the constrained problem becomes unconstrained and then we can use the basic PSO to find out the smallest $F(x)$.

Depending on the knowledge about the constrained PSO, we would employ it to find out the optimal solution for the parameters of our decision fusion rule. In the next section, we would like to discuss this issue in detail.

3. The Linear Decision Fusion Rule

In this section, we plan to discuss this topic in three parts. In Section 3.1, we formulate the decision fusion problem regarding a binary hypothesis situation and the expressions of some crucial parameters are discussed, including the local error probabilities α_i , β_i , and threshold λ_i . In Section 3.2, our linear decision fusion rule is introduced and, with the expressions given by Section 3.1, the expressions of global error probabilities α_g , β_g , and total error probability R can be figured out. From Section 3.2, it can be seen that the total error probability R mainly depends on weights k_1, k_2, \dots, k_n and λ_g , so we would like to discuss how to use PSO to optimize the parameters $(k_1, k_2, \dots, k_n, \lambda_g)$ for our fusion rule in Section 3.3.

3.1. Decision Fusion Problem Formulation. We consider a binary hypothesis testing problem in an n -node distributed

wireless network. The i th sensor observation under the two hypotheses is given by

$$\begin{aligned} H_0 : z_i &= v_i, & u_i &= 0, \quad i = 1, 2, \dots, n \\ H_1 : z_i &= x_i + v_i, & u_i &= 1, \quad i = 1, 2, \dots, n, \end{aligned} \quad (4)$$

where v_i is zero-mean Gaussian observation noise with variance σ_i^2 , x_i is the signal to be detected, and u_i is the decision to be made. In this paper, we consider the detection of a constant signal (e.g., $x_i = m$). As a brief review, the problem of radar detection can be formulated as a hypothesis testing problem where the two hypotheses are

$$u_i = \begin{cases} 0, & \text{the enemy does not exist in the network,} \\ 1, & \text{the enemy exists in the network.} \end{cases} \quad (5)$$

The conditional probability density functions are $p(z_i | H_1)$ and $p(z_i | H_0)$, so the decision could be made based on the following likelihood ratio test:

$$u_i = \begin{cases} 1, & \text{when } \frac{p(z_i | H_1)}{p(z_i | H_0)} > l_0, \\ 0, & \text{when } \frac{p(z_i | H_1)}{p(z_i | H_0)} < l_0, \end{cases} \quad (6)$$

where l_0 is a threshold computed according to the Bayesian framework as follows:

$$l_0 = \frac{(C_{10} - C_{00})q}{(C_{01} - C_{11})p}. \quad (7)$$

C_{ij} is the error cost (C_{10} : we consider there is no enemy but actually the enemy exists in the network; C_{01} : we consider the enemy exists but actually there is no enemy in the network; C_{00} and C_{11} are the costs when we judge the existence of the enemy rightly), and the prior probabilities of the two hypotheses H_1 and H_0 are denoted by $P(H_1) = p$ and $P(H_0) = q$, respectively. In real applications, the value of C_{10} and C_{01} depends on what kinds of error would cause more serious effect. Taking the radar detection mentioned above as an example, in the real world the error that the radar misses an enemy is much more serious than the error that the radar considers something else as an enemy, so in this example $C_{10} > C_{01}$. In this paper, we set $C_{10} = 0.8$, $C_{01} = 0.2$, and $C_{00} = C_{11} = 0$, so expression (7) becomes

$$l_0 = \frac{C_{10}q}{C_{01}p}. \quad (8)$$

The values of prior probabilities p and q depend on prior experiences, so we assume that they can be known before applying this decision fusion rule. Let us consider the detection of a constant signal $x_i = m$, and the noise v_i mentioned above is zero-mean Gaussian observation noise with variance σ_i^2 , so, for all i , the i th sensor observation obeys Gaussian distribution as

$$\begin{aligned} H_0 : z_i &\sim N(0, \sigma_i^2), \quad i = 1, 2, \dots, n, \\ H_1 : z_i &\sim N(m, \sigma_i^2), \quad i = 1, 2, \dots, n. \end{aligned} \quad (9)$$

Consequently, the conditional probability density functions $p(z_i | H_1)$ and $p(z_i | H_0)$ can be calculated as

$$\begin{aligned} p(z_i | H_1) &= \frac{1}{\sqrt{2\pi}\sigma_i} \exp \left[-\frac{(x - m)^2}{2\sigma_i^2} \right], \\ p(z_i | H_0) &= \frac{1}{\sqrt{2\pi}\sigma_i} \exp \left[-\frac{x^2}{2\sigma_i^2} \right], \end{aligned} \quad (10)$$

and then, with (6) and (10), the decision rule of the i th sensor can be formulated as

$$u_i = \begin{cases} 1, & \text{when } x_i > \frac{m}{2} + \sigma_i^2 \ln \frac{C_{10}q}{C_{01}p}, \\ 0, & \text{when } x_i > \frac{m}{2} + \sigma_i^2 \ln \frac{C_{10}q}{C_{01}p}. \end{cases} \quad (11)$$

According to (11), the threshold of the i th sensor is

$$\lambda_i = \frac{m}{2} + \sigma_i^2 \ln \frac{C_{10}q}{C_{01}p}. \quad (12)$$

From (12), we can see that the threshold of each sensor is different owing to the different σ_i which represents the noise character of a sensor. Accordingly, taking advantage of (10), (12), and (13), the local error probabilities α_i and β_i can be obtained as follows:

$$\begin{aligned} \alpha_i &= P(u_i = 1 | H_0) = \int_{\lambda_i}^{\infty} p(z_i | H_0) dz_i, \\ \beta_i &= P(u_i = 0 | H_1) = \int_0^{\lambda_i} p(z_i | H_1) dz_i. \end{aligned} \quad (13)$$

3.2. Our Linear Decision Fusion Rule. In this paper, we design a novel decision fusion model and in this model, a linear equation is used to denote the integrated contribution of all the local decisions as

$$z_g = \sum_{i=1}^n k_i u_i, \quad (14)$$

where u_i is the local decision of the i th sensor and k_i is the corresponding weight which is able to indicate the i th sensor decision-making capability. In order to understand this model better, we take a group of workers who are working together to check an event for example. Firstly, all the workers are required to try their best to draw a conclusion for this event, and then they send their decision to their leader. The leader makes a final decision depending on all the conclusions given by the workers. During this decision-making process, the decisions made by the more capable workers would be paid more attention by the leader. In this example, the workers correspond to the sensors in the mobile sensor networks, the leader corresponds to the fusion center, and the ability of the workers corresponds to the weight k_i . When a sensor decision-making capability is better, its weight could be greater. In other words, if a sensor can make a decision more accurately, its contribution to z_g is considered to be bigger, so it is reasonable to use $z_g = \sum_{i=1}^n k_i u_i$ to denote

the integrated contribution of local decisions in the fusion center. Then, z_g would be compared with a threshold λ_g , and the final decision can be made according to the following inequality expression:

$$u_g = \begin{cases} 1, & \text{when } z_g > \lambda_g, \\ 0, & \text{when } z_g < \lambda_g. \end{cases} \quad (15)$$

Suppose all the local decisions are independent of each other; then we can get $(P(u_1 u_2 \cdots u_n | H_0)) = \prod_{i=1}^n P(u_i | H_0)$ and $(P(u_1 u_2 \cdots u_n | H_1)) = \prod_{i=1}^n P(u_i | H_1)$. Besides, the error probabilities α_g and β_g take all the combinations of the local decisions (i.e., $u_1 u_2 \cdots u_n = 00 \cdots 0 \sim 11 \cdots 1$) into account. Therefore, the two kinds of errors that the fusion center may make can be formulated as follows:

$$\begin{aligned} \alpha_g &= \sum_{u_1 u_2 \cdots u_n = 00 \cdots 0}^{11 \cdots 1} P_{u_1 u_2 \cdots u_n}(u_g = 1 | H_0) \\ &= \sum_{u_1 u_2 \cdots u_n = 00 \cdots 0}^{11 \cdots 1} P_{u_1 u_2 \cdots u_n} \left(\sum_{i=1}^n k_i u_i > \lambda_g | H_0 \right) \\ &= \sum_{u_1 u_2 \cdots u_n = 00 \cdots 0}^{11 \cdots 1} \left(P(u_1 u_2 \cdots u_n | H_0) \right. \\ &\quad \left. \times P \left(\sum_{i=1}^n k_i u_i > \lambda_g \right) \right) \\ &= \sum_{u_1 u_2 \cdots u_n = 00 \cdots 0}^{11 \cdots 1} \left(\prod_{i=1}^n P(u_i | H_0) \times P \left(\sum_{i=1}^n k_i u_i > \lambda_g \right) \right), \\ \beta_g &= \sum_{u_1 u_2 \cdots u_n = 00 \cdots 0}^{11 \cdots 1} P_{u_1 u_2 \cdots u_n}(u_g = 0 | H_1) \\ &= \sum_{u_1 u_2 \cdots u_n = 00 \cdots 0}^{11 \cdots 1} P_{u_1 u_2 \cdots u_n} \left(\sum_{i=1}^n k_i u_i < \lambda_g | H_1 \right) \\ &= \sum_{u_1 u_2 \cdots u_n = 00 \cdots 0}^{11 \cdots 1} \left(P(u_1 u_2 \cdots u_n | H_1) \right. \\ &\quad \left. \times P \left(\sum_{i=1}^n k_i u_i < \lambda_g \right) \right) \\ &= \sum_{u_1 u_2 \cdots u_n = 00 \cdots 0}^{11 \cdots 1} \left(\prod_{i=1}^n P(u_i | H_1) \times P \left(\sum_{i=1}^n k_i u_i < \lambda_g \right) \right), \end{aligned} \quad (16)$$

where

$$\begin{aligned} P(u_i = 0 | H_0) &= q - \alpha_i, \\ P(u_i = 1 | H_0) &= \alpha_i, \\ P(u_i = 0 | H_1) &= \beta_i, \\ P(u_i = 1 | H_1) &= p - \beta_i. \end{aligned} \quad (17)$$

According to (13), (16), and (17), the total error cost can be calculated as

$$R = \frac{C_{10}q}{C_{01}p} \alpha_g + \beta_g. \quad (18)$$

In order to understand this model better, we take the situation $n = 2$ for example. Suppose there are only two sensors in the network and their local decisions are u_1 and u_2 , respectively. Then according to the linear decision fusion model, the integrated contribution of the two local decisions can be figured out as $z_g = k_1 u_1 + k_2 u_2$.

It is obvious that there are four combinations of u_1 , u_2 , and z_g , as shown in Table 1.

Then referring to expression (16), we can have

$$\begin{aligned} \alpha_g &= P(u_g = 1 | H_0) \\ &= P(u_1 = 0 | H_0)P(u_2 = 0 | H_0)P(0 > \lambda_g) \\ &\quad + P(u_1 = 1 | H_0)P(u_2 = 0 | H_0)P(k_1 > \lambda_g) \\ &\quad + P(u_1 = 0 | H_0)P(u_2 = 1 | H_0)P(k_2 > \lambda_g) \\ &\quad + P(u_1 = 1 | H_0)P(u_2 = 1 | H_0)P(k_1 + k_2 > \lambda_g), \end{aligned}$$

$$\begin{aligned} P(u_1 = 0 | H_0) &= q - \alpha_1, \\ P(u_1 = 1 | H_0) &= \alpha_1, \\ P(u_2 = 0 | H_0) &= q - \alpha_2, \\ P(u_2 = 1 | H_0) &= \alpha_2, \\ \beta_g &= P(u_g = 0 | H_1) \end{aligned} \quad (19)$$

$$\begin{aligned} &= P(u_1 = 0 | H_1)P(u_2 = 0 | H_1)P(0 < \lambda_g) \\ &\quad + P(u_1 = 1 | H_1)P(u_2 = 0 | H_1)P(k_1 < \lambda_g) \\ &\quad + P(u_1 = 0 | H_1)P(u_2 = 1 | H_1)P(k_2 < \lambda_g) \\ &\quad + P(u_1 = 1 | H_1)P(u_2 = 1 | H_1)P(k_1 + k_2 < \lambda_g), \end{aligned}$$

$$\begin{aligned} P(u_1 = 0 | H_1) &= \beta_1, \\ P(u_1 = 1 | H_1) &= p - \beta_1, \\ P(u_2 = 0 | H_1) &= \beta_2, \\ P(u_2 = 1 | H_1) &= p - \beta_2, \end{aligned}$$

and then we can achieve

$$R = \frac{C_{10}q}{C_{01}p} \alpha_g + \beta_g. \quad (20)$$

Here, it can be seen that the value of the total error probability R depends upon k_1 , k_2 , and λ_g , so if the optimal k_1 , k_2 , and λ_g can be found, the smallest error probability R can be achieved. In the following content, let us discuss how to get the optimal weights k_i and threshold λ_g with the help of constrained PSO.

TABLE 1: The combinations of u_1 , u_2 , and z_g .

u_2	u_1	z_g
0	0	0
0	1	k_1
1	0	k_2
1	1	$k_1 + k_2$

TABLE 2: PSO terminology and corresponding parameters of linear decision fusion rule.

PSO terminology	Linear decision fusion rule
Location	$(k_1, k_2, \dots, k_n, \lambda_g)$
Fitness	$R' = (C_{10}q/C_{01}p)\alpha_g + \beta_g + M(\sum_{i=1}^n k_i - 1)^2$ ($k_i \geq 0$)
p_{best}	The location in parameter space of the smallest R' returned for a specific particle
g_{best}	The location in parameter space of the smallest R' returned in the entire swarm
V_{\max}	The maximum allowed velocity range in a given direction $[0, 1]$
X_{\max}	The maximum allowed range in a given direction $[0, 1]$

3.3. Using Constrained PSO to Control the Linear Decision Fusion Rule. It is obvious that if all the local decisions are H_1 ($u_i = 1, i = 1, 2, \dots, n$), the final decision must be H_1 ($z_g = 1$), so we can consider that $\sum_{i=1}^n k_i = 1$ ($k_i \geq 0$), and then, it becomes a constrained PSO problem which can be denoted as follows:

$$\begin{aligned} R &= \frac{C_{10}q}{C_{01}p} \alpha_g + \beta_g, \\ \sum_{i=1}^n k_i &= 1 \quad (k_i \geq 0). \end{aligned} \quad (21)$$

Here, $R = (C_{10}q/C_{01}p)\alpha_g + \beta_g$ is the initial objective function and $\sum_{i=1}^n k_i = 1$ ($k_i \geq 0$) is the linear constraint. The penalty function approach is used to solve this problem. According to formulation (3) the penalty function can be established as

$$R' = \frac{C_{10}q}{C_{01}p} \alpha_g + \beta_g + M \left(\sum_{i=1}^n k_i - 1 \right)^2, \quad (22)$$

where M is the fixed penalty value and $(\sum_{i=1}^n k_i - 1)^2$ is the penalty factor. Then we should try to find out the smallest R' taking advantage of the basic PSO.

Basing on the analysis above, we can map from the basic PSO to the linear decision fusion rule, as shown in Table 2.

After that, the PSO is carried out as follows.

- (1) Firstly, the locations of all the particles are initialized and the current location is considered as p_{best} .
- (2) Secondly, the velocity and the location of each node are updated in accordance with expressions (1) and (3).

TABLE 3: The simulation results with different inertia parameter w when the sensor number is 10.

Inputs ($c_1 = c_2 = 2$)	Outputs	
Inertia parameter w	Total error probability R	Time consumed for convergence (iterations)
$w = 0.5$	0.00181079	447
$w = w_{\max} - ((w_{\max} - w_{\min})/T_{\max}) \times t = 0.9 - (0.5/1500) \times t$	0.00120768	123

TABLE 4: The simulation results with different learning parameters c_1 and c_2 when the sensor number is 10.

Inputs ($w = w_{\max} - ((w_{\max} - w_{\min})/T_{\max}) \times t$)	Outputs	
c_1 and c_2	Total error probability R	The number of iteration
$c_1 = c_2 = 2$	0.00120768	123
$c_1 = 0, c_2 = 2$	0.00274277	150
$c_1 = 2, c_2 = 0$	0.00121506	Cannot converge

TABLE 5: The inputs and outputs of the simulation.

Inputs (mean = 0, $M = 10, P = 0.8, q = 0.2$)	Outputs			
n	Differences	(k_1, k_2, \dots, k_n)	λ_g	$R = R' - M(\sum_{i=1}^n k_i - m)^2 k_i$
5	(43.86, 52.63, 26.2, 39.39, 80.08)	(0.102567, 0.463625, 0.0806701, 0.019778, 0.333359)	0.704058	0.0342967
10	(43.86, 52.63, 26.2, 39.39, 80.08, 47.16, 45.09, 34.65, 67.89, 54.78)	(0.191843, 0.253623, 0.0650721, 0.00822591, 0, 0.158283, 0.0213777, 0.251185, 0.00907494, 0.0413086)	0.745094	0.00116502
15	(43.86, 52.63, 26.2, 39.39, 80.08, 47.16, 45.09, 34.65, 67.89, 54.78, 34.76, 56.34, 27.89, 47.89, 38.97)	(0.0244901, 0.157784, 0.368528, 0.10523, 0.0614695, 0.0173535, 0.0114619, 0, 0.00495967, 0.048092, 0.00231704, 0, 0.0210975, 0.127067, 0.0500054)	0.334601	0.0000403

- (3) Thirdly, the new fitness is computed and the new p_{best} and g_{best} are generated.
- (4) Finally, if the number of iterations increases to T_{\max} , then stop; otherwise, proceed back to step two. In this way, the optimal $(k_1, k_2, \dots, k_n, \lambda_g)$ is discovered and g_{best} is the smallest R' . Therefore, the smallest total error probability can be computed as $R = R' - M(\sum_{i=1}^n k_i - 1)^2$.

4. Performance Evaluation

In this section, we would like to demonstrate the performance of our decision fusion rule. On the platform Matlab, the basic PSO algorithm is realized and the objective function is set as $R' = (C_{10}q/C_{01}p)\alpha_g + \beta_g + M(\sum_{i=1}^n k_i - 1)^2$. In order to obtain the smallest R' , a swarm consists of 25 particles flying around in an $(n+1)$ -dimensional search space, where n is the number of network sensors. In the end of searching process, the g_{best} should be the smallest R' , and then, with R' , the smallest error probability R for our fusion rule can be obtained. Consequently, the corresponding k_i and λ_g are the optimal parameters of the fusion rule.

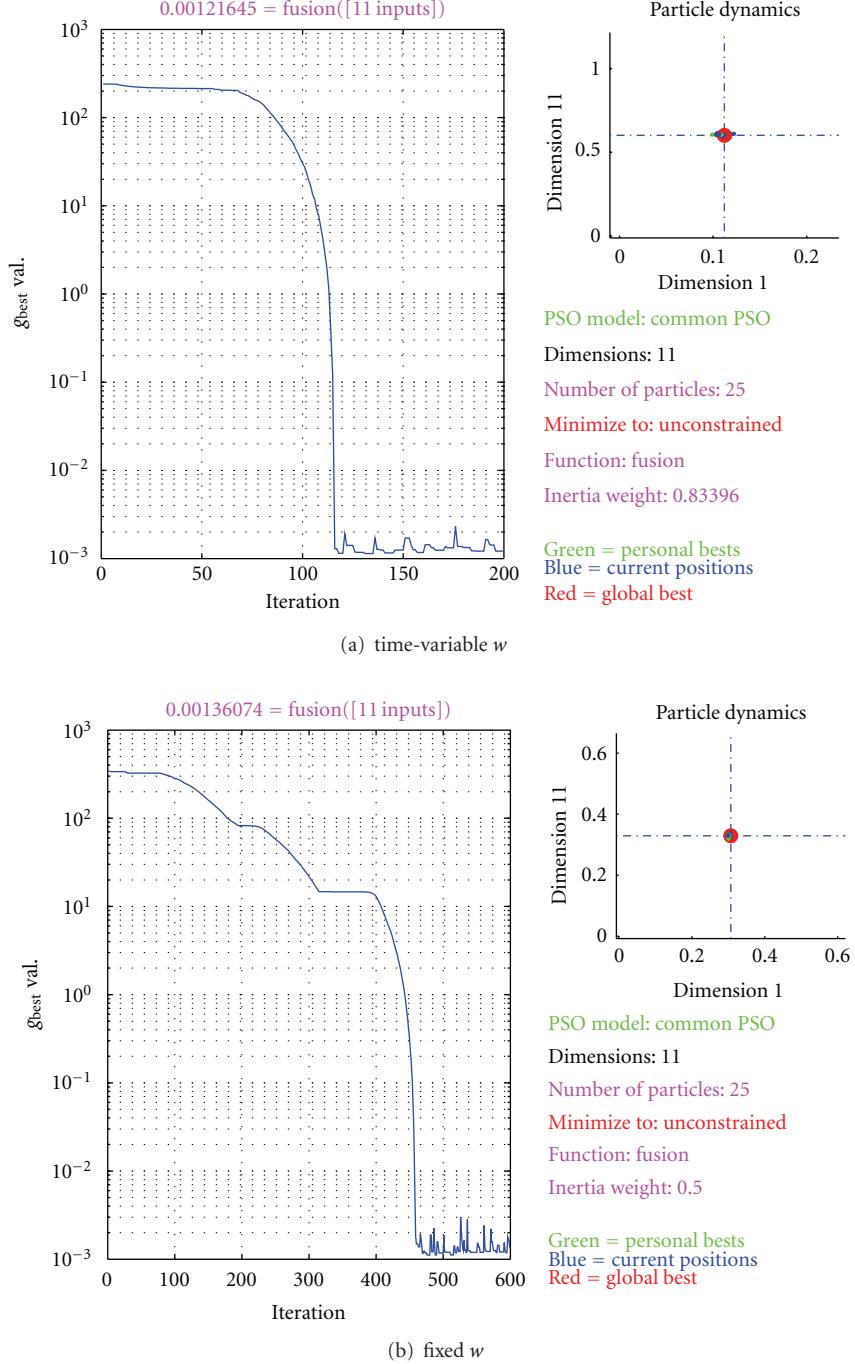
Firstly, whether the parameters of PSO can affect the performance of our decision fusion rule is demonstrated in

Section 4.1. Then, in Section 4.2, let us show that what would happen to our experimental results as the number of sensors increases.

4.1. Choosing PSO Parameters w , c_1 , and c_2 . It can be seen from expression (1) that the optimal performance of PSO depends on the values of parameters w , c_1 , and c_2 . The inertia parameter w controls the momentum of each particle. A larger w can increase the global exploration ability while a smaller w tends to improve the local exploration ability of particles. The cognitive learning parameter c_1 limits the contribution of a specific particle p_{best} while the social learning parameter c_2 limits the contribution of the entire swarm g_{best} .

At first, let us discuss the performance of our fusion rule when inertia parameter w is fixed or changes along with time. In the first case, we set $w = 0.5$ while, in the second case, we set $w = w_{\max} - ((w_{\max} - w_{\min})/T_{\max}) \times t$, where t and T_{\max} are, respectively, the current and maximum number of iterations and w_{\max} and w_{\min} are, respectively, the upper and lower bounds of w . In this paper, we set $w_{\max} = 0.9$, $w_{\min} = 0.4$, and $T_{\max} = 1500$. The inputs and outputs of this simulation are as shown in Table 3.

From Table 3 and Figure 3, we can see that when the inertia parameter w changes along with time, the total error

FIGURE 3: The simulation results with time-variable and fixed inertia parameter w .

probability R is lower and the time consumed for convergence is shorter than those when w is fixed. The reason is that if we choose the time-variable inertia parameter, particles can have good global exploration ability at the beginning phase of the searching process and good local exploration ability as the current number of iterations increases. Thereby, we decide to choose the time-variable inertia parameter w .

After that, let us discuss the learning parameters c_1 and c_2 . If $c_1 = 0$ or $c_2 = 0$, it means that we do not take the cognitive

learning process or the society learning process into account. Under this situation, the simulation results are as shown in Table 4.

From Table 4 and Figure 4, it can be seen that when the cognitive learning parameter c_1 is not taken into account, PSO still can get converged, but the total error probability R becomes larger and time consumed for convergence is longer comparing with the situation when $c_1 \neq 0$. When we do not consider the society learning parameter c_2 , PSO cannot get

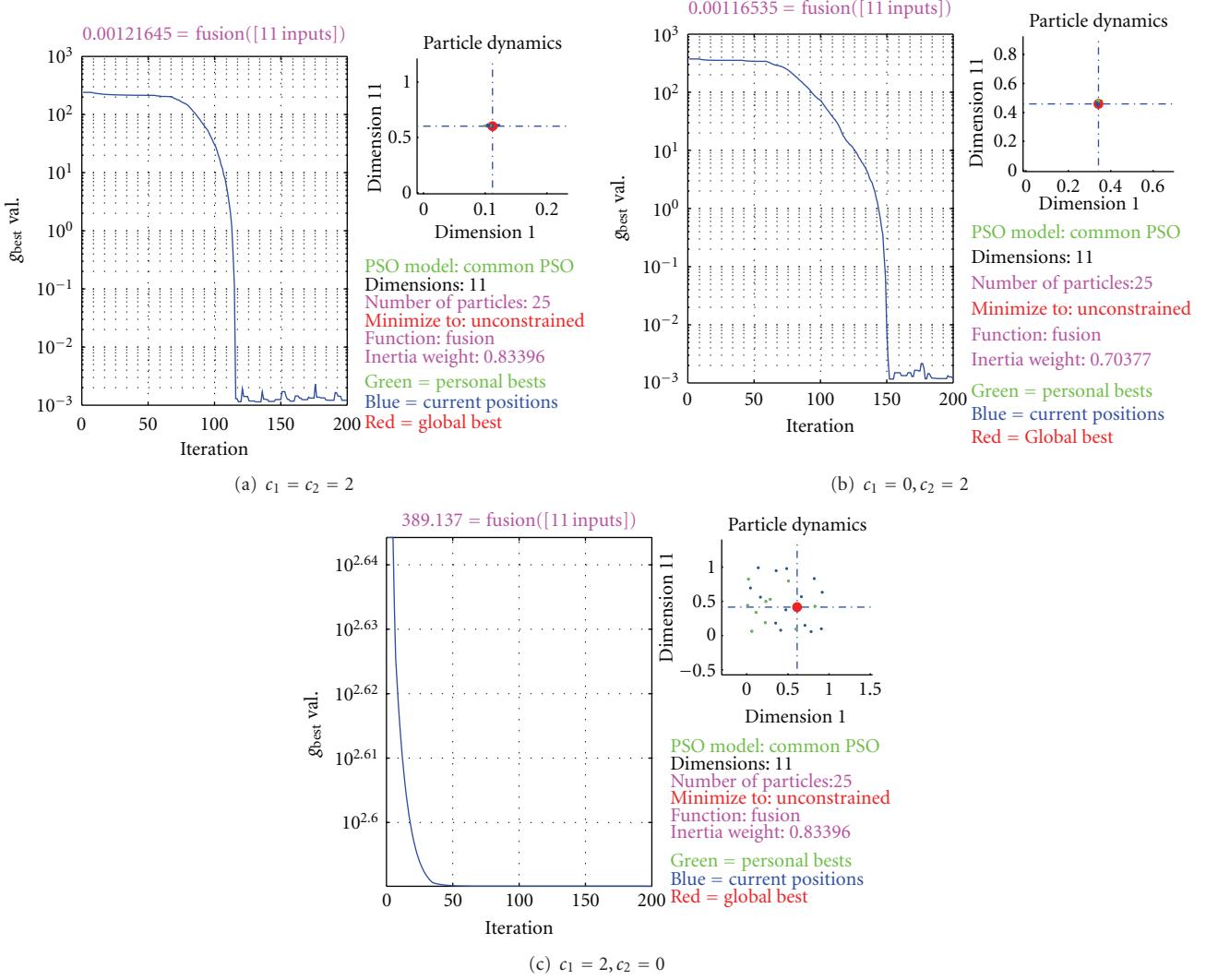


FIGURE 4: The experimental results with different learning parameters.

converged at all. The reason is that each particle only has the knowledge about itself, so the swarm can never evolve to the same direction.

In sum, from the analysis above, we can see that the time-variable inertia parameter w is better than the fixed one and the situation that both learning parameters c_1 and c_2 are taken into account is better than the situation when $c_1 = 0$ or $c_2 = 0$. Therefore, in the following simulation, the parameters are set as

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{T_{\max}} \times t, \quad (23)$$

$$c_1 = c_2 = 2.$$

4.2. The Performance of Linear Fusion Rule with Different Sensor Numbers. Here, the number of sensors in the network is set as $n = 5$, $n = 10$ and $n = 15$, respectively. According to the experimental results, it can be seen that our decision fusion rule can indeed get very high accuracy. The inputs

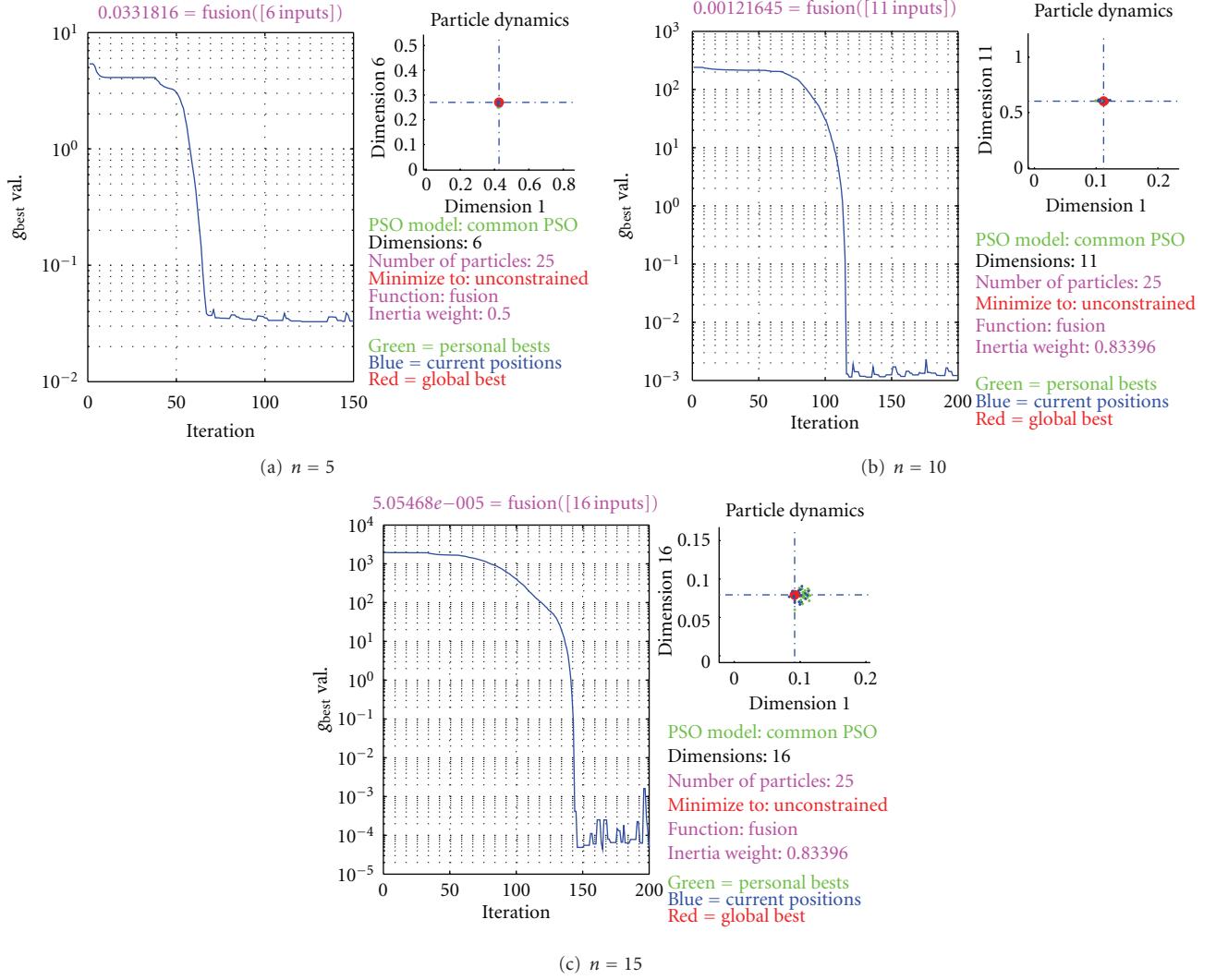
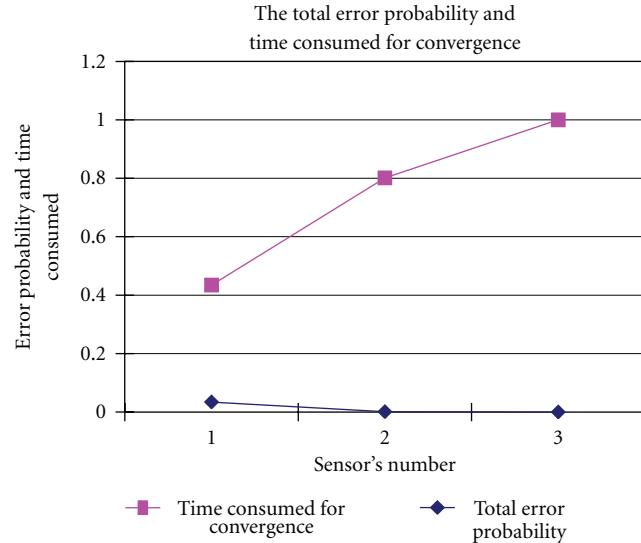
and outputs of our simulation are as shown in Table 5 and Figure 5.

When $n = 5$, the decision fusion rule becomes

$$u_g = \begin{cases} 1, & \text{when } 0.102567u_1 + 0.463625u_2 + 0.0806701u_3 \\ & + 0.019778u_4 + 0.333359u_5 > 0.704058, \\ 0, & \text{when } 0.102567u_1 + 0.463625u_2 + 0.0806701u_3 \\ & + 0.019778u_4 + 0.333359u_5 < 0.704058, \end{cases} \quad (24)$$

and, with these parameters, we can get the total error probability $R = 0.0342967$ which is very small. After about more than 50 iterations, the PSO has already got converged, so the time consumed is very short.

Similarly, from Table 5 and Figure 5(b), when $n = 10$, we can see that the total error probability becomes $R = 0.00116502$ and after about 120 iterations the PSO gets

FIGURE 5: The experimental results under different sensor numbers n .FIGURE 6: The total error probability R and time consumed for convergence change along with sensor number.

converged. When $n = 15$, $R = 0.0000403$ which is close to zero and after 150 iterations the PSO gets converged.

As shown in Figure 6, along with the increase of n , the error cost becomes smaller and smaller. This phenomenon is easy to understand because the decision result is more precise when there are more sensors working in the network. However, as n increases, the time consumed for PSO to find the optimized parameters becomes longer, so when this fusion rule is used we should better get a balance between accuracy and time.

In brief, this simulation indicates that our linear fusion rule is capable of getting really small total error probability under the control of the constrained PSO, also showing that PSO does very well for the parameter optimization of the linear decision fusion rule.

5. Conclusion

In this paper, we present a linear decision fusion model and propose a way of controlling the parameters of the model taking the advantage of the constrained PSO. In the model, the integrated contribution of local decisions is computed with a linear equation which is made up with local decision weights and local decisions, and then the integrated contribution is compared with a threshold in the fusion center. After that, according to the comparison results, the final decision can be made. Furthermore, the constrained PSO is creatively employed to discover the weights and the threshold. The simulation results show that our linear decision rule and the way of parameter optimization are efficient to get very high accuracy.

References

- [1] J.-J. Xiao, S. Cui, Z.-Q. Luo, and A. J. Goldsmith, "Joint estimation in sensor networks under energy constraints," *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON '04)*, pp. 264–271, October 2004.
- [2] L. Snidaro, R. Niu, P. K. Varshney, and G. L. Foresti, "Sensor fusion for video surveillance," in *Proceedings of the 7th International Conference on Information Fusion (FUSION '04)*, pp. 564–571, Stockholm, Sweden, June 2004.
- [3] A. Tiwari, F. L. Lewis, and S. S. Ge, "Wireless sensor network for machine condition based maintenance," *Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision (ICARCV '04)*, Amari Watergate Bangkok, vol. 1, pp. 461–467, December 2004.
- [4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, December 1995.
- [5] N. A. B. A. Aziz, A. W. Mohammed, and B. S. Daya Sagar, "Particle swarm optimization and Voronoi diagram for wireless sensor networks coverage optimization," in *Proceedings of the International Conference on Intelligent and Advanced Systems (ICIAS '07)*, pp. 961–965, November 2007.
- [6] J. Hu, J. Song, M. Zhang, and X. Kang, "Topology optimization for urban traffic sensor network," *Tsinghua Science and Technology*, vol. 13, no. 2, pp. 229–236, 2008.
- [7] J. Li, K. Li, and W. Zhu, "Improving sensing coverage of wireless sensor networks by employing mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO' 07)*, pp. 899–903, Sanya, China, December 2007.
- [8] X. Wang, S. Wang, and J. J. Ma, "An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment," *Sensors*, vol. 7, no. 3, pp. 354–370, 2007.
- [9] T. P. Hong and G. N. Shiu, "Allocating multiple base stations under general power consumption by the particle swarm optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '07)*, pp. 23–28, April 2007.
- [10] A. Gopakumar and L. Jacob, "Localization in wireless sensor networks using particle swarm optimization," in *Proceedings of the IET Conference on Wireless, Mobile and Multimedia Networks*, pp. 227–230, January 2008.
- [11] R. V. Kulkarni, G. K. Venayagamoorthy, and M. X. Cheng, "Bio-inspired node localization in wireless sensor networks," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '09)*, pp. 205–210, San Antonio, Tex, USA, October 2009.
- [12] K. S. Low, H. A. Nguyen, and H. Guo, "A particle swarm optimization approach for the localization of a wireless sensor network," in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE '08)*, pp. 1820–1825, July 2008.
- [13] H. Guo, K.-S. Low, and H.-A. Nguyen, "Optimizing the localization of a wireless sensor network in real time based on a low-cost microcontroller," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 3, pp. 741–749, 2011.
- [14] K. S. Low, H. A. Nguyen, and H. Guo, "Optimization of sensor node locations in a wireless sensor network," in *Proceedings of the 4th International Conference on Natural Computation (ICNC '08)*, pp. 286–290, October 2008.
- [15] S. M. Guru, S. K. Halgamuge, and S. Fernando, "Particle swarm optimisers for cluster formation in wireless sensor networks," in *Proceedings of the Intelligent Sensors, Sensor Networks and Information Processing Conference*, S. K. Halgamuge, Ed., pp. 319–324, December 2005.
- [16] N. M. A. Latiff, C. C. Tsimenidis, and B. S. Sharif, "Energy-aware clustering for wireless sensor networks using particle swarm optimization," in *Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '07)*, pp. 1–5, September 2007.
- [17] X. Cao, H. Zhang, J. Shi, and G. Cui, "Cluster heads election analysis for multi-hop wireless sensor networks based on weighted graph and particle swarm optimization," in *Proceedings of the 4th International Conference on Natural Computation (ICNC '08)*, vol. 7, pp. 599–603, October 2008.
- [18] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: a brief survey," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 41, no. 2, pp. 262–267, 2011.
- [19] T. Wimalajeewa and S. K. Jayaweera, "Optimal power scheduling for correlated data fusion in wireless sensor networks via constrained PSO," *IEEE Transactions on Wireless Communications*, vol. 7, no. 9, pp. 3608–3618, 2008.
- [20] K. Veeramachaneni and L. Osadciw, "Swarm intelligence based optimization and control of decentralized serial sensor networks," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '08)*, pp. 1–8, September 2008.
- [21] K. Veeramachaneni and L. A. Osadciw, "Dynamic sensor management using multi objective particle swarm optimizer," in *Multisensor, Multisource Information Fusion: Architectures,*

- Algorithms, and Applications*, vol. 5434 of *Proceedings of SPIE*, pp. 1–12, Orlando, Fla, USA, April 2004.
- [22] C. A. Floudas and P. M. Pardalos, “A collection of test problems for constrained global optimization algorithms,” in *Lecture Notes in Computer Science*, vol. 455, p. 180, Springer, Berlin, Germany, 1987.
 - [23] D. M. Himmelblau, *Applied Nonlinear Programming*, McGraw-Hill, New York, NY, USA, 1972.
 - [24] S. S. Rao, *Optimization: Theory and Applications*, Wiley Eastern Limited, 1977.
 - [25] J.-M. Yang, Y.-P. Chen, J.-T. Horng, and C.-Y. Kao, “Applying family competition to evolution strategies for constrained optimization,” in *Lecture Notes in Computer Science*, vol. 1213, pp. 201–211, Springer, Berlin, Germany, 1997.

Research Article

Capacity-Preserved, Energy-Enhanced Hybrid Topology Management Scheme in Wireless Sensor Networks for Hazardous Applications

A. Jawahar, S. Radha, and R. Sharath Kumar

Electronics and Communication Engineering, Sri Sivasubramaniya Nadar College of Engineering, Rajiv Gandhi Salai, Kalavakkam, Tamil Nadu, Chennai 603110, India

Correspondence should be addressed to A. Jawahar, jawahara@ssn.edu.in

Received 30 May 2011; Revised 6 September 2011; Accepted 6 September 2011

Academic Editor: Don Sofge

Copyright © 2012 A. Jawahar 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 wireless sensor network is composed of large number of sensor nodes and they are densely deployed in the field to monitor the environment, collect the data and route it to a sink. The main constraint is that the nodes in such a network have a battery of limited stored energy the network lifetime gets reduced. There are various topology management schemes such as SPAN, STEM, GAF, BEES and so forth, for improving network parameters such as capacity, lifetime, coverage and latency. These schemes do not improve all the mentioned network parameters. Sustainable Physical Activity in Neighbourhood (SPAN) scheme, preserves network capacity, decreases latency but provides less energy savings. Sparse Topology and Energy Management (STEM) scheme does not preserve capacity resulting in great energy savings and high latency. In the proposed scheme, new coordinator rule is implemented in SPAN, and then integrated with STEM. It is observed that the energy conserved increases by about 3.18% to 4.17% without sacrificing network capacity. Due to definite path in the proposed scheme the latency is reduced by almost half the latency of STEM scheme.

1. Introduction

Sensor nodes consist of a processing unit, transceiver unit, sensing unit, and power unit. A wireless sensor network [1, 2] consists of large number of sensor nodes densely deployed in the field. These nodes will monitor the environment and detect if any event occurs and sends the corresponding information to the sink. In sensor networks, the nodes operate with battery of limited energy storage capacity. The transceiver unit consumes more power compared to other units and the sensor nodes can be used efficiently by putting their transceiver units in an off state. Various topology management schemes have already been proposed [3–5] to use the transceiver effectively and improve the network parameters such as lifetime at the cost of latency and capacity. Sustainable physical activity in neighbourhood (SPAN) [6] is a topology scheme in which a few nodes will be in active state called coordinator and form a backbone path, lowering the latency in SPAN. The drawback of SPAN is that

it has less system lifetime compared to sparse topology and energy management (STEM). STEM [7], is another topology scheme in which each and every node will have two radios, a data plane radio, and a wake-up plane radio. Usually, the data plane radio will be in off state, and if any event occurs, the wake-up plane radio will send wake-up message and activate the data plane radio of another node. The main advantage of STEM is that it has longer system lifetime. The drawback of STEM is high latency and less capacity. In this paper, 80, 90, 100, 110, 120 nodes are deployed in various field sizes such as 60 m * 60 m, 85 m * 85 m, 105 m * 105 m (15 scenarios), and the interaction of STEM and SPAN is analysed in all these scenarios.

Wireless sensor networks (WSNs) have wide applications [8, 9] such as health, military, and environment monitoring for detecting any hazards. This growth has led to widespread popularity in wireless communication, and hence numerous research works are being carried out in this field. In large-scale wireless sensor networks or in hazardous applications,

it is impossible to either recharge or replace the batteries. Hence energy has to be used efficiently in order to improve the lifetime of the network. This motivated us to propose a scheme to improve the important network parameters.

The rest of the paper is organised as follows. Different topology management schemes are discussed in Section 2. Section 3 deals with the proposed scheme; in Section 4, the performance analysis are discussed. Section 5 concludes the paper.

2. Related Work

The main aim of topology schemes [3, 10–12] in wireless sensor network is to reduce the energy consumption and maintain connectivity to efficiently forward data to sink. In STEM, [7, 13, 14] the wake-up radio is periodically on for monitoring the environment. Typically, the data radio in the next hop between a node and the sink will be in the off state. To overcome this problem, each node will periodically turn on their radio for a short time to check if any other nodes want to communicate with it. In principle, the communication capacity could be reduced to virtually zero, by turning off the radios of all nodes (i.e., putting them in the sleep mode). When a possible event is detected, the main processor is woken up to analyze the data in detail. The radio, which is normally turned off, is woken up if the processor decides that the data needs to be forwarded to sink. Now, the problem is the radio of the next hop to the data sink is still turned off, if it did not detect that same event. As a solution, each node periodically turns on its radio for a short time to listen if someone wants to communicate with it [7, 15].

In most of the applications, nodes are in the idle state and waiting for the event to happen. STEM reduces the energy consumption of the sensor node by switching its radio off when it is idle. STEM with wake-up interval of 600 ms, which reduces the energy consumption of the node by a factor of about 2.5. STEM conserves energy at the expense of network capacity and higher latency.

In sustainable physical activity in neighbourhood scheme (SPAN) [6], only few nodes will be elected as coordinators. If two neighbours of a node cannot reach each other directly, then that node becomes a coordinator. The coordinators are selected based on this coordinator-election rule such that equal chance is given for all the nodes to become coordinators. The coordinators will form a definite backbone path in the network through which data is forwarded from source to sink. Since there is a definite backbone path, the latency is less. SPAN preserves network capacity.

In geographic adaptive fidelity scheme (GAF) [16, 17], the network is divided into several grids, and in each and every grid, only one node will remain in “on” state. Thus, if there are “ n ” grids in the network, then there will be only “ n ” nodes in “on” state in the network. Here energy is conserved. The main drawback of GAF is that it typically uses GPS to determine node location. In many settings such as indoors or under trees, where GPS does not work properly, location information is not available. The dependency on global location information thus limits GAFs usefulness, and

also GPS is used for location information that is relatively costly. GAF sizes its grid based on radio range R . In order to reach the nodes in the neighbouring grid, the size of the square grid is fixed as $r \leq (R/\sqrt{5})$. With this definition nodes in a grid can reach all the nodes in the horizontally and vertically neighbouring grids. As a result, few nodes in the corner of the diagonal grid is not reachable.

In the adaptive self-configuring sensor network topology scheme (ASCENT) [18], nodes will be in active, passive, test and sleep states. Active nodes participate in the data transmission. The nodes in test and passive state are checked continuously to become active for successful transmission of data packets. The node that detects the loss of packets sends help message to neighbouring nodes to join the network. The nodes in passive state receive these messages and check to become active node if needed. Hence data loss is reduced and successful transmission takes place.

In enhanced SPAN (E-SPAN) [19], directional antennas are used. In this method, data is sent in directional mode and “hello” message is sent in omni mode. The receiver was in omni-directional mode, and transmitter was either in omni or directional mode. Here only one antenna can be enabled at a time. Reduction in energy should be of the order of 2.7 times of the energy gain at the best case. Hence E-SPAN is more efficient than SPAN.

2.1. System Model. The following assumptions and notations are used in the proposed model shown in Figure 1.

Assumptions made:

- (i) both uniform and random deployments of nodes;
- (ii) fixed transmission radius;
- (iii) sensing range is less than transmission range;
- (iv) dual radio;
- (v) homogeneous network;
- (vi) boundary effects are negligible.

2.2. Problem Description. Wireless sensor nodes have a battery of limited energy, and therefore the network lifetime depends on how wisely the energy is used. In critical applications such as chemical plants, forest fires, and nuclear reactors, it is often not possible to replace or recharge the battery. Our objective was to improve lifetime and reduce the latency without sacrificing the capacity. We achieve this by integrating the definite backbone of SPAN and dual radio approach of STEM to develop the network architecture with low power consumption and low latency.

3. Proposed Scheme

In the proposed technique, two different topology management schemes, namely, sparse topology and energy management (STEM) scheme and sustainable physical activity in neighbourhood (SPAN) scheme are integrated. Each and every node has two radios, namely, data plane and wake up plane. The coordinators are elected using the proposed coordinator eligibility algorithm. The noncoordinator nodes

will be put in sleep state and coordinator nodes will be turned on. These coordinators will form a definite backbone path through which data is forwarded to sink. When the sensor node detects an event, it will wake up its radio when it is needed to transmit data to the sink. Here the problem is that the radio of the next hop in the path to the data sink is still turned off. To overcome this problem, each node periodically turns on its radio at same time as shown in Figure 7, to check whether any of the other nodes wants to communicate with it. The wake-up plane will wake up the data plane radio, and thus a connection is established between two nodes and thereby data is sent. A node, which wants to communicate with other node, is initiator node and node, which is been communicated, is target node. The proposed scheme has been tested by deploying various numbers of node such as 80, 90, 100, 110, 120 in different field sizes such as 60 m * 60 m, 85 m * 85 m, 105 m * 105 m, and performance was analysed. Due to definite backbone path and dual radio, latency is reduced and more energy is conserved, preserving the capacity of the network. Capacity is the number of packets the network can successfully deliver per unit time. It is inversely proportional to the network's packet loss rate. The proposed hybrid scheme does not degrade the network capacity as that of SPAN. Hence, the capacity is preserved.

3.1. Uniform Deployment of Nodes. Nodes are deployed uniformly as shown in Figure 2. This uniform deployment of nodes will form segments of hexagon. Uniform deployment of nodes is used for many static applications such as precision agriculture, car parking, and chemical plants [20, 21]. Nodes are deployed uniformly with distance between two nodes in horizontal direction as “ r ” and the distance between two nodes in vertical direction as “ h .” Here radio range is “ R ” = 20 metres.

3.1.1. Derivation of Mathematical Model for Node Deployment. Let, $L * L$ be the size of the field,

“ N ” be the total number of nodes,

“ N_x ” be the number of nodes in the horizontal direction,

“ N_y ” be the number of nodes in the vertical direction,

“ D_x ,” and “ D_y ” be the distance between two nodes in horizontal and vertical directions, respectively;

“ r ” be the side of the hexagon;

“ h ” be the half the height of the hexagon in the vertical direction;

“ R ” be the radio range.

For uniform deployment, the distance between the nodes in the horizontal direction (r) and the distance between the nodes in the vertical direction (h) are predetermined and deployed accordingly.

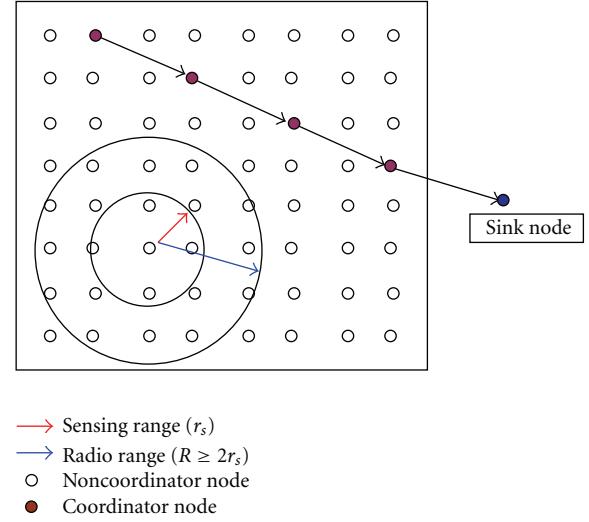


FIGURE 1: Proposed system model.

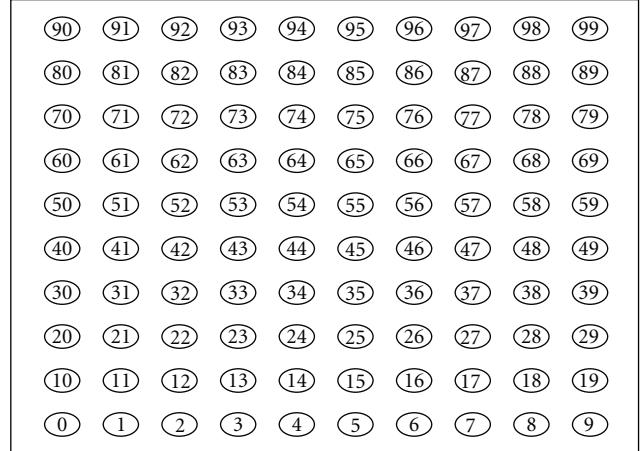


FIGURE 2: Uniform deployment of 100 nodes.

The relationship of the distance between the nodes in the vertical direction “ h ” and horizontal direction “ r ” is given by

$$h^2 = \left[r^2 - \left(\frac{r}{2} \right)^2 \right], \quad (1)$$

$$h = \frac{r\sqrt{3}}{2},$$

to ensure the connectivity $r \leq R$.

The product of nodes arranged in horizontal and vertical directions gives the total number of nodes. So

$$N_x N_y = N, \quad (2)$$

$$D_x = \frac{L}{N_x - 1}, \quad (3)$$

$$D_y = \frac{L}{N_y - 1}. \quad (4)$$

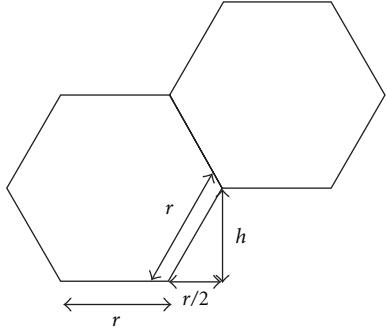


FIGURE 3: Calculation of location of nodes in uniform deployment.

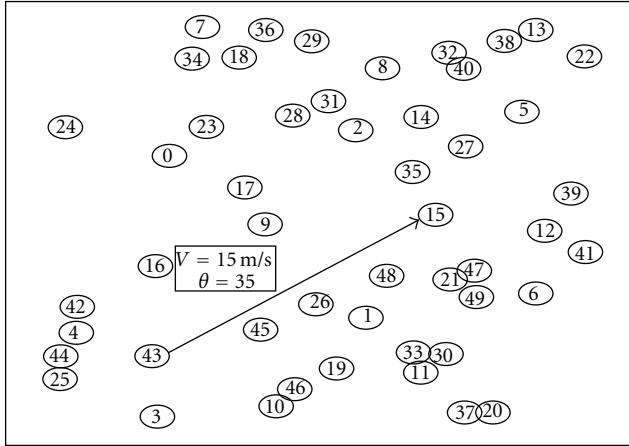


FIGURE 4: Random deployment of nodes (random waypoint model).

The distance between two nodes is arranged as shown in Figure 3, that is,

$$D_x : D_y = 1 : \frac{\sqrt{3}}{2}. \quad (5)$$

By using (2) to (5), we get,

$$1.155N_x^2 - 0.155N_x - N = 0. \quad (6)$$

By solving this we get the number of nodes in horizontal direction and by substituting this in (2) the number of nodes in vertical direction can be determined (see Algorithm 1).

3.2. Random Deployment of Nodes with Mobility. Here the nodes are deployed randomly in the field as shown in Figure 4 using random waypoint mobility model, which is the most widely used model [22]. Sensor nodes move randomly in the field and the destination, speed, and direction are all chosen randomly.

3.3. Coordinator Eligibility Rule. For more details (see Algorithms 2 and 3 or Figures 5 and 6).

3.4. Theoretical Analysis: Coordinator Calculation. In uniform deployment of nodes, the nodes will be at the vertex of hexagon [23]. Each hexagon has six vertices. Each node at the vertex is shared by three hexagons. Therefore, each hexagon has two coordinators. In order to find the number of coordinators, number of hexagons in the given area has to be known, which is given by

$$N_h = \frac{A_f}{A_h}. \quad (7)$$

Let the number of coordinator be “C” and non-coordinator be N_c :

$$\begin{aligned} C &= 2 \left[\frac{A_f}{A_h} \right], \\ C &= 2 \left[\frac{L^2}{6(\sqrt{3}/4)R^2} \right]. \end{aligned} \quad (8)$$

Thus, the number of coordinators is calculated.

The total number of nodes (N) is

$$N = C + N_c. \quad (9)$$

For N nodes, the coordinator node ratio is α and is given by

$$\alpha = \frac{C}{N}. \quad (10)$$

The noncoordinator node ratio is given by

$$\beta = 1 - \frac{C}{N}. \quad (11)$$

Thus,

$$C = N(1 - \beta). \quad (12)$$

Hence $N(1 - \beta)$ nodes will be in on state and $N\beta$ nodes will be in off state. For a field size of $L \times L$, the area is L^2 (m^2).

3.5. Theoretical Analysis: Latency Calculation. The initiator node will first start to send beacons to target node, and after receiving the beacons, the target node will respond to it. Once both of the nodes turned their data radio on, a link is established between them and data is transferred. If the transferred data is not intended for this node, then this becomes the initiator node and sends the packet to the node in the next hop towards destination, and this process is repeated. For simplicity, we did not use a location service in our simulations. A node obtains the location of the destination node from the general operations director (GOD) module in NS2 [24]. The location is required once per flow at the sender. Nevertheless, location services such as grids location service (GLS) [25] can be used with the hybrid scheme. To avoid the problem of interference between the wake-up beacon and the data transmission, transceiver uses dual radio and each radio operates at different frequency bands.

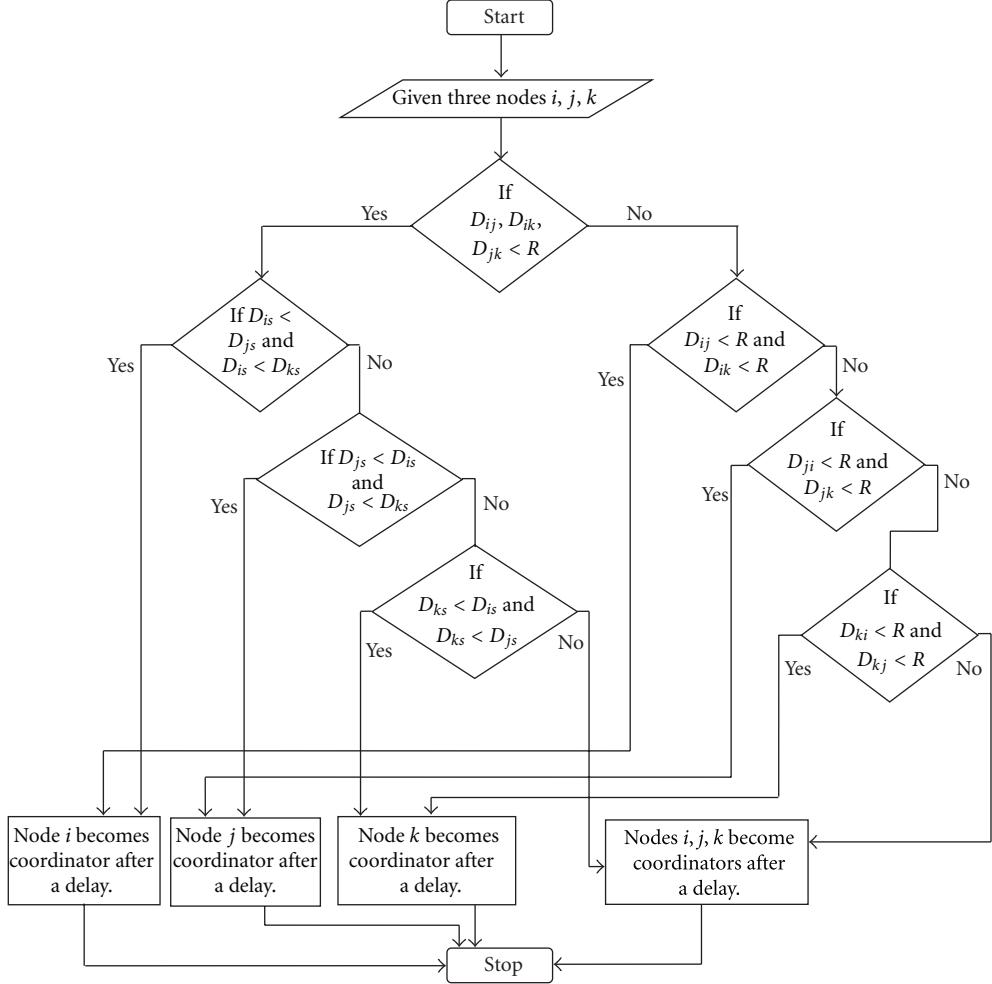


FIGURE 5: Flowchart for coordinator election.

The frequency band f_w (wake-up plane radio) is used to transmit the wake-up messages. Once the target node has received a wake-up message, both the nodes will turn on its radio operating at frequency band f_d . The data packets are transmitted in this frequency band, and they are called data plane (f_d). The time taken for this process is set up latency. The probability of the target node being turned on at the same time as the initiator node during the time interval “ T ” is given by

$$P(T_S = B_{XY}) = \frac{T_B - B_X}{T}. \quad (13)$$

If the total time period (T) is greater than “ON” time duration (T_b) of wake-up plane radio, then the average set up Latency per hop is given by

$$T_S = \frac{T + B_{XY}}{2}. \quad (14)$$

If the total time period (T) is equal to “ON” time duration (T_b) of wake-up plane radio, then the average set up latency per hop is given by

$$T_s = B_{xy}. \quad (15)$$

The total latency between source and sink is given by

$$T_l = T_s * (C - 1). \quad (16)$$

4. Performance Evaluation

The proposed hybrid topology management scheme is implemented and performance parameters like energy conservation, latency, and capacity are analysed and compared with SPAN and STEM topology management schemes. It is observed that the combined scheme has better performance and overcomes the limitations of both STEM and SPAN.

4.1. Simulation Environment. In this work, network simulator-2 (NS-2) tool is used, and both uniform and random deployments of nodes are considered separately over the sensor field. Here transmission range is fixed as $R = 20$ m, which has the radio characteristics as shown in Table 1. The power consumed by the node in transmit and receive mode is the power required for transmission and reception, respectively. Idle power is the power consumed when radio is on but no

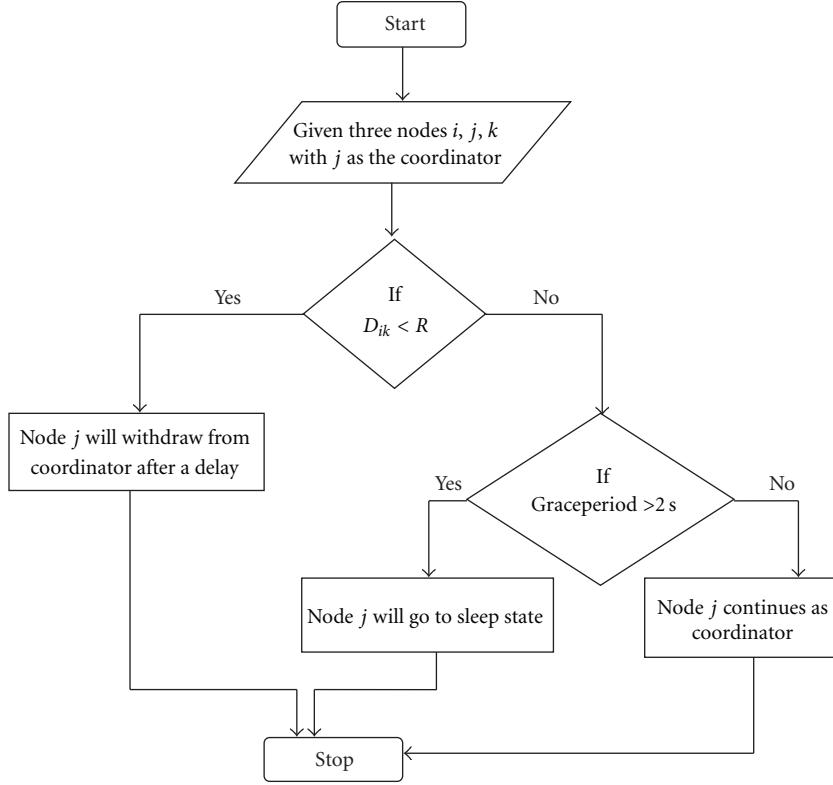


FIGURE 6: Flowchart for coordinator withdrawal.

Step 1: Compute the number of nodes in x -direction by solving for N_x ,

$$1.55N_x^2 - 0.155N_x - N = 0$$

Step 2: Compute the number of nodes in y -direction,

$$N_y = \frac{N}{N_x}$$

Step 3: Compute the distance between two nodes in x and y -direction,

$$D_x = \frac{L}{N_x - 1}, \quad D_y = \frac{L}{N_y - 1}$$

Step 4: If $(D_x > R) \&& (D_y > R)$ then communication capacity becomes zero.
else assign nodes as shown below.

$$\begin{bmatrix} \left(\frac{D_x}{2}, 0\right) & \left(\frac{3D_x}{2}, 0\right) & \left(\frac{5D_x}{2}, 0\right) & \dots & \left(\left(N_x - \frac{1}{2}\right)D_x, 0\right) \\ (0, D_y) & (0+D_x, D_y) & (0+2D_x, D_y) & \dots & (0+N_x D_x, D_y) \\ \left(\frac{D_x}{2}, 2D_y\right) & \left(\frac{3D_x}{2}, 2D_y\right) & \left(\frac{5D_x}{2}, 2D_y\right) & \dots & \left(\left(N_x - \frac{1}{2}\right)D_x, 2D_y\right) \\ (0, (N_y - 1)D_y) & (D_x, (N_y - 1)D_y) & (2D_x, (N_y - 1)D_y) & \dots & (N_x D_x, (N_y - 1)D_y) \end{bmatrix}$$

ALGORITHM 1: Node deployment algorithm.

data is transferred. Sleeping power corresponds to the power consumed when the node is in sleep state.

In this simulation, traffic loads are generated by constant bit rate (CBR) flows. Different number of nodes such as 80, 90, 100, 110, and 120 is deployed both uniformly and randomly in 60 m * 60 m, 85 m * 85 m, 105 m * 105 m field sizes separately. Here simulation time is set as 600 seconds.

4.2. Simulation Results. In uniform deployment of the nodes, the location of the nodes is predetermined and deployed.

For random deployment of nodes, nodes are deployed randomly, and random way point model is used as the mobility model. The proposed scheme is implemented by integrating STEM and SPAN topology management scheme, and coordinators are elected using proposed coordinator eligibility rule. The number of coordinators is observed in various field sizes (60 m * 60 m, 85 m * 85 m, 105 m * 105 m) by deploying different number of nodes (80, 90, 100, 110, 120). Figure 8 shows the number of coordinators for all these scenarios and it is inferred that combined

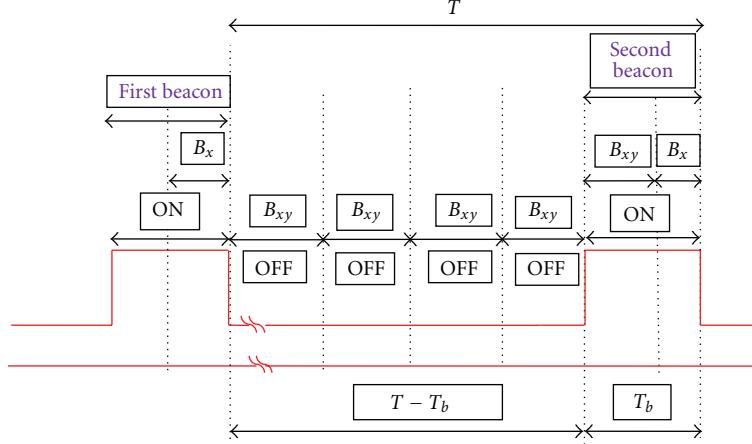


FIGURE 7: Latency analysis B_x : Transmit time of beacon; B_y : Inter beacon spacing; B_{xy} : Time taken for initiator node to send the beacon and receive response ($B_{xy} = B_x + B_y$); T_b : “ON” Time duration of target node ($T_b = B_{xy} + B_x$); and T : Total time period.

```

Step 1: Compute the number of neighbors for each node. Given nodes  $i$  and  $j$  in a wireless sensor network where all the nodes have the same radio range ( $R$ ).  $D_{ij}$  denotes the Euclidean distance from  $i$  to  $j$ , and  $D_{ij} \leq R$ . Node  $j$  is node  $i$ 's neighbor.
Step 2: Nodes having maximum neighbors waits for the following delay and later announces as a coordinator.

$$\text{Delay} = \left[ \left( 1 - \frac{E_y}{E_x} \right) + \left( 1 - \frac{C}{0.5n(n-1)} \right) * R \right] * n * T$$

Remaining nodes are put in sleep state.
Step 3: Let  $N_i, N_k$  are neighbors to  $N_j$ 
    If  $N_{ik} > R$ , then  $N_j$  become coordinator after the delay in Step 2.
Step 4: For all  $N_c$ , checks whether it is within the radio range of any coordinator node  $C$ .
    Else that  $N_c$  become coordinator  $C$ .
Step 5: If two or more nodes satisfies Steps 1 to 4, then each node check its distance to the sink.
Step 6: if ( $N_{is} < N_{js}$ )&& ( $N_{is} < N_{ks}$ ) then  $N_i$  becomes coordinator
        else if ( $N_{js} < N_{is}$ )&& ( $N_{js} < N_{ks}$ ) then  $N_j$  becomes coordinator
        else  $N_k$  becomes coordinator

```

ALGORITHM 2: Coordinator election algorithm.

TABLE 1: Radio characteristics.

Radio mode	Power consumption (W)
Transmit	0.01488
Receive	0.01250
Idle	0.01236
Sleep	0.000016
Simulation Time	600 s

scheme has almost same number of coordinator as that of SPAN.

Also, it is inferred that as the number of nodes increases, the coordinator will almost remain constant because only a smaller fraction of nodes will become coordinator. As the field size increases, the distance between two nodes will increase, but the radio range remains constant. Since the distance between the nodes increases, the number of neighbours getting benefited will reduce. In order to cover

all the nodes in the field, number of coordinators elected increases for a constant radio range. Similarly, the number of coordinators in random deployment is also analysed.

Figure 9 shows the number of coordinators in random deployment for various scenarios as considered in uniform deployment. Unlike uniform deployment, the number of coordinators keeps varying to cover the entire field due to mobility.

Figure 10 shows the total energy conserved in the network for both combined scheme and SPAN scheme for different scenarios in uniform deployment. It is inferred that as the number of node increases, the total energy conserved in the network also increases. This is due to the fact that as the number of nodes increases, number of coordinators will remain constant, whereas the number of non-coordinator will increase. These non-coordinator nodes are put in sleep state and thereby more energy is conserved. It is also inferred that the combined scheme conserves more energy compared to SPAN scheme because

```

Step 1: Each coordinator checks periodically if it should withdraw as coordinator.
Step 2: Let  $N_i, N_k$  are neighbors to  $N_j$ 
       if  $N_{ik} \leq R$ , then  $N_j$  withdraws its coordinator after the delay
Step 3: graceperiod = current time-last withdrawn
       if (graceperiod  $\geq 2$  s)
           sleep state

```

ALGORITHM 3: Coordinator withdrawal algorithm.

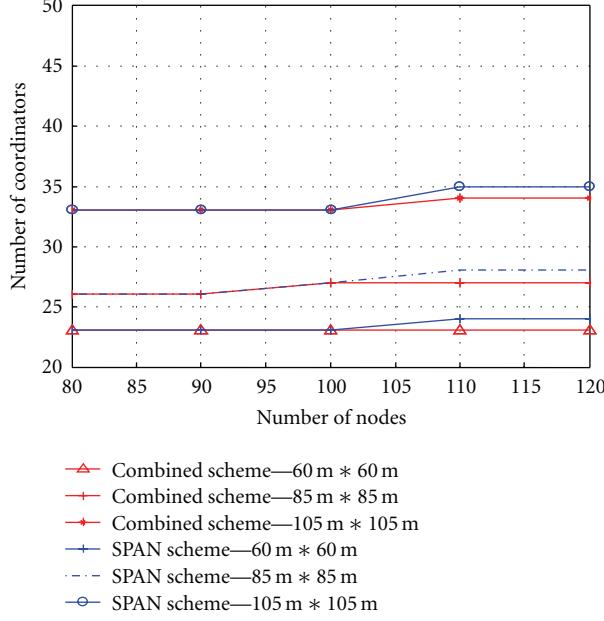


FIGURE 8: Number of coordinators in uniform deployment.

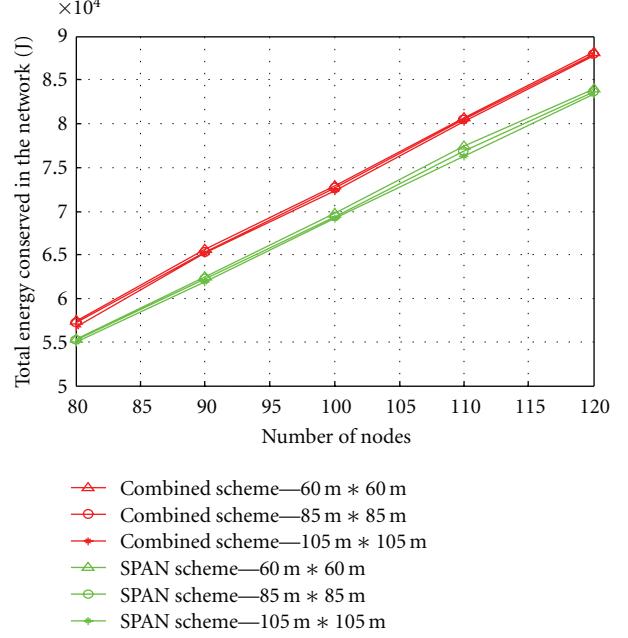


FIGURE 10: Energy conserved in the network (uniform deployment of nodes).

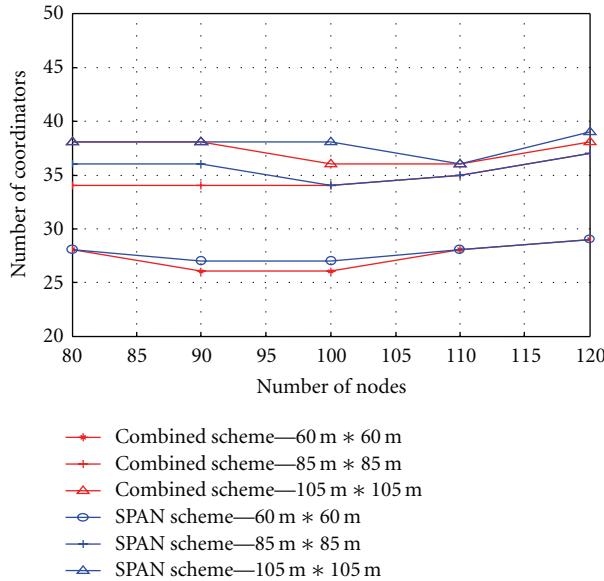


FIGURE 9: Number of coordinators in random deployment.

of low duty cycle radio concept and proposed coordinator eligibility rule. Here each and every node has initial energy as 1000 J. Hence, total energy conserved in the network is given as

$$E_{\text{conserved}} = E_{\text{total}} - E_{\text{consumed}}. \quad (17)$$

Similarly, total energy conserved in the network in random deployment for both SPAN and combined scheme for different scenarios is shown in Figure 11. It is inferred that the energy conserved in random deployment is less compared to uniform deployment. In random deployment, the combined scheme conserves more energy compared to SPAN scheme as the number of nodes increases and when field size is reduced.

Figure 12 shows the lifetime improvement factor of the combined scheme and SPAN scheme in uniform deployment. It is observed that combined scheme extends the network lifetime compared to SPAN. As the number of nodes increases, the lifetime improvement factor of the network increases for all the field size for both the schemes. SPAN improves the lifetime by about 3.3 times when the

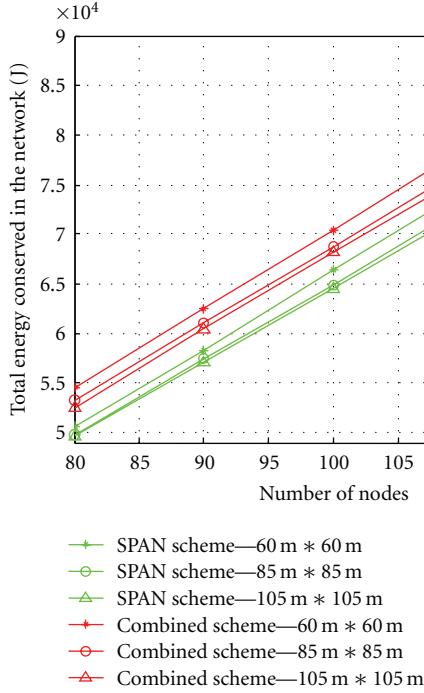


FIGURE 11: Energy conserved in the network (random deployment of nodes).

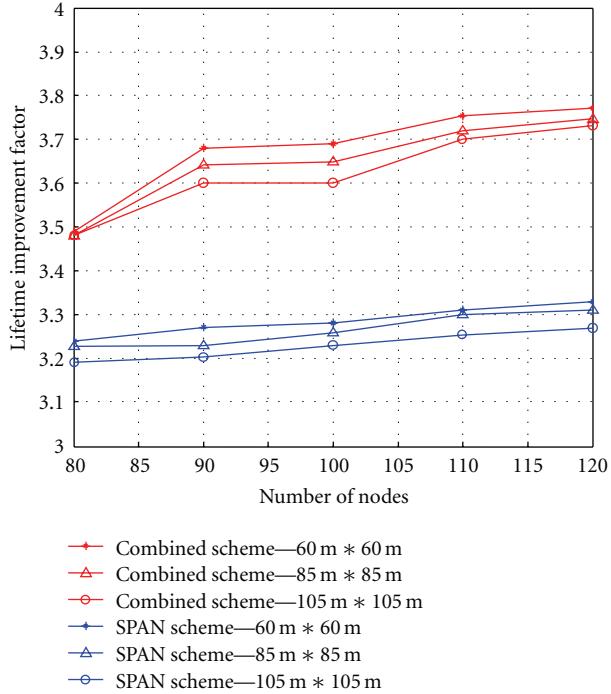


FIGURE 12: Lifetime improvement factor for uniform deployment of nodes.

number of nodes is 120 in 60 m * 60 m field size. Compared to SPAN, combined scheme improves lifetime further to approximately 3.8 when the number of nodes is 120 in 60 m * 60 m field size. It is inferred that as the number of nodes increases for a given field size, lifetime increases. Also

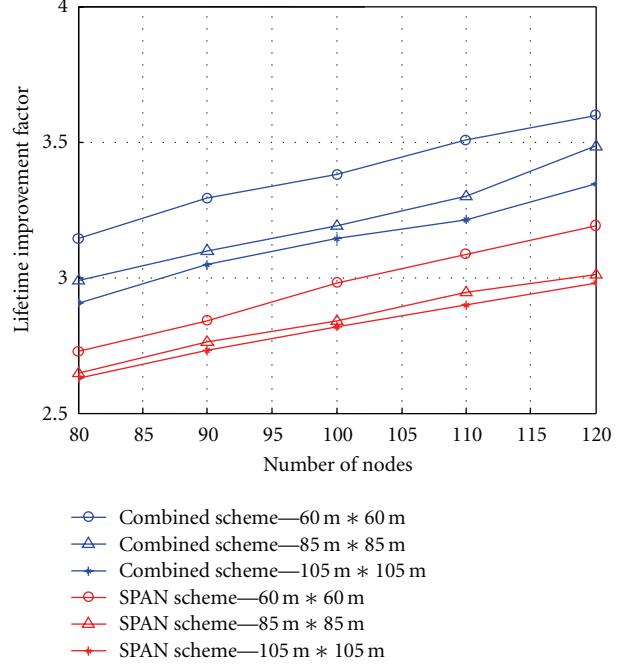


FIGURE 13: Lifetime improvement factor for random deployment with mobility.

as the field size is reduced for a given number of nodes, lifetime increases.

Figure 13 shows the lifetime improvement factor of combined scheme and SPAN scheme in random deployment. As the number of nodes increases, the lifetime improvement factor of the network increases for all the field size for both the schemes. SPAN improves the lifetime by about 3.2 times when the number of nodes is 120 in 60 m * 60 m field size. Compared to SPAN, combined scheme improves lifetime further to approximately 3.6 when the number of nodes is 120 in 60 m * 60 m field size. It is observed that as the number of nodes increases for a given field size, lifetime increases. Also as the field size is reduced for given number of nodes, lifetime increases. SPAN preserves capacity to a great extent thereby exploiting the energy factor. STEM conserves energy whereas it does not preserve capacity. Thus by combining STEM and SPAN, capacity is preserved and also energy is conserved. Capacity is the total number of packets delivered successfully per unit time. Here, graph is plotted between number of nodes and capacity by varying the field size as shown in Figure 14. The combined scheme preserves capacity without sacrificing energy and latency. It is observed that as the number of nodes increases, the capacity is reduced. Figure 15 shows the capacity in the network where random deployment is employed for eight different scenarios. The combined scheme preserves the capacity as that of the SPAN in all the scenarios. Figure 16 shows the latency for STEM and combined scheme by deploying nodes uniformly in the field. It is observed that STEM has more latency due to indefinite backbone path. This increase goes further as the number of nodes increases in

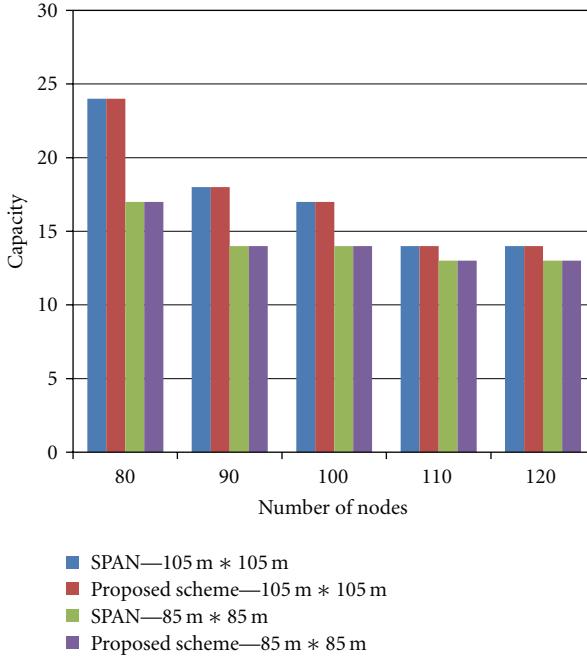


FIGURE 14: Capacity in uniform deployment network.

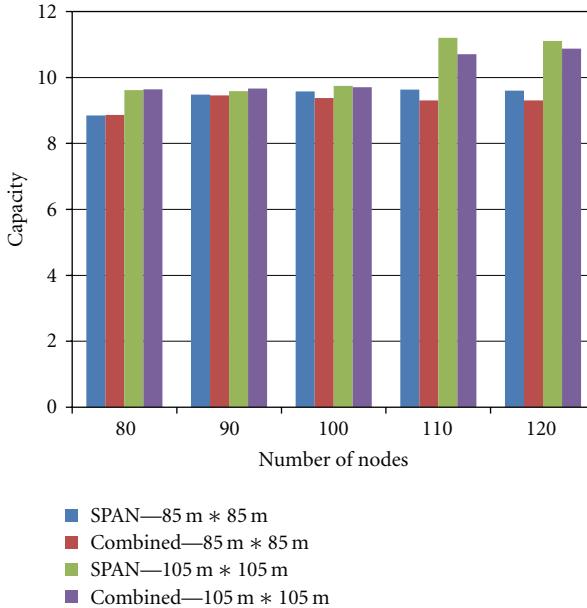


FIGURE 15: Capacity in random deployment network.

the different fields. Also the latency increases as the field size increases. The latency decreases for all the scenarios in combined scheme due to definite backbone path formed by the coordinators. The data is forwarded from source to sink through this path.

Figure 17 shows the latency of combined scheme and STEM scheme by deploying nodes randomly in the field. It is observed that the combined scheme has less latency compared to STEM scheme.

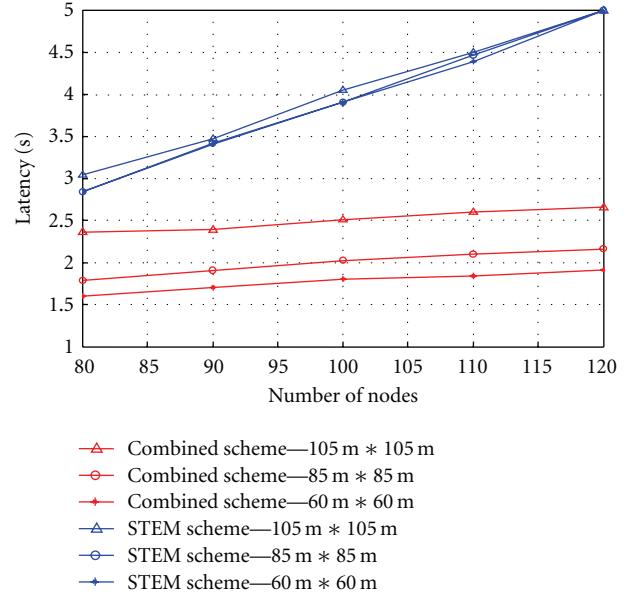


FIGURE 16: Latency in the network (uniform deployment).

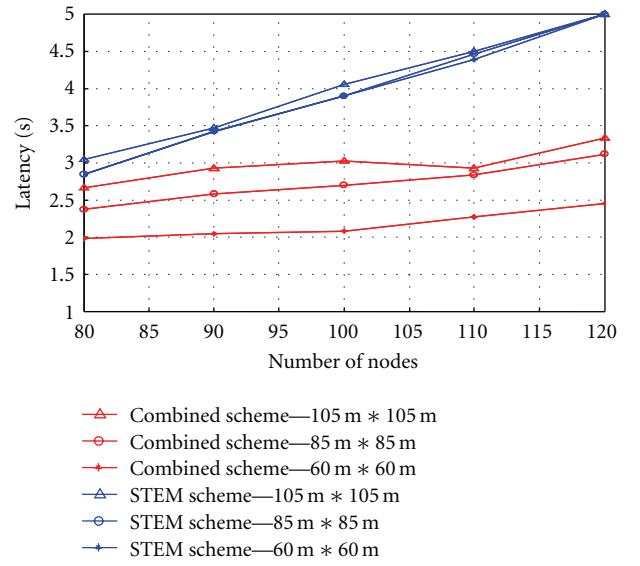


FIGURE 17: Latency in the network (random deployment).

5. Conclusion

In this paper, hybrid topology management scheme is proposed by integrating STEM and SPAN and coordinators are elected using new coordinator eligibility rule. Here various parameters such as energy conservation, lifetime, capacity, and latency are analysed by deploying 80, 90, 100, 110, and 120 nodes in various field sizes such as 60 m * 60 m, 85 m * 85 m, and 105 m * 105 m using uniform and random deployments of nodes. It is inferred that in both cases, as number of nodes increases, the total energy conserved in the network increases by keeping field size constant. Similarly, as the field size is reduced, the total energy conserved in

the network is increased for fixed number of nodes. Thus, it can be concluded that deploying more nodes (120 nodes) in small field size ($60\text{ m} \times 60\text{ m}$) conserves more energy.

It is also inferred that the combined scheme results in more energy conservation compared to SPAN, which reflects in further improvement in lifetime factor of about 3.8 compared to SPAN. The combined scheme has less latency compared to STEM due to definite backbone path through which the data is forwarded to sink. These improvements in the network parameters such as energy conservation, lifetime, and latency are achieved without sacrificing capacity. The critical application requires the data to be forwarded quickly to the sink by conserving more energy without any loss of data. Since the proposed scheme meets all these requirements, it will be well suited for hazardous applications.

Notations and Definition

R :	Radio range
r :	Distance between two nodes in horizontal direction
h :	Distance between two nodes in vertical direction
f_d :	Data plane radio
f_w :	Wake up plane radio
B_x :	Transmit time of beacon
B_y :	Inter beacon spacing
B_{xy} :	Time taken for initiator node to send the beacon and receive response
T_b :	"ON" Time duration of target node
T_s :	Set up latency
C :	Number of coordinators
N :	Total number of nodes
N_c :	Number of non-coordinators
α :	Coordinator node ratio
β :	Non-coordinator ratio
$E_{\text{conserved}}$:	Energy conserved in the network
E_x :	Maximum amount of energy
E_y :	Remaining amount of energy
N_h :	Number of hexagons
A_f :	Area of the field
A_h :	Area of the hexagon.

References

- [1] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*, John Wiley & Sons, New York, NY, USA, 2010.
- [2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [3] S. Jardosh and P. Ranjan, "A survey: topology control for wireless sensor networks," in the *International Conference on Signal Processing Communications and Networking (ICSCN '08)*, pp. 422–427, January 2008.
- [4] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Computing Surveys*, vol. 37, no. 2, pp. 164–194, 2005.
- [5] X. Li, Y. Mao, and Y. Liang, "A survey on topology control in wireless sensor networks," in *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV '08)*, pp. 251–255, Hanoi, Vietnam, December 2008.
- [6] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 481–494, 2002.
- [7] 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.
- [8] C. S. Raghavendra, K. Sivalingam, and T. Znati, *Wireless Sensor Networks*, Springer, Berlin, Germany, 2004.
- [9] R. V. Biradar, V. C. Patil, S. R. Sawant, and R. R. Mudholkar, "Classification and comparison of routing protocols in wireless sensor networks," *UbiCC Journal*, vol. 4, pp. 704–711, 2009.
- [10] M. A. Labrador and P. M. Wightman, *Topology Control in Wireless Sensor Networks*, Springer, Berlin, Germany, 2009.
- [11] M. Ilyas and I. Mahgoub, *Handbook of Sensor Networks: Wireless and Wired Sensing Systems*, CRC Press, 2004.
- [12] Mo Li and Baijian Yang, "A survey on topology issues in wireless sensor network," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, Los Angeles, Calif, USA, April 2005.
- [13] Y. Rong, S. Zhi, and M. Shunliang, "Scalable topology and energy management in wireless sensor networks," in *IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 3450–3455, March 2007.
- [14] "Vlasisos Tsiatsis in "Topology Management for Sensor Networks: Exploiting Latency and Density"" IEEEAC paper.
- [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] Z. Jiang, J. Wu, A. Agah, and B. Lu, "Topology control for secured coverage in wireless sensor networks," in *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS '07)*, October 2007.
- [17] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in the *7th Annual International Conference on Mobile Computing and Networking*, pp. 70–84, July 2001.
- [18] A. Cerpa and D. Estrin, "ASCENT: adaptive self-configuring sensor networks topologies," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 272–285, 2004.
- [19] V. Kumar, T. Arunan, and N. Balakrishnan, "E-SPAN: enhanced-SPAN with directional antenna," in *Proceedings of IEEE Conference on Convergent Technologies for Asia-Pacific Region*, vol. 2, pp. 675–679, 2002.
- [20] K. Nirmal Kumar, V. R. Sarma Dhulipala, R. Prabakaran, and P. Ranjith, "Future sensors and utilization of sensors in chemical industries with control of environmental hazards," in the *2nd International Conference on Environmental Science and Development (IPCBEE '11)*, vol. 4, 2011.
- [21] C. F. García-Hernández, P. H. Ibargüengoytia-González, J. García-Hernández, and J. A. Pérez-Díaz, "Wireless sensor networks and applications: a survey," *International Journal of Computer Science and Network Security*, vol. 7, no. 3, pp. 264–273, 2007.
- [22] T. Camp, J. Boleng, and V. Davies, "A survey on mobility models for Ad hoc networks Research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

- [23] R. P. Liu, G. Rogers, S. Zhou, and J. Zic, "Topology Control with Hexagonal Tessellation," February 2003.
- [24] "The ns Manual," http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf.
- [25] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, "Scalable location service for geographic ad hoc routing," in *the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 120–130, August 2000.

Research Article

Constructing a Distributed AUV Network for Underwater Plume-Tracking Operations

Stephanie Petillo, Henrik Schmidt, and Arjuna Balasuriya

*Laboratory for Autonomous Marine Sensing Systems, Department of Mechanical & Ocean Engineering,
Massachusetts Institute of Technology, Room 5-204, 77 Massachusetts Avenue, Cambridge, MA 02139, USA*

Correspondence should be addressed to Stephanie Petillo, spetillo@mit.edu

Received 1 June 2011; Revised 26 August 2011; Accepted 6 September 2011

Academic Editor: Don Sofge

Copyright © 2012 Stephanie Petillo 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 recent years, there has been significant concern about the impacts of offshore oil spill plumes and harmful algal blooms on the coastal ocean environment and biology, as well as on the human populations adjacent to these coastal regions. Thus, it has become increasingly important to determine the 3D extent of these ocean features (“plumes”) and how they evolve over time. The ocean environment is largely inaccessible to sensing directly by humans, motivating the need for robots to intelligently sense the ocean for us. In this paper, we propose the use of an autonomous underwater vehicle (AUV) network to track and predict plume shape and motion, discussing solutions to the challenges of spatiotemporal data aliasing (coverage versus resolution), underwater communication, AUV autonomy, data fusion, and coordination of multiple AUVs. A plume simulation is also developed here as the first step toward implementing behaviors for autonomous, adaptive plume tracking with AUVs, modeling a plume as a sum of Fourier orders and examining the resulting errors. This is then extended to include plume forecasting based on time variations, and future improvements and implementation are discussed.

1. Introduction

The underwater environment itself is hazardous to humans, as we cannot survive without air to breathe and our bodies cannot withstand the ambient pressure deep underwater, yet we could not exist without the presence of large bodies of water on our planet. The health of the oceans has a significant impact on both marine and human life. This has been observed most recently through the impact of offshore oil spill plumes and harmful algal blooms (HABs) on coastal waters. However, even in healthy ocean conditions, the ocean environment can be dangerous for humans, such as near the extreme temperatures and chemicals spewing out of hydrothermal vents into fluid clouds deep in the ocean. These features of the ocean environment create a challenge for underwater exploration and oceanographic data collection. The use of autonomous (unmanned) underwater vehicles (AUVs) in such environments is crucial to safely and efficiently completing these tasks, as they can be designed to withstand biological and chemical contaminants, high pressures, and extreme temperature variations. AUVs

(especially actively propelled ones) can also be programmed to react autonomously and adaptively to changes in their environments by controlling their own motion, unlike drifters, moored sensing arrays, or sensing buoys.

Oil spill plumes, HABs, and clouds of hydrothermal vent fluid in particular can each be viewed as a type of underwater plume (much like a cloud or plume of smoke), evolving in 3D space and over time. These plumes can range in scale from tens of meters to hundreds of kilometers in horizontal space at their neutrally buoyant depths and move with the prevailing currents as well as spread and diffuse into the surrounding water masses [1, 2]. Trying to track meso- and large-scale features (as plumes often are) with relatively small AUVs requires the coordinated effort of multiple AUVs, due largely to both battery life and AUV speed limitations. Willcox et al. [3] take a unique approach to this challenge in which they determine an optimal AUV survey and sampling strategy by quantifying an AUV’s energy efficiency, quantifying the degree of synopticity with which an AUV can measure an ocean process, and accounting for inherent survey errors in the sampling strategy. Plume tracking also

brings forth the problem of spatiotemporal aliasing of data when the plume is too large and/or moving too fast for a single AUV to collect a cohesive data set to accurately detect and track the plume edges as the plume evolves in space and time. That is, the samples taken by the AUV(s) must overlap within the plume's characteristic temporal and spatial scales to collect a synoptic data set. The importance of an ocean feature's spatial and temporal scales to feature detection and classification using AUVs is further emphasized by the work of Zhang et al. [4]. Thus, in this paper, we address the motivation for and challenges of constructing a network of AUVs to perform plume boundary tracking over two dimensions in space (horizontal) with time variations. We have chosen to track the boundary of a plume, rather than its center or maximum concentration, because the boundary gives a complete picture of the plume's spatial extent in the horizontal plane, where it is most likely to intersect a coastline or get entrained by currents and carried to another part of the ocean. We also present a simulated plume environment sampled by AUVs, from which we attempt to reconstruct the plume as a sum of Fourier orders as an initial estimate of the plume shape. The example of an oil leak, such as that from the Deepwater Horizon disaster in the Gulf of Mexico in 2010 [1], will be used to motivate a number of numerical assumptions in this paper, though we try to keep this first-pass plume simulation as general as possible to other types of plumes as well.

In addition, it is useful to know a bit about the AUVs we are using to guide numerical values for AUV simulation. For most field trials and autonomy testing, our group in the Laboratory for Autonomous Marine Sensing Systems at the Massachusetts Institute of Technology uses two Bluefin 21" AUVs (21" hull diameter, ~3 m in length), as shown in Figure 1. These vehicles demonstrate the best motion and stability control at speeds between 1 and 1.8 m/s, with navigational error of about 1%–5% of the distance traveled between surfacing to get a position fix via GPS. The AUVs navigate using a Leica DMC-SX Magnetic Compass and a Crossbow AHRS (attitude heading reference sensor). The navigational error quoted above assumes the AUV has constant DVL (Doppler velocity log) bottom lock, has completed a compass hard iron/soft iron calibration and has completed a compass star maneuver (for compass calibration in the water). Beyond this, the Bluefin software on the AUV also does some calibrations and math to improve the navigational accuracy to achieve the range above. To maintain reasonable stability control and navigational accuracy, the AUVs are usually commanded to travel at 1.5 m/s (though this speed varies due to autonomous adaptation to the AUVs' situations) and surface for a GPS position fix every 30 minutes, resulting in about 50–100 m of navigational error. Other instrumentation currently on board consists of a conductivity-temperature (CT) sensor, a pressure sensor, and an acoustic modem with transducer; however, these vehicles could also be equipped with sensors that measure chemical tracer concentrations or biological (Chlorophyll-a, colored dissolved organic matter, etc.) concentrations for the purposes of detecting oil, hydrothermal vent fluid, or algal concentrations. For communicating with the AUVs (Sections 3.3 and 4), we make extensive (and nearly exclusive) use of an



FIGURE 1: One of the Bluefin 21" AUVs operated by the MIT Laboratory for Autonomous Marine Sensing Systems.

acoustic communication structure (AUV-to-AUV and AUV-to-ship/lab) that has been actively developed and refined in recent years to give virtually real-time updates (delays on the order of minutes) of scientific and navigational data (more details on this are found in the Goby project documentation [5, 6]). Linking all of these pieces together is the autonomy system on board each AUV. This includes the Mission Oriented Operating Suite (MOOS) and the IvP Helm (IvP stands for Interval Programming), which coordinate to implement the execution of autonomy behaviors by the AUVs. These behaviors autonomously and adaptively control the heading, speed, and depth of the vehicle, depending on the behavior the AUV operators have chosen to run (more on this in Section 4 and [7, 8]).

2. Spatiotemporal Aliasing Problem

One of the most common challenges of working with AUVs to track ocean features is that of spatiotemporal aliasing, that is, when the samples taken are too far apart in space and/or time to be able to resolve the boundaries or position of a dynamic feature at a given point in space and time. This is effectively a trade-off between data coverage and data resolution. There are two extremes here (for example).

- (1) A single AUV can survey a small area ($\sim O(1 \text{ km})$, low spatial coverage) with very high spatial sampling resolution ($> O(1 \text{ sample/m})$) to resolve small-scale features in the water, such as pockets of turbulence. However, this survey would not have great enough coverage to determine the bounds of a 10 km wide algal bloom encompassing the sampling area.
- (2) A single AUV can survey an area once over a long time period ($\geq O(10 \text{ hr})$, high temporal coverage) for hydrothermal vent plumes. However, it may take so long ($> 10 \text{ hours}$) to perform a spatially comprehensive survey, as witnessed by Jakuba et al. in [9] that the plume has advected away from its initial surveyed position during that period (poor temporal resolution), and the survey must be redone with less coverage to resolve the motion of the plume.

Somewhere in the middle of the above “coverage versus resolution,” scenarios resides a delicate balance in which the characteristic scales of a dynamic feature (say, a plume of oil) coincide with (one-half) the rate at which the feature is sampled. This is essentially a sampling of the plume at its spatial and temporal Nyquist frequencies to maximize both coverage and resolution of the plume within the data set. Thus, it is necessary to know the characteristic spatial and temporal scales of the feature of interest for more intelligent path-planning purposes (see Figure 2), most likely involving multiple AUVs for tracking mesoscale features that are dominantly dynamic in two or more dimensions of space or any feature highly dynamic in time (such that an AUV moving ≤ 2 m/s could not keep up).

The necessity for designing a multi-AUV network to implement more intelligent and efficient mission planning is highly motivated by this aliasing problem, and relevant methods used by Willcox et al. and Zhang et al. to optimize AUV surveys and motivate the use of solo and multiple AUVs in efficient spatiotemporal ocean sampling and feature tracking will be important to take into account in implementing robust plume-tracking algorithms and techniques on board AUVs [3, 4].

3. Advantages and Challenges of an AUV Network

3.1. Working as a Team. An AUV network allows for the dynamic interaction of multiple AUVs to better adapt to dynamic features in the marine environment. That is, a network of AUVs has the ability to distribute its nodes around the entire boundary of a plume and move with the plume boundary, whereas a solo AUV may be optimally placed for sampling within a plume but could not determine the horizontal spatial extent of a plume and track it simultaneously on its own. Using the estimated characteristic scales of the plume (from satellite imagery, past surveys, or physics-based calculations) in guiding the AUV autonomy behaviors (described in Section 4), the network of AUVs can be distributed in space and time to detect and track the plume boundary and avoid aliasing the data. This desire for adaptive feature tracking also underscores the necessity for using mobile (self-propelled) sensing platforms instead of, or in conjunction with, fixed and drifting sensing platforms (e.g., buoys, Argo floats), such that sampling is performed more efficiently (minimizing overlapping data), and the scientist can be certain that he/she has captured a complete data set describing the plume.

3.2. Autonomous Coordinated Control. The brains behind coordinating a sophisticated network of AUVs for plume tracking is the underlying autonomy system that must run on board each AUV. An autonomy system, such as that described in Section 4, allows an AUV to adapt to its environment in near real time, without human intervention. A few of the minimum requirements of using and interacting with a robust autonomy system are inter-AUV (acoustic) communications, support for (user-supplied) adaptive autonomy behaviors to be executed by the AUVs, and an intelligent

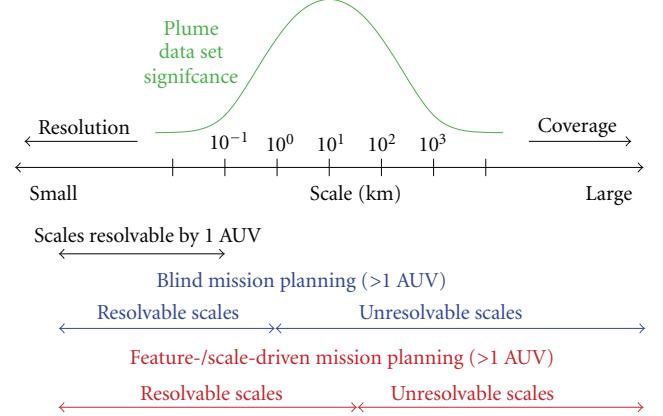


FIGURE 2: This figure depicts the characteristic length scale (in km) of an $O(1\text{ km})$ plume in the horizontal plane. A similar figure can be drawn for the temporal dimension based on the characteristic time scale of a plume (with units of time). If we assume a plume has an approximate Gaussian distribution over its characteristic length scale, as shown here, we must plan AUV missions, such that the collective sampling of our AUVs overlaps with the primary length scale of the plume to optimize over coverage and resolution (“feature-/scale-driven” mission planning). This will improve the range of resolvable length scales in the resulting data set over that of “blind” mission planning, especially when the AUVs’ distribution is “driven” by the characteristic spatiotemporal scales of the plume. Adapted from [12].

(autonomous) means of deciding which behaviors have priority during a given mission. We propose a tiered mission planning structure for this system in which the large-scale, overall mission drives the initial formation of the AUVs (assigning each an initial position) and then allows each AUV to use individual autonomy behaviors to follow the plume edge in its local vicinity. After a period of time, the local data collected by all AUVs is then exchanged across the network to update the plume model and, subsequently, the large-scale mission of the AUVs. From here, the overall mission, to local missions, to data collection, exchange, and reprocessing loop continues for as long as required by the scientist/user.

3.3. Acoustic Communication. One of the primary challenges of using multiple AUVs simultaneously in the underwater environment is that of communication. Radio frequency (RF) waves are quickly attenuated in the water within a few meters of the surface, leaving acoustics as the primary method of real-time underwater communication. Until now, there have been few (if any) options for intelligent multi-AUV (>2 AUVs) acoustic communication schemes, though the Goby underwater communication and autonomy project (version 2.0) strives to remedy the need for coordinated message queuing and passing between multiple (and potentially an unknown total number of) AUVs [5, 6]. This will allow each AUV to discover and communicate with neighboring AUVs and share data and knowledge with the sensing platforms in its underwater network. As this part of version 2.0 of the Goby project is still in development, it is currently undergoing initial field testing and will hopefully come into use in the next year.

It is important to note, however, that plumes are often mesoscale features or larger, and AUV-to-AUV and AUV-to-ship/lab acoustic communication (at least in the public domain and on power-limited AUVs) is only possible up to a range of about 10 km. Our group at MIT has found that our equipment is usually limited to about 2 km of acoustic communication range in the coastal ocean and lake environments we have performed most experiments in recently. Our Bluefin 21" AUVs and lab setup, which are each equipped with a WHOI Micromodem and model WH-BT-2 28 kHz transducer, transmit data in the frequency band of 23–27 kHz, centered around 25 kHz [10]. There are two realistic solutions to the acoustic communication range restriction we experience. The first and more complex solution is to implement a multihop acoustic communication scheme in which data from one AUV is passed down through a chain of AUVs to its destination. This is time consuming due to the nature of sending and listening for transmitted data packets, one at a time between communicating AUVs. Given that AUVs will often be hundreds of meters apart or more and sound speed propagation is about 1500 m/s in the ocean, data packets take an observable amount of time to transmit through the water ($O(1 \text{ sec})$). This method would also require extensive research into data routing on dynamic and time-scheduled messaging networks. The second and more immediately feasible (potentially more reliable) solution would be to restrict communication of large environmental data sets to RF or satellite methods while an AUV is on the surface and utilize a delay tolerant network rescheduling scheme. Although this method removes much of the real-time underwater data passing between AUVs (with the exception of basic position updates of nearby AUVs for avoiding collisions), it would take a large burden off of the acoustic channel and still allow each AUV to be redirected based on the most current overall picture of the plume while still performing solo autonomous and adaptive plume boundary tracking in its local vicinity in real time. Periodic surface communication would work best in the case that the AUVs can surface with great enough frequency (within the characteristic time scale of the plume) to be re-directed to a more optimal sampling position but with low enough frequency that the plume tracking mission is not significantly disrupted by the AUV taking the time to come to the surface more often.

3.4. Data Fusion. The fusion of data both from multiple sensors on a single AUV and all sensors across all networked AUVs is crucial to the success of coherently adapting a fleet of AUVs to track an ocean feature and collect a synoptic data set. When fusing data from a single vehicle, the largest concerns are keeping all data accurately time and position stamped. Across multiple AUVs, the data must also be quality checked for corruption during transmission after passing it from one vehicle to the next. It is proposed that on-board each AUV, the computer must mesh the data sets from all AUVs into a single data set, sorted over the times and positions at which each data point was taken, for each variable (i.e., temperature, salinity, etc.). Upon processing of these data on board (as on board processing is the only way to adapt to a dynamic environment in real time), for each

variable, probability weighting functions over time and space must be applied to each data point based on the characteristic spatiotemporal scales of that variable. We prefer to use a basic Gaussian-shaped weighting function for this task. This will associate, say, all temperature readings taken in the last few minutes and within a radius of a kilometer horizontally (assuming the AUV can resolve its position with even better accuracy), but will ignore any temperature readings that fall outside of these ranges as independent from those inside. This essentially creates an overlap of data within a radius of one standard deviation about the sample point, as sketched in Figure 3, that can be used to prevent insufficient sampling in a data set. This data fusion method could be implemented using an SQLite (or similar) database on each AUV to compound and sort all of the environmental data from all AUVs, which may then be processed in a mathematics program such as MATLAB or Octave, or by a simple C++ parser with algorithms utilizing C++ vector math libraries. This is similar to creating an evidence grid of the AUVs' environmental data [11]. The resulting ocean environment reconstructed through data fusion with weighting can guide the mission planning for a fleet of AUVs tasked to track a plume. The AUVs can survey an area with high enough resolution to find the boundary of the plume, approximate the plume shape (see Section 5) with higher weighting near the actual sample points, and revise their coordinated survey strategy based on this new estimate of the plume boundary position.

4. Adaptive Behavior Implementation

When conducting field experiments with AUVs (usually only 1 or 2) in the water, our group at MIT runs the Mission Oriented Operating Suite (MOOS) as the underlying autonomy system on board the AUVs and on our topside mission-command computer. MOOS provides a publish-subscribe architecture that essentially deals with information sharing between autonomy processes and behaviors on board each AUV, as well as through the water between the AUVs and the topside computer [7]. To add some intelligence to the system, the IvP Helm (IvP stands for Interval Programming) is used in conjunction with MOOS to implement the use of autonomy behaviors (e.g., vertical yo-yos, trail-an-AUV, horizontal racetracks, and safety behaviors) on the AUVs, optimizing over a vehicle's speed, heading, and depth [7, 8]. The acoustic communications are handled through the Goby (stable version 1.0) autonomy software on all platforms, where it schedules the transmissions of each node (AUVs, communication buoys, topside operator, etc.) in the network [5, 6]. Goby encodes data on one node, initializes the data transmission through the acoustic channel, and then decodes the data when they are received on another node. All of these pieces to our autonomy architecture allow our AUVs to adapt their motion based on sensor readings, without a human in the loop. This allows for ocean feature detection and tracking by AUVs to occur both autonomously and adaptively, as demonstrated in the following examples.

4.1. Thermocline Tracking as a Proof of Concept. The aforementioned autonomy system has been put to the test in

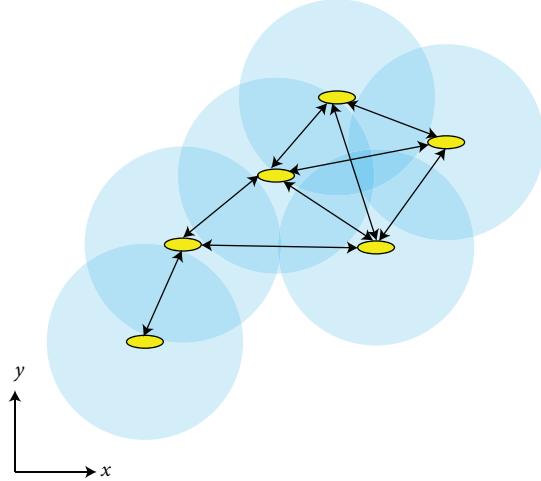


FIGURE 3: Blue circles around AUV sample points represent the range of significant data association possible (the radius of standard deviation of the Gaussian distribution). For any two AUV samples with overlapping range circles, an arrow is drawn to represent the fusion of data between those positions, which may be used to construct a larger-scale ocean data model when chains of fused data are combined to form a web of unaliased connections.

performing autonomous, adaptive thermocline tracking in the Tyrrhenian Sea (Italy) and Lake Champlain (Vermont, USA). As described in [13], a simple thermocline-tracking algorithm, which also accounts for the characteristic scales of the thermocline, has been developed and tested over the past few years using single AUVs of varying manufacture. Figure 4 is a conceptual sketch of the adaptive thermocline-tracking process, while more detail can be found in [13]. The idea here is that the thermocline, which is a feature only qualitatively defined in most oceanographic literature, must be quantitatively defined using actual data in real time for more efficient and adaptive oceanographic sampling. Here, it is assumed that the thermocline is relatively homogeneous in horizontal space within the AUV's operational region (for our vehicles, usually about 25 km^2 or less). That is, given an AUV's temperature measurements through the water column, on-board processing of the temperature data is accomplished spatially in 1D by binning the temperatures by depth ranges smaller than the characteristic (vertical) length scale of the thermocline in the experimental area ($O(10 \text{ m})$ in shallow water) and using finite differences to determine the region of greatest change in temperature over change in depth. The characteristic time scale of shallow water thermoclines (in the regions this algorithm has been tested) was determined by observation during our field trials to be $O(1 \text{ hr})$. Thus, temperature measurements were averaged over windows of 30 minutes to smooth out small local variations and spurious data points. Once the thermocline region has been determined by the AUV, the AUV will autonomously adapt its depth range to stay within the current boundaries of the thermocline and continue to collect a synoptic data set through the thermocline without expending extra energy to dive unnecessarily deeper or shallower.

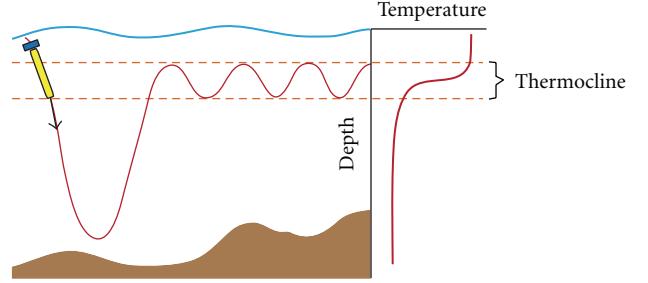


FIGURE 4: A conceptual sketch of an AUV performing thermocline tracking. The AUV completes a dive from the surface to as deep as allowable, collecting temperature data. The depth range of maximum temperature change per unit depth is determined as the thermocline region. The calculated upper and lower bounds of the thermocline region are then used to bound the vertical yo-yos of the AUV, essentially tracking the thermocline region. Used with permission from [13].

The successful field testing of this thermocline tracking process serves as a proof of concept for the feasibility of performing adaptive, autonomous feature tracking with an AUV, guided by the feature's spatial scale in 1D (vertically) and temporal scale to drive intelligent and efficient data collection. Thermocline tracking provides a solid first stepping stone into the field of multidimensional oceanographic feature tracking, from which we can move on to implementing applications with more complex features (dynamic in 2D or 3D space, and time) such as underwater plumes.

4.2. Plume Tracking. Plume detection and tracking using AUVs has come to the forefront of the oceanographic research community in recent years through the impacts of HABs and oil spills on coastal populations and the intrigue of studying the alien environment in the vicinity of hydrothermal vents. Smith et al. use a regional ocean model to predict the advection of a patch of water representing a HAB off the California coast, which is tagged by an actual Lagrangian drifter to passively mark and track the centroid of the imaginary HAB. AUVs (gliders) are then deployed to arrive at waypoints on the approximate boundary of the HAB when the HAB is predicted to reach that point. The calculated arrival paths of the AUVs are based on the plume boundary predictions from a regional ocean model, and the waypoints of the gliders are updated every few hours based on the previous dive's data and the model's predictions of the future boundary location of advecting the patch of water [14]. Similarly, Das et al. use satellite and high-frequency (HF) radar data sets to determine the location of high-concentration HAB patches and target these "hotspots" using AUV- (glider-) path-planning algorithms guided by the paths of the drifter tags for finer resolution sampling [15]. In a second paper, Das et al. expand this HAB tracking method further to perform Lagrangian observation studies in which the AUVs' (gliders') survey paths are precalculated to survey an advecting patch of water in its Lagrangian frame of reference to maintain sufficient spatial and temporal data resolution [16].

The difference between the aforementioned works and the implementation methods in this paper lie in the ability

of the propelled AUVs we propose to use to exhibit much better navigation control, faster speeds, limited but sufficient acoustic communication while underwater, entirely on-board data processing, and real-time feedback and reaction to sensed changes in the ocean environment without a human in the loop (no path-planning algorithms or pre-determined paths/waypoints fed to the AUVs by scientists), which makes the AUVs truly autonomous and adaptive. This is, of course, at the cost of the battery duration of the AUVs, which must be recharged much more frequently. Since complex dynamic ocean models are often very large, it is not realistic to run them on board AUVs that must be fully autonomous. Satellite and HF radar images are only useful for detecting plumes with surface expressions, eliminating their usefulness in detection of neutrally buoyant plumes below the top 10 m of water. Thus, we seek to develop a method of plume tracking that can rely solely on the environmental data collected over space and time by the AUVs. The only caveat here is the assumption that a single initial large-scale survey has already been done by an AUV or other sensing platform (or a recently updated regional ocean model has been run) in the region encompassing the plume, such that an approximate plume boundary location at the plume's neutrally buoyant depth is known at the time of AUV deployment. The details of obtaining this initial plume boundary location are beyond the scope of this paper.

As mentioned in Section 3.1, it is useful to approach plume tracking by knowing something about the general dynamics and characteristic scales of the plume as well as any information about its source (for oil leaks or hydrothermal vent sites) or ocean conditions necessary for occurrence (for HABs), and what data values from various sensors might signal that a measurement was taken inside a plume. As mentioned above, since there are many approaches to first detect a plume that are beyond the scope of this paper, we will assume here that the initial 2D boundary of the plume in the horizontal plane has been detected or approximated via satellite imagery, recent oceanographic surveys, or the physics of the region of interest before any AUVs are deployed to track the plume. We will start by concerning ourselves with the horizontal extent of the plume at its neutrally buoyant depth, over a time span shorter than the plume's characteristic time scale (over which the plume boundary displays only minor variations in position). From here, we can sample the plume boundary (defined by a threshold chemical or biological concentration value during field experiments) with varying numbers of AUVs and estimate the plume shape as a sum of Fourier orders.

With an estimation of the location of a plume boundary at a given depth, multiple AUVs (preferably enough to maintain slightly overlapping one-standard deviation spatial-scale range circles along the plume boundary within the plume's characteristic time scale, similar to the range rings in Figure 3) can be deployed within the plume, and an algorithm can be used to assign each AUV a starting position near the estimated plume boundary with approximate equal spacing azimuthally between AUVs about the estimated plume center point. This initial AUV spacing can be written into an IvP Helm "equal azimuth angle" autonomy behavior

that would attempt to maintain equal azimuthal spacing of the AUVs, even as they progress along the plume boundary and the boundary shifts position, adjusting the speed of each AUV to compensate if any one gets too far ahead or falls behind. A second tier of autonomy control will govern the reactions of each AUV to its local environment with a "plume boundary tracking" behavior. This behavior will have a threshold concentration value set for whatever tracer is used to signify levels of chemicals or biological productivity indicative of the plume of interest. The plume boundary tracking behavior will direct the AUV to zigzag horizontally back and forth across the position of this threshold (as it travels azimuthally around the plume center) to maintain an up-to-date position of the local plume boundary. Finally, on a time interval sufficiently small (less than the characteristic time scale of the plume) to average these data over time from each vehicle, each AUV will share its collected plume boundary position data with the other AUVs in the vicinity via acoustic (or RF or satellite) communication, and each vehicle will sort and process the collective data to determine the most current plume boundary position by estimating it as a sum of Fourier orders. Each AUV can then determine if it needs to adjust its speed and big-picture position about the plume edge using the equal azimuth angle behavior. Not only will this method of plume tracking capture the shorter/smaller-scale variations of the plume from one time interval to the next, but also create a continuously evolving track of plume evolution in space and time for a given depth.

With further development to track a plume over longer time scales, we will be able to detect the radial expansion rate of the plume boundary (if any) and its development due to advection, diffusion, and/or biological processes and thus forecast its motion to improve forward-looking mission planning. The best way to develop this plume-tracking process is through simulation, as described in Section 5. Once the simulation is complete, we will be able to initialize implementation of autonomous and adaptive plume tracking with our autonomy architecture by simulating AUVs, (acoustic) communication, and data fusion as described in Section 3 until the plume tracking algorithms and their supporting autonomy behaviors are robust enough for field testing.

5. Plume Simulation Environment

Towards the goal of developing plume-following strategies for AUVs, we must first get a sense of the characteristics of a plume and what the best method is in distributing AUVs about the plume. This requires examining the results and errors associated with reconstructing the shape of a simulated plume from simulated AUV sample points along the plume's edge. Instead of diving into incorporating a more robust or dynamic plume model developed by an outside group, we choose to simulate a very simple plume boundary in horizontal space using Fourier orders (a rough 2D plume approximation) such that we could exactly reconstruct the original plume (again by using Fourier orders) under ideal (though very unrealistic) conditions. This gives us validation that our plume reconstruction algorithms were derived correctly. Though we introduce a few sources of

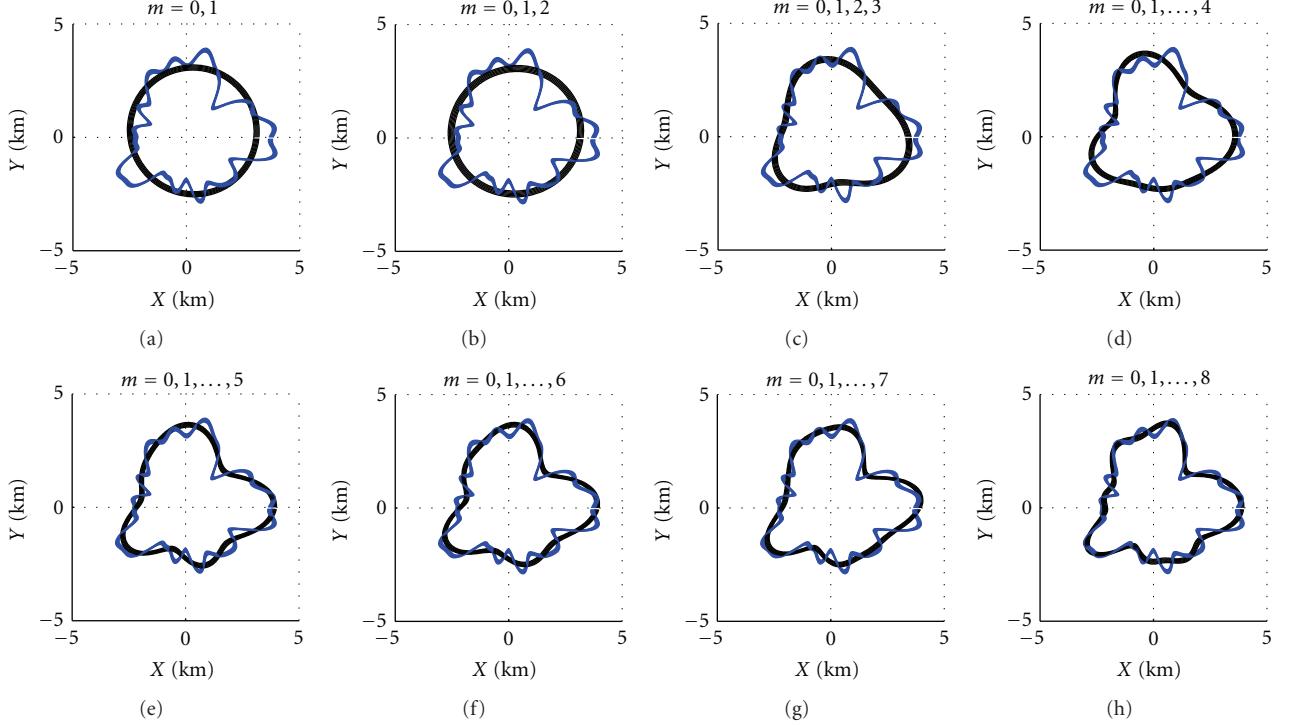


FIGURE 5: A progression of simulated plume shapes (black) of $R = 5$ km, building up to $M_{hi} = 20$ (blue).

plume reconstruction error in the plume simulation and reconstruction process described in this and subsequent sections, we expect to incorporate a more realistic, already developed plume model in the near future such that we are not setting up a situation in which our simulation is doomed to (mostly) succeed. As this is a first-pass simulation experiment, the effects of advection, dispersion, diffusion, holes in the plume shape, multiple plume sources, algal life cycle dynamics, and other complexities that may influence a plume's development over time are beyond the scope of this paper. To test our algorithms and experimental setup over a simulated characteristic plume time scale, we expand the plume in the horizontal plane over a short period of time, sample the plume boundary with varying numbers of AUVs (approximating navigation errors), and then reconstruct the plume from these time-varying sample points. This process is described below.

5.1. Modeling a Plume. A rough estimate of a plume boundary in the horizontal plane is achieved using Fourier orders of the form

$$\Phi_{M_{hi}} = \sum_{m=0}^{M_{hi}} [A_m * \cos(m\theta + \phi_m)] + R, \quad (1)$$

where M_{hi} is the highest Fourier order of the series (here, we will solve for a plume of $M_{hi} = 20$ orders by estimating it with up to 8 Fourier orders from AUV sample points), A_m is the radial amplitude perturbation of the plume boundary for the m th order, ϕ_m is the phase shift of the m th order, and R is the unperturbed radius of the plume. The angles, θ , are

in the range $[0, 2\pi]$ rad about the center of the plume, and $\Phi_{M_{hi}}$ is the radial distance to the edge of the plume from the center at each angle, θ , for a maximum Fourier order, M_{hi} . Generating coefficients A and ϕ at random for each m results in the progression of plume development shown in Figure 5, leading to the overall “actual” plume in Figure 6. We have bounded A_m to $\pm R/2m$, placing the most energy in the lower orders to somewhat realistically represent the amplitude variations of the plume and minimize sharp radial inversions in the boundary shape.

Although it is possible to solve for a very large number of Fourier orders (given enough AUVs over time), this is not computationally efficient and (as seen in Section 5.4) has diminishing returns. Using a sum of many Fourier orders, however, is the most realistic approach (in this simulation) to adding complexity to the simulated plume shape. Time variation (within the characteristic time scale of the simulated plume) is also incorporated into this model, providing more total sample points per AUV (Section 5.2). Over time scales greater than the characteristic time scale of the simulated plume, it is also possible to simulate the development of the plume through turbulent and diffusive processes as well as represent the effect of dominant currents and algae life cycles on the plume shape. Though the effects of long-term time variation have yet to be incorporated into the plume simulation, we describe a means of simulating, detecting, and forecasting basic longer-time-scale radial variations in Section 6.

5.2. Sampling a Plume. First, it is important to backwards-engineer the simulated plume as follows to be sure that the

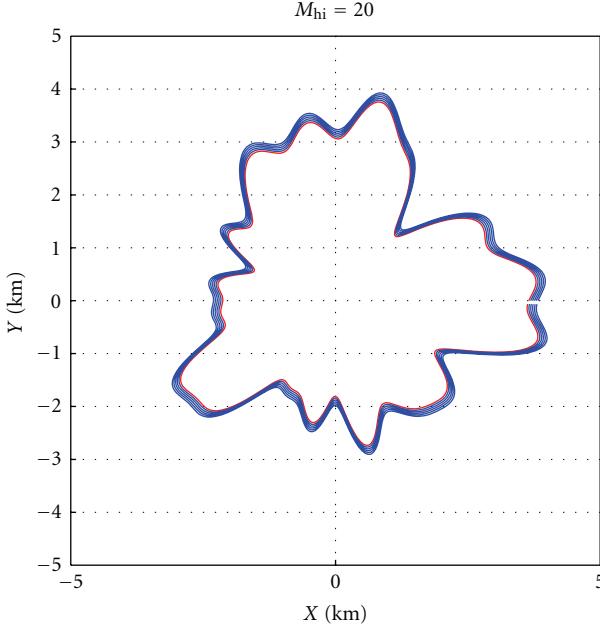


FIGURE 6: The “actual” plume of $M_{hi} = 20$ ($R = 5$ km).

AUV-sampled plume boundary reconstruction algorithms are correct. In a perfect world (with obviously unrealistic assumptions) in which a plume is exactly delineated by a finite sum of Fourier orders and AUVs are evenly spaced around the center of this sharply defined plume at a radius that is on the exact boundary (no navigational error), theory suggests that $2(M_{hi} + 1)$ AUVs are necessary to exactly solve (1) for its $2(M_{hi} + 1)$ unknowns (here, we assume that we can approximate R as the average of all AUV distances radially from the plume center, Φ_{avg}). However, since the 0th order is of constant radius, we incorporate $\cos(\phi_0)$ into A_0 and say $\phi_0 = 0$ rad, reducing the number of unknowns (and AUVs) to $2M_{hi} + 1$. Noise may be added to the angular and radial positions of the AUVs to simulate navigation error and the imperfection in trying to coordinate multiple AUVs spaced at exact angles about a circle, on the exact radius of the plume. Further error will arise from the use of a finite number of AUVs and the necessity of approximating a high order plume with an often relatively low number of Fourier orders calculated from AUV sample points. Since plumes in the ocean and in more robust plume models cannot be fully characterized in closed form as a sum of Fourier orders, error will inherently be added to the AUVs’ Fourier order plume reconstruction when real data or data from a more robust model are used.

Time steps (within the characteristic time scale of the plume boundary position) may be added to increase the number of sample points available, giving $N_{samples} = N_{timesteps} * N_{AUVs}$, and to increase the maximum number of Fourier orders, $M_{AUV,max}$, that can be used to solve for the plume boundary shape. In this implementation, we applied a bounded, random, linear rate of (positive) radial expansion to the amplitude of each Fourier order in the “actual” plume, examining time steps of 2 minutes over a

sufficiently small period of 10 minutes for a plume expanding radially at a rate of up to 0.5 m/s. In real-world applications, this expansion rate is based upon the vertical flow rate from the plume’s source (if present, counteracted somewhat by buoyancy changes with depth) and horizontal spreading (via advection) and diffusion of the plume at the sampled depth [2]. If dealing with a HAB, the life cycle of the algae must also be considered.

5.3. Reconstructing a Plume from AUV Sample Points. Given N_{AUV} AUVs located about the plume boundary at an instant in time, at radii, Φ_{AUV} , at known angles, θ_{AUV} , a fast Fourier transform algorithm, $fft(\bullet)$, is applied to these data to determine the unknown coefficients of the plume with Fourier orders $M \leq \lfloor (N_{AUV} - 1)/2 \rfloor$. The following algorithms are then used to extract out the coefficients:

$$\begin{aligned} R &\approx \Phi_{avg} = \frac{\sum_{\theta_{AUV}} \Phi_{AUV}}{N_{AUV}}, \\ A_{AUV,m=0} &= \frac{1}{2} * \frac{|fft(\Phi_{AUV,m=0} |_{\theta_{AUV}})|}{N_{AUV}/2} - \Phi_{avg}, \\ A_{AUV,m=1:M} &= \frac{|fft(\Phi_{AUV,m=1:M} |_{\theta_{AUV}})|}{N_{AUV}/2}, \\ \phi_{AUV,m=0:M} &= \text{angle}[fft(\Phi_{AUV,m=0:M} |_{\theta_{AUV}})]. \end{aligned} \quad (2)$$

From coefficients $A_{AUV,m}$ and $\phi_{AUV,m}$, we reconstruct the AUV-derived estimation of the plume boundary, $\Phi_{AUV,M}$, as we constructed it in (1):

$$\Phi_{AUV,M} = \sum_{m=0}^M [A_{AUV,m} * \cos(m\theta_{AUV} + \phi_{AUV,m})] + \Phi_{avg}. \quad (3)$$

The reconstructed plume should match the original M_{hi} -order plume exactly (except for numerical round-off error) when all of the following criteria are met:

- (i) $M_{hi} \leq M_{AUV,max} = \lfloor (N_{AUV} - 1)/2 \rfloor$, that is, the maximum Fourier order used to construct the original plume is less than or equal to the maximum Fourier order used to reconstruct it from AUV data (in reality $M_{hi} = \infty$, so this could never be achieved),
- (ii) $\Phi_{avg} = R$,
- (iii) there is no AUV navigation error,
- (iv) there is no time variation,
- (v) all AUVs are evenly spaced about the plume center and exactly on the boundary, and
- (vi) there is instantaneous all-to-all communication of data.

Obviously, some error is introduced when any one of these criteria is not met. If time steps are used to increase the number of sample points, thus increasing $M_{AUV,max}$, N_{AUV} should be replaced by $N_{samples}$ in all equations in this section (Section 5.3), and the spacing of the clustered AUV samples must be interpolated to equal angular spacing about the plume edge to perform the fast Fourier transform (we have used a cubic interpolation function).

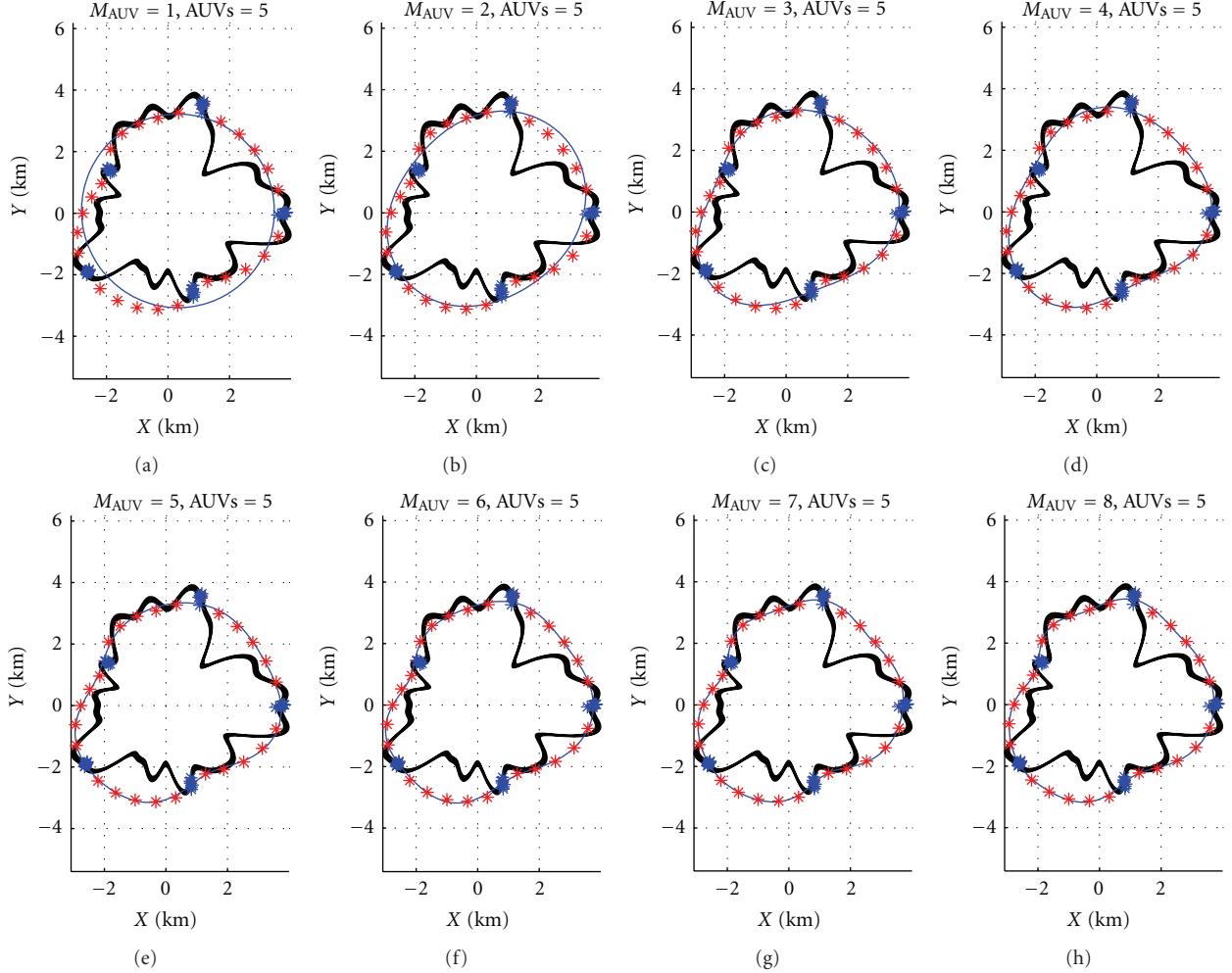


FIGURE 7: Plume estimates (blue line) of $M_{\text{AUV}} = 1, 2, \dots, 8$ for an “actual” plume of $M_{\text{hi}} = 20$ (black lines, time varying), $R = 5$ km, navigation error = 100 m, and time steps = 0, 2, 4, …, 10 min. For the 5 AUVs, the noninterpolated (blue stars) and interpolated (cubic interpolation, red stars) AUV sample points are also shown for reference.

5.4. Results. A set of plume estimates of Fourier orders 1 through 8 are plotted in Figure 7 in contrast to the “actual” time-varying plume. These plots also show the noninterpolated (with navigation error) and interpolated AUV positions. The “actual” plume was chosen to have $M_{\text{hi}} = 20$ to keep the high-frequency variations in boundary radius to a minimum while maintaining more higher-order variation that a reasonable number of AUVs (<10) can exactly resolve. Other numerical assumptions had to be made for the sake of simulation testing and evaluation based on the Bluefin 21” AUVs that our lab group operates and the approximate area and expansion rate of a mesoscale plume (similar to that of the Deepwater Horizon disaster in the Gulf of Mexico in 2010 [1]). Specifically, we take $R = 5$ km, AUV navigation error = 100 m, and time steps = 0, 2, 4, …, 10 min within the characteristic time scale of plume evolution.

A set of Monte Carlo simulations was used to quantify the overall mean percent error in the model based on the number of Fourier orders solved for, varying the number of AUVs while keeping the time steps consistent over all trials. This is

accomplished by comparing the boundary of the estimated plume to the time-averaged boundary of the actual plume as follows:

$$\% \text{Error}_{\text{plume}} = \frac{|\Phi_{\text{estimated}} - \Phi_{\text{actual,time-avg}}|}{\Phi_{\text{actual,time-avg}}}. \quad (4)$$

These results are shown in Figure 8 for each set of Fourier orders, with $M_{\text{AUV,max}}$ determined by N_{samples} .

It is interesting to note that, for a fixed number of AUVs, the general trend appears to be an exponential decrease in error as a larger number of Fourier orders is solved for. However, upon closer examination of the error values, the order of lowest error is approximately $M_{\text{AUV,min_error}} = 2N_{\text{AUV}}$. This result will help minimize the error while reasonably limiting the amount of data processing necessary to estimate the plume boundary. Alternately, for a given Fourier order M_{AUV} , as the number of AUVs increases, the percent error decreases, as is expected.

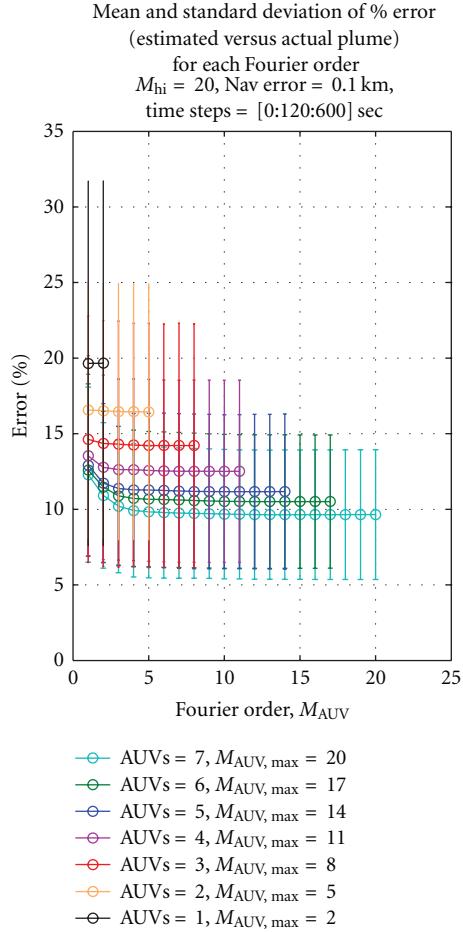


FIGURE 8: Percent error in plume radius between the estimated and actual (time-averaged) plumes, averaged over 500 trials. Mean values are shown for each maximum Fourier order, with error bars showing ± 1 standard deviation.

6. Forecasting Long-Term Variations

Having simulated and analyzed a plume over a short time span, we will now explore expanding the plume simulation to longer time spans to enable plume shape forecasting. There are two formulations here for the basic time expansion approximation. We may either assume that the plume expands linearly in time in the radial direction, with a constant coefficient of expansion, $d\Phi/dt$ (5), or that both the amplitude and phase coefficients change linearly in time, with constant coefficients dA/dt and $d\phi/dt$ (6). These are the most simplistic cases, which may be built upon in the future into nonlinear coefficients to account for further complexities from real ocean dynamics:

$$\Phi_{M_{hi}}(\theta, t) = \sum_{m=0}^{M_{hi}} [A_m * \cos(m\theta + \phi_m)] + \frac{d\Phi}{dt} * (t - t_0) + R, \quad (5)$$

$$\begin{aligned} \Phi_{M_{hi}}(\theta, t) &= \sum_{m=0}^{M_{hi}} \left[\left(A_m + \frac{dA}{dt} * (t - t_0) \right) \right. \\ &\quad \left. * \cos \left(m\theta + \left(\phi_m + \frac{d\phi}{dt} * (t - t_0) \right) \right) \right] + R. \end{aligned} \quad (6)$$

Assuming one of the above plume formulations and sampling it with AUVs over a number of large time steps, we can determine the differences in overall plume shape from one point in time to the next and back out the constant coefficients from there. If the formulation in (5) is assumed, we may simply find the mean difference (over all θ s) in radius between the estimated plumes at times t_0 and t_1 , as shown in (7):

$$\frac{d\Phi}{dt} \approx \text{mean} \left[\frac{\Phi_{AUV,M}(\theta, t_1) - \Phi_{AUV,M}(\theta, t_0)}{t_1 - t_0} \right]. \quad (7)$$

Solving (6) for formulation coefficients is more complex. Given estimated plumes from AUVs at times t_0 and t_1 sufficiently far apart in time, we must maximize the correlation between $\Phi_{AUV,M}(\theta, t_1)$ and $\Phi_{AUV,M}(\theta, t_0)$ over radius and azimuth angle. The tool for this will be a matched filter applied to $\Phi_{AUV,M}(\theta, t_1)$ and $\Phi_{AUV,M}(\theta, t_0)$, allowing us to back out the constant coefficients once we determine the phase and amplitude changes between t_0 and t_1 . Repeating either of the above processes over multiple time steps will further improve the accuracy of the coefficients.

Once we solve for the constant coefficients using either of the above methods, a forecast can be made for the plume shape by simply applying the linear changes to the estimated plume shape at the last known time slice and projecting it forward to the next time step(s). As with any forecasting, however, the accuracy of the forecast decreases with time steps further into the future. A weighting function (potentially the right side of a Gaussian) should be included with the forecast to account for this.

7. Looking Ahead

It is important to take what we have learned from this exercise and apply it to a more robust plume simulation, such as a theory- and data-derived dynamic plume model, as well as to prepare for taking this application into the field. Following the first iteration of this plume simulation, the next step is to use the plume estimated by the AUVs over progressive time steps to estimate the linear time perturbation coefficients of each Fourier order and use these coefficients for future prediction.

Jumping ahead to prepare for realistic implementation of plume tracking in the field, we plan to use our IvP Helm and Goby autonomy to move the AUVs along the actual plume boundary (in “follow-the-leader” fashion) as described in Section 4.2, autonomously adapting their tracks to their real-time measurements by zigzagging across the boundary, and keeping their angular spacing relatively constant. As AUVs travel along the boundary azimuthally, all at the same

speed, the radial excursions in the boundary may cause the azimuthal spacing of adjacent AUVs to degrade. To counter this effect, we will employ autonomy behaviors to change speed and maintain azimuthal distribution when a significant degradation in spacing is detected. This will first be implemented in simulation to work out any bugs before taking it into the field with the AUVs.

Other features to add to the plume simulation will account for the effects of advection by currents and turbulent diffusion. A good estimation for diffusion, used widely in the underwater community, is Fick's Law [17], and examples of current effects can be found in [1, 9]. These effects may be best incorporated into the simulation as time-varying coefficients similar to those in Section 6, only nonlinear in time and space. Another option would be to take advantage of a commercial computational fluids simulator to simulate these effects. It will also be important to account for the direction of motion of a plume, as this may or may not cause the leading edge of the plume to be more distinct than the trailing edge. Again, however, we do not want to reinvent the wheel and may prefer to research and take advantage of already existing plume models and data that account for some of these effects with greater detail and accuracy than achievable by the above method. In the case of the evolution of HABs, we must also account for life cycle evolution of the algae, and testing with historical data of algal bloom evolution would be useful here [15].

Finally, it is important to gain a knowledge of how each source of error (i.e., navigation error, higher modes and sharper inversions in plume shape, overall plume radius, cubic interpolation of AUV spacing about the plume, etc.) affects the overall error in the estimated plume boundary. Such an error review will require a wide range of tests, changing only one variable at a time. The cubic interpolation of AUV position alone will be evaluated against other interpolation techniques, such as the Lomb-Scargle method [18], to minimize errors.

8. Conclusion

This paper provides a conceptual outline of the requirements for implementing adaptive, autonomous plume tracking using a network of AUVs, including a first-pass simulation of detecting and reconstructing plume shapes solely from AUV sample points, with the example of a plume of oil originating from the sea floor. Using a sum of M_{hi} Fourier orders to represent a plume shape at its neutrally buoyant depth, we added noise in the AUV positions to represent navigation error. We also incorporated linear radial expansion of the plume over time to simulate plume spreading due to the continuous influx of oil. Reconstruction of the plume from the time-varying AUV samples was seen to result in errors in the estimated versus original plume shapes ranging from 9 to 20% (for 1 through 7 AUVs, $M_{hi} = 20$, $R = 5$ km, navigation error = 100 m, and time steps = 0, 2, 4, ..., 10 min), largely decreasing with an increase in the number of Fourier orders being solved for, keeping the number of AUVs, navigation errors, and time steps constant. The errors also decrease as the number of AUVs is increased. With this knowledge and

technology, we will be able to improve the plume simulation further based on the physics of plume spreading via currents and diffusion and employ adaptive autonomy behaviors with the AUVs to progress them along the plume boundary. In the end, the plume tracking process presented here will provide a synoptic data set describing the plume based on the spatiotemporal scales of the feature, using a network of AUVs to prevent data aliasing.

Acknowledgments

This research was made with Government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. The authors would also like to acknowledge the NATO Undersea Research Centre (NURC) in La Spezia, Italy for all their help in organizing and conducting field trials (GLINT '09, GLINT '10) to test AUV feature tracking behaviors as well as for use of their marine assets that allowed us to build up to this research. They would like to acknowledge the US Office of Naval Research for funding portions of the necessary travel, underlying research, and sea trials. As well as the Naval Undersea Warfare Center Division, Newport (Code 25)—Technical Support and Logistics, and the Office of Naval Research TechSolutions Program Office (Lightweight NSW UUV program)—Technical Support for funding Champlain '09 and allowing us to use the Iver AUV for thermocline tracking experiments. Finally, the authors would like to thank Toby Schneider and Donald Eickstedt for numerous discussions and feedback about this and related research as well as Pierre Lermusiaux's MSEAS group at MIT for supplying ocean models, and the rest of the LAMSS group at MIT for their support of these efforts.

References

- [1] R. Camilli, C. M. Reddy, D. R. Yoerger et al., "Tracking hydrocarbon plume transport and biodegradation at deepwater horizon," *Science*, vol. 330, no. 6001, pp. 201–204, 2010.
- [2] J. S. Turner, *Buoyancy Effects in Fluids*, Cambridge Monographs on Mechanics and Applied Mathematics, Cambridge University Press, Cambridge, UK, 1973.
- [3] J. S. Willcox, J. G. Bellingham, Y. Zhang, and A. B. Baggeroer, "Performance metrics for oceanographic surveys with autonomous underwater vehicles," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 711–725, 2001.
- [4] Y. Zhang, A. B. Baggeroer, and J. G. Bellingham, "Spectral-feature classification of oceanographic processes using an autonomous underwater vehicle," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 726–741, 2001.
- [5] T. Schneider and H. Schmidt, "The dynamic compact control language: a compact marshalling scheme for acoustic communications," in *Proceedings of the IEEE Oceans Conference*, Sydney, Australia, May 2010.
- [6] Goby Developers, "Goby underwater autonomy project documentation," <http://gobysoft.com/doc/1.0>.
- [7] M. R. Benjamin, H. Schmidt, P. M. Newman, and J. J. Leonard, "Nested autonomy for unmanned marine vehicles

- with MOOS-IvP,” *Journal of Field Robotics*, vol. 27, no. 6, pp. 834–875, 2010.
- [8] M. R. Benjamin, J. J. Leonard, H. Schmidt, and P. M. Newman, “An overview of MOOS-IvP and a brief users guide to the IvP Helm autonomy software,” Tech. Rep. MIT-CSAIL-TR-2009-028, 2009, <http://hdl.handle.net/1721.1/45569>.
 - [9] M. Jakuba, D. Yoerger, A. Bradley, C. German, C. Langmuir, and T. Shank, “Multiscale, multimodal AUV surveys for hydrothermal vent localization,” in *Proceedings of the Fourteenth International Symposium on Unmanned Untethered Submersible Technology (UUST ’05)*, Durham, NH, USA, 2005.
 - [10] Woods Hole Oceanographic Institution, “WHOI acoustic communications: micro-modem overview,” 2011, <http://acomms.whoi.edu/umodem/>.
 - [11] M. Martin and H. Moravec, “Robot evidence grids,” Tech. Rep. CMURI-TR-96-06, The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pa, USA, 1996.
 - [12] D. Wang, P. F. J. Lermusiaux, P. J. Haley, D. Eickstedt, W. G. Leslie, and H. Schmidt, “Acoustically focused adaptive sampling and on-board routing for marine rapid environmental assessment,” *Journal of Marine Systems*, vol. 78, supplement 1, pp. S393–S407, 2009.
 - [13] S. Petillo, A. Balasuriya, and H. Schmidt, “Autonomous adaptive environmental assessment and feature tracking via autonomous underwater vehicles,” in *Proceedings of the IEEE Oceans Conference*, Sydney, Australia, May 2010.
 - [14] R. N. Smith, Y. Chao, P. P. Li, D. A. Caron, B. H. Jones, and G. S. Sukhatme, “Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model,” *International Journal of Robotics Research*, vol. 29, no. 12, pp. 1475–1497, 2010.
 - [15] J. Das, K. Rajan, S. Frolov et al., “Towards marine bloom trajectory prediction for AUV mission planning,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA ’10)*, pp. 4784–4790, May 2010.
 - [16] J. Das, F. Py, T. Maughan et al., “Simultaneous tracking and sampling of dynamic oceanographic features with autonomous underwater vehicles and Lagrangian drifters,” in *Proceedings of the 12th International Symposium on Experimental Robotics*, New Delhi, India, December 2010.
 - [17] “Oceanic diffusion and mixing,” <http://flux.ocean.washington.edu/WQ2009/oc515.mixing.pdf>.
 - [18] W. H. Press and G. B. Rybicki, “Fast algorithm for spectral analysis of unevenly sampled data,” *Astrophysical Journal*, vol. 338, pp. 277–280, 1989.