



Computer Science and Information Systems

Published by ComSIS Consortium

**Special Issue on Cyber-Physical
Networks and Software**

Volume 8, Number 4
October 2011

ComSIS Consortium:

University of Belgrade:

Faculty of Organizational Science, Belgrade, Serbia
Faculty of Mathematics, Belgrade, Serbia
School of Electrical Engineering, Belgrade, Serbia

Serbian Academy of Science and Art:

Mathematical Institute, Belgrade, Serbia

Union University:

School of Computing, Belgrade, Serbia

University of Novi Sad:

Faculty of Sciences, Novi Sad, Serbia
Faculty of Technical Sciences, Novi Sad, Serbia
Faculty of Economics, Subotica, Serbia
Technical Faculty "M. Pupin", Zrenjanin, Serbia

University of Montenegro:

Faculty of Economics, Podgorica, Montenegro
Faculty of Mathematics and Science, Podgorica, Montenegro

EDITORIAL BOARD:

Editor-in-Chief: Mirjana Ivanović, University of Novi Sad

Vice Editor-in-Chief: Ivan Luković, University of Novi Sad

Managing editor: Zoran Putnik, University of Novi Sad

Managing editor: Miloš Radovanović, University of Novi Sad

Managing editor: Gordana Rakić, University of Novi Sad

Editorial Assistant: Vladimir Kurbalija, University of Novi Sad

Editorial Assistant: Jovana Vidaković, University of Novi Sad

Editorial Assistant: Ivan Pribela, University of Novi Sad

Editorial Assistant: Slavica Aleksić, University of Novi Sad

Editorial Assistant: Srđan Škrbić, University of Novi Sad

Associate editors:

P. Andrae, Victoria University, New Zealand

D. Bojić, University of Belgrade, Serbia

D. Bečejski-Vujaklija, University of Belgrade, Serbia

I. Berković, University of Novi Sad, Serbia

G. Bhutkar, Vishwakarma Institute of Technology, India

L. Böszörményi, University of Clagenfurt, Austria

K. Bothe, Humboldt University of Berlin, Germany

Z. Budimac, University of Novi Sad, Serbia

G. Devedžić, University of Kragujevac, Serbia

V. Devedžić, University of Belgrade, Serbia

J. Đurković, University of Novi Sad, Serbia

S. Guttormsen Schar, ETH Zentrum, Switzerland

P. Hansen, University of Montreal, Canada

L. C. Jain, University of South Australia, Australia

V. Jovanović, Georgia Southern University, USA

Lj. Kaščelan, University of Montenegro, Montenegro

P. Mahanti, University of New Brunswick, Canada

M. Maleković, University of Zagreb, Croatia

N. Mitić, University of Belgrade, Serbia

A. Mitrović, University of Canterbury, New Zealand

N. Mladenović, Serbian Academy of Science, Serbia

P. Mogin, Victoria University, New Zealand

S. Mrdalj, Eastern Michigan University, USA

G. Nenadić, University of Manchester, UK

Z. Ognjanović, Serbian Academy of Science, Serbia

A. Pakstas, London Metropolitan University, England

P. Pardalos, University of Florida, USA

M. Racković, University of Novi Sad, Serbia

B. Radulović, University of Novi Sad, Serbia

M. Stanković, University of Niš, Serbia

D. Tošić, University of Belgrade, Serbia

J. Trninić, University of Novi Sad, Serbia

J. Woodcock, University of Kent, England

EDITORIAL COUNCIL:

S. Ambroszkiewicz, Polish Academy of Science, Poland

Z. Arsovski, University of Kragujevac, Serbia

D. Banković, University of Kragujevac, Serbia

T. Bell, University of Cantrebury, New Zealand

S. Bošnjak, University of Novi Sad, Serbia

Ž. Branović, University of Novi Sad, Serbia

H. D. Burkhard, Humboldt University of Berlin, Germany

B. Chandrasekaran, Ohio State University, USA

V. Čirić, University of Belgrade, Serbia

D. Domazet, Faculty of Information Technology,
Belgrade, Serbia

S. Đorđević-Kajan, University of Niš, Serbia

P. Hotomski, University of Novi Sad, Serbia

Z. Jovanović, University of Belgrade, Serbia

L. Kalinichenko, Russian Academy of Science, Russia

Z. Konjović, University of Novi Sad, Serbia

I. Koskosas, University of Western Macedonia, Greece

W. Lamersdorf, University of Hamburg, Germany

T. C. Lethbridge, University of Ottawa, Canada

A. Lojpur, University of Montenegro, Montenegro

Y. Manolopoulos, Aristotle University, Greece

A. Mishra, Atılım University, Turkey

S. Mishra, Atılım University, Turkey

J. Protić, University of Belgrade, Serbia

D. Simpson, University of Brighton, England

D. Starčević, University of Belgrade, Serbia

S. Stojanov, Faculty for Informatics, Plovdiv, Bulgaria

D. Surla, University of Novi Sad, Serbia

M. Tuba, University of Belgrade, Serbia

P. Tumbas, University of Novi Sad, Serbia

P. Zarate, IIRIT-INPT, Toulouse, France

K. Zdravkova, University of Ss. Cyril and Methodius,
Skopje, FYR of Macedonia

ComSIS Editorial office:

**University of Novi Sad, Faculty of Sciences,
Department of Mathematics and Informatics**

Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia

Phone: +381 21 458 888; **Fax:** +381 21 6350 458

www.comsis.org; Email: comsis@uns.ac.rs

**Volume 8, Number 4, 2011
Novi Sad**

Computer Science and Information Systems

Special Issue on Cyber-Physical Networks and Software

ISSN: 1820-0214

ComSIS Journal is sponsored by:

Ministry of Education and Science of Republic of Serbia - <http://www.mpn.gov.rs/>



Computer Science and Information Systems

AIMS AND SCOPE

Computer Science and Information Systems (ComSIS) is an international refereed journal, published in Serbia. The objective of ComSIS is to communicate important research and development results in the areas of computer science, software engineering, and information systems.

We publish original papers of lasting value covering both theoretical foundations of computer science and commercial, industrial, or educational aspects that provide new insights into design and implementation of software and information systems. ComSIS also welcomes surveys papers that contribute to the understanding of emerging and important fields of computer science. Regular columns of the journal cover reviews of newly published books, presentations of selected PhD and master theses, as well as information on forthcoming professional meetings. In addition to wide-scope regular issues, ComSIS also includes special issues covering specific topics in all areas of computer science and information systems.

ComSIS publishes invited and regular papers in English. Papers that pass a strict reviewing procedure are accepted for publishing. ComSIS is published semiannually.

Indexing Information

ComSIS is covered or selected for coverage in the following:

- Science Citation Index (also known as SciSearch) and Journal Citation Reports / Science Edition by Thomson Reuters, with 2010 two-year impact factor 0.324,
- Computer Science Bibliography, University of Trier (DBLP),
- EMBASE (Elsevier),
- Scopus (Elsevier),
- Summon (Serials Solutions),
- EBSCO bibliographic databases,
- IET bibliographic database Inspec,
- FIZ Karlsruhe bibliographic database io-port,
- Index of Information Systems Journals (Deakin University, Australia),
- Directory of Open Access Journals (DOAJ),
- Google Scholar,
- Center for Evaluation in Education and Science and Ministry of Science of Republic of Serbia (CEON) in cooperation with the National Library of Serbia,
- Serbian Citation Index (SCIIndeks),
- doiSerbia.

Information for Contributors

The Editors will be pleased to receive contributions from all parts of the world. An electronic version (MS Word or LaTeX), or three hard-copies of the manuscript written in English, intended for publication and prepared as described in "Manuscript Requirements" (which may be downloaded from <http://www.comsis.org>), along with a cover letter containing the corresponding author's details should be sent to official Journal e-mail.

Criteria for Acceptance

Criteria for acceptance will be appropriateness to the field of Journal, as described in the Aims and Scope, taking into account the merit of the content and presentation. The number of pages of submitted articles is limited to 30 (using the appropriate Word or LaTeX template).

Manuscripts will be refereed in the manner customary with scientific journals before being accepted for publication.

Copyright and Use Agreement

All authors are requested to sign the "Transfer of Copyright" agreement before the paper may be published. The copyright transfer covers the exclusive rights to reproduce and distribute the paper, including reprints, photographic reproductions, microform, electronic form, or any other reproductions of similar nature and translations. Authors are responsible for obtaining from the copyright holder permission to reproduce the paper or any part of it, for which copyright exists.

Computer Science and Information Systems

Volume 8, Number 4, Special Issue, October 2011

CONTENTS

Editorial

Papers

- 953 An Improved Node Localization Algorithm Based on DV-Hop for Wireless Sensor Networks**
Qingji Qian, Xuanjing Shen, Haipeng Chen
- 973 Energy-Efficient Opportunistic Localization with Indoor Wireless Sensor Networks**
Feng Xia, Xue Yang, Haifeng Liu, Da Zhang, Wenhong Zhao
- 991 An Energy Efficient and Load Balancing Routing Algorithm for Wireless Sensor Networks**
Jin Wang, Tinghuai Ma, Jinsung Cho, Sungoung Lee
- 1009 A Redistribution Method to Conserve Data in Isolated Energy-harvesting Sensor Networks**
Peng Liu, Song Zhang, Jian Qiu, Xingfa Shen, Jianhui Zhang
- 1027 A Packet Buffer Evaluation Method Exploiting Queueing Theory for Wireless Sensor Networks**
Tie Qiu, Lin Feng, Feng Xia, Guowei Wu, Yu Zhou
- 1051 On the Efficiency of Cluster-based Approaches for Motion Detection using Body Sensor Networks**
Kun-chan Lan, Chien-Ming Chou, Tzu-nung Wang, Mei-Wen Li
- 1073 Optimization of Multiple Gateway Deployment for Underwater Acoustic Sensor Networks**
Jugen Nie, Deshi Li, Yanyan Han
- 1097 A Distributed Power Management Design Based on MOST Networks**
Yushan Jin, Ruikai Liu, Xingran He, Yongping Huang
- 1117 An Energy-Efficient Localization Strategy for Smartphones**
Haifeng Liu, Feng Xia, Zhuo Yang, Yang Cao
- 1129 Modeling Disease Spreading on Complex Networks**
Xiangjie Kong, Yu Qi, Xiumiao Song, Guojiang Shen
- 1143 An Improved Spectral Clustering Algorithm Based on Local Neighbors in Kernel Space**
Xinyue Liu, Xing Yong, Hongfei Lin

- 1159 A Novel Capacity and Trust Based Service Selection Mechanism for Collaborative Decision Making in CPS**
Bo Zhang, Yang Xiang, Peng Wang, Zhenhua Huang
- 1185 Privacy Preserving in Ubiquitous Computing: Classification & Hierarchy**
Tinghuai Ma, Sen Yan, Jin Wang, Sungyoung Lee
- 1207 TRM-IoT: A Trust Management Model Based on Fuzzy Reputation for Internet of Things**
Dong Chen, Guiran Chang, Dawei Sun, Jiajia Li, Jie Jia, Xingwei Wang
- 1229 A Reusable Agent Design Pattern with Flexibility and Extensibility**
Haolan Zhang, Wenhua Zeng, Christian Van der Velden
- 1251 Quantitative Analysis for Symbolic Heap Bounds of CPS Software**
Renjian Li, Ji Wang, Liqian Chen, Wanwei Liu, Dengping Wei
- 1277 A Model-Based Software Development Method for Automotive Cyber-Physical Systems**
Zhigang Gao, Haixia Xia, Guojun Dai
- 1303 Velocity Adaptation for Synchronizing a Mobile Agent Network**
Lijing Li, Hui Yan, Hui Li, Chunxi Zhang
- 1317 A Case Study on REST-Style Architecture for Cyber-Physical Systems: Restful Smart Gateway**
Qiang Li, Weijun Qin, Bing Han, Ruicong Wang, Limin Sun

EDITORIAL

There is a strong trend towards networked cyber-physical systems that bridge the virtual world of networking and computing and the real world. One of the major characteristics of cyber-physical systems is the tight integration of computation (more specifically, software) and physical objects through networks. In a typical cyber-physical system, various components are networked to sense, monitor and control the physical world. Cyber-physical systems promise to improve the quality of our daily lives by enabling innovative services and applications. To realize the potential of cyber-physical applications, high-confidence networks and software become essential.

In this ComSIS special issue on “Cyber-Physical Networks and Software”, we are focused on the latest achievements that address key issues and topics related to design and applications of cyber-physical networks and software. A collection of 19 papers is selected from 79 open submissions through a rigorous peer-review process. The selection provides a glimpse of the state-of-the-art research in the field.

In the paper “An Improved Node Localization Algorithm Based on DV-Hop for Wireless Sensor Networks”, Qian et al. propose a novel localization algorithm called NDV-Hop_Bon based on the popular DV-Hop localization algorithm for wireless sensor networks (WSNs). The proposed algorithm selects the reference nodes of a certain number to position the unknown nodes based on different environments. The simulation results show that the positioning accuracy is improved.

The paper “Energy-Efficient Opportunistic Localization with Indoor Wireless Sensor Networks” proposes an Energy-Efficient Opportunistic Localization (EEOL) scheme to satisfy the requirements of both positional accuracy and power consumption. The idea of opportunistic wake-up probability is exploited to wake up an appropriate numbers of sensor nodes, while ensuring the high positional accuracy. Through utilizing this method, the number of active sensors in the sensing range of the user is decreased, and the power consumption is significantly reduced.

In the paper “An Energy Efficient and Load Balancing Routing Algorithm for Wireless Sensor Networks”, Wang et al. propose a Ring-based Energy Aware Routing (REAR) algorithm for WSNs which can achieve both energy balancing and energy efficiency for all sensor nodes. The algorithm considers not only the hop number and distance but also the residual energy of the next hop node during the routing process.

The purpose of the paper “A Redistribution Method to Conserve Data in Isolated Energy-harvesting Sensor Networks” is to design a solution for fair data storage under space and energy limitation only based on local information. The authors propose a heuristic Distributed Energy-aware Data Conservation method (DEDC), which considers following two issues: i) where to store data with respect to energy and space storage, and ii) how to prioritize the transmission of important data.

In the paper “A Packet Buffer Evaluation Method Exploiting Queueing Theory for Wireless Sensor Networks”, Qiu et al. present a new evaluation method for packet buffer capacity of wireless sensor nodes using queueing network model, whose packet buffer capacity is analyzed for each type node, when it is in the best working condition. The authors establish an M/M/1/N-type queueing network model with holding nodes for WSNs and design approximate iterative algorithms.

In the paper “On the Efficiency of Cluster-based Approaches for Motion Detection using Body Sensor Networks”, Lan et al. first discuss the efficiency of cluster-based approaches for saving energy, and then propose a novel cluster head selection algorithm to maximize the lifetime of a body sensor network for motion detection.

In the paper “Optimization of Multiple Gateway Deployment for Underwater Acoustic Sensor Networks” by Nie et al, the deployment of surface gateways in underwater acoustic sensor networks is studied by considering the acoustic characteristics. The authors propose an optimization method of surface gateways deployment dynamically based on a genetic algorithm, design a novel transmission mechanism – simultaneous transmission, and realize two efficient routing algorithms that achieve minimal delay and payload balance among sensor nodes.

In the paper “A Distributed Power Management Design Based on MOST Networks”, a distributed power management solution is designed for MOST (Media Oriented Systems Transport) networks, in which the slave nodes can sleep independently and the master node manages the network state, and new wake-up mechanisms in the sleep state are proposed.

Liu et al. present an energy-efficient localization strategy for smartphone applications in the paper “An Energy-Efficient Localization Strategy for Smartphones”. On one hand, the strategy can dynamically estimate the next localization time point to avoid unnecessary localization operations. On the other hand, it can also automatically select the energy-optimal localization method. The authors evaluate the strategy through a series of simulations.

Based on a complex network approach, a contact network model with scale-free property is built in the paper “Modeling Disease Spreading on Complex Networks” by Kong et al. By analyzing the fact data of H1N1 influenza

provided by the Beijing Health Bureau, a contact tracing mechanism is used to research H1N1 virus transmission dynamics with this model. Furthermore, the contact tracing coefficient and random checking coefficient are studied to analyze their impact on the peak value of new infections and cumulative number of infections.

In the paper “An Improved Spectral Clustering Algorithm Based on Local Neighbors in Kernel Space”, a novel spectral clustering method is proposed based on local neighborhood in kernel space (SC-LNK), which assumes that each data point can be linearly reconstructed from its neighbors. The SC-LNK algorithm tries to project the data to a feature space by the Mercer kernel, and then learn a sparse matrix using linear reconstruction as the similarity graph for spectral clustering.

In the paper “A Novel Capacity and Trust Based Service Selection Mechanism for Collaborative Decision Making in CPS”, Zhang et al. propose a novel capacity and trust computation based cyber-physical systems (CPS) service selection mechanism in intelligent and automatic manners. The mechanism comprises three phases, including capacity evaluation, trust computation and negotiation selection.

In the paper “Privacy Preserving in Ubiquitous Computing: Classification & Hierarchy”, Ma et al. present a privacy-preserving architecture utilizing the classification of personal information and hierarchy of services, which are derived from the concept of Class in the Object Orient Programming. In a sense, the authors strike a balance between two goals of Ubiquitous Computing: interaction and privacy preserving.

In the paper “TRM-IoT: A Trust Management Model Based on Fuzzy Reputation for Internet of Things”, Chen et al. present a trust and reputation model TRM-IoT to enforce the cooperation between things in a network of IoT/CPS based on their behaviors. The accuracy, robustness and lightness of the proposed model is validated through a wide set of simulations.

In the paper “A Reusable Agent Design Pattern with Flexibility and Extensibility”, Zhang et al. introduce a novel design for agent-based systems, which is able to provide an efficient design pattern for improving the reusability, extensibility and flexibility of agent design. The novel agent capability design offers an open and flexible structure, and implements several practical algorithms that can improve the system performance.

In the paper “Quantitative Analysis for Symbolic Heap Bounds of CPS Software”, Li et al. present a framework for statically analyzing symbolic heap bounds of CPS software. A novel list abstraction method is also proposed, which maintains precise shape properties and quantitative properties. The authors build a prototype tool that can analyze the heap bounds automatically.

The development of automotive CPS software needs to consider not only functional requirements, but also non-functional requirements and the interaction with physical environments. In the paper “A Model-Based Software Development Method for Automotive Cyber-Physical Systems”, a model-based software development method for automotive CPS (MoBDAC) is presented. The authors illustrate the development workflow of MoBDAC by an example of a power window development.

The paper “Velocity Adaptation for Synchronizing a Mobile Agent Network” investigates the problem of synchronizing a mobile agent network by means of a velocity adaptation strategy, where each agent is assigned different moving velocities to establish a time-varying network topology, and the velocity of each agent develops adaptively according to the local property between itself and its neighbors.

Li et al. present a REST-style architecture for CPS in the paper “A Case Study on REST-Style Architecture for Cyber-Physical Systems: Restful Smart Gateway”. The authors propose a path towards solving requirements of CPS architecture through Restful principles. A prototyping system called the restful smart gateway is built, which seamlessly integrates conceptual and physical resources into the Web.

It has been a great pleasure to run this special issue, which reveals important research results in the field of Cyber-Physical Networks and Software. I would like to thank Prof. Mirjana Ivanović, Editor-in-Chief of ComSIS, and other staff in the Editorial Office for giving me the opportunity to organize this special issue and for their great help in the organization of this issue. I thank all authors for their submissions and all reviewers for their diligent work in evaluating these submissions. I sincerely hope that you enjoy reading these distinguished papers.

Guest Editor
Feng Xia
School of Software, Dalian University of Technology
Dalian 116620, China
f.xia@ieee.org

Editor-in-Chief
Mirjana Ivanović

An Improved Node Localization Algorithm Based on DV-Hop for Wireless Sensor Networks

Qingji Qian¹, Xuanjing Shen^{2,3}, and Haipeng Chen^{2,3}

¹ College of Physics, Jilin University,
130022 Changchun, China
qianqj@jlu.edu.cn

² College of Computer Science and Technology, Jilin University,
130012 Changchun, China

³ Key Laboratory of Symbolic Computation and Knowledge Engineering,
Ministry of Education, Jilin University, 130012 Changchun, China
{xjshen,chenhp}@jlu.edu.cn

Abstract. Sensor node localization is the basis for the entire wireless sensor networks. Because of restricted energy of the sensor nodes, the location error, costs of communication and computation should be considered in localization algorithms. DV-Hop localization algorithm is a typical positioning algorithm that has nothing to do with distance. In the isotropic dense network, DV-Hop can achieve position more precisely, but in the random distribution network, the node location error is great. This paper summed up the main causes of error based on the analysis on the process of the DV-Hop algorithm, aimed at the impact to the location error which is brought by the anchor nodes of different position and different quantity, a novel localization algorithm called NDV-Hop_Bon (New DV-Hop based on optimal nodes) was put forward based on optimal nodes, and it was simulated on Matlab. The results show that the new proposed location algorithm has a higher accuracy on localization with a smaller communication radius in the circumstances, and it has a wider range of applications.

Keywords: Wireless sensor networks, Node localization, DV-Hop Algorithm, Optimal nodes

1. Introduction

In the application of wireless sensor networks, the sensor node location information is critical to the monitoring activities of sensor networks. The location of the incident or the node position of the obtained information is the important information that must be included in the sensor nodes monitor messages. It is meaningless for the monitor messages of no place messages. GPS (Global Positioning System, GPS) is the most accurate and most perfect positioning technology, which has high accuracy, strong anti-interference, and can locate in time, etc. But if we use GPS to position, the cost of the node is two times higher in terms of magnitude than the ordinary nodes. That is to

say, when the nodes of 10% are equipped with GPS, the cost of the entire network will increase 10 times around. At the same time, for sensor nodes using battery-powered, GPS equipment with high energy consumption is not more suitable for extensive application in sensor nodes with limited energy. So, generally, it does not use GPS to position all the nodes, but through some nodes of the known location to locate other nodes.

In recent years, research on localization algorithm of wireless sensor networks has achieved fruitful results [1-4]. According to whether to measure the actual distance between nodes in the process of position, the localization algorithm of wireless sensor networks can be divided into two parts: range-based localization algorithm and range-free localization algorithm. The range-based localization algorithm need to measure distance of nodes or angle information, and some algorithms may also need precise clock synchronization. Therefore, there is a high requirement for hardware of the wireless sensor networks nodes, and the algorithm in some applications is also greatly influenced by the surrounding environment, which has limited the application of such methods to some extent. Compared with the range-based localization algorithm, the range-free location algorithm has the advantages of low cost, small power consumption, simple hardware and robust to noise and so on, and can provide acceptable accuracy in positioning. So people are very concerned about it in recent years.

The range-free localization algorithm can achieve localization of nodes according to the connectivity of the network. In 2000, Bulusu, the professor of the Southern California University proposed a location algorithm not based on distance—centroid method, which points out, when the quantity of signals of the anchor nodes received by the unknown nodes, is more than a preset threshold in a time, this node will identify the position of itself for the polygon centroid which is constituted of anchor nodes and connected with this node. The method is completely based on network connectivity, relatively simple to achieve, and owns smaller computation, but more number of anchor nodes is needed. In 2001, Niculescu, the professor of the Rutgers University of the United States proposed a new solution—DV-Distance algorithm [5-6]. The method requires measuring the distance between adjacent nodes, then, substitutes the sum of sub-distance with straight-line distance between unknown nodes and anchor nodes. A disadvantage of DV-Distance is that at the time of multi-hop transmission, the measure error of nodes can produce accumulation effect. When the network is large, number of anchor nodes is relatively small and hardware error of node measure distance is relatively large, the cumulative errors become more apparent. A more robust approach is to use the network topology information by calculating the number of hops, rather than the cumulative distance to finish, Niculescu and Nath named it as DV-Hop [5-6].

In this paper, we propose an improved DV-Hop scheme localization, the main idea of our approach is to select a certain number of anchor nodes to achieve the positioning of the unknown nodes without increasing hardware cost of sensor node. Simulation results show the proposed algorithm has better performance than the classic DV-Hop. The main contribution of this

paper to the localization problem in WSN is: proposing a practical, effective and easy localization scheme with relatively higher accuracy and lower cost. The performance evaluation is conducted on a medium scale WSN.

The paper is organized as follows. Section 2 introduced the related work on DV-Hop algorithm. Section 3 described the classic DV-Hop algorithm, and then summed up the main source of error based on the detailed analysis of DV-Hop algorithm, and the simulation experiment's results also proved the existence of error. In section 4, we introduced the improved DV-Hop scheme, and the section 5 introduced the simulation experiments, the experiment results have been given and the localization errors were discussed. Finally, we concluded the paper looked into the future in section 6.

2. Related Work

The research on wireless sensor networks node localization is still one of the focus at home and abroad at present [7-13]; especially, some internal research has been made on the DV-Hop algorithm research.

Zhang Zhaoyang et al [14] proposed two improved DV-Hop algorithms and integrated them reasonably into a self-adaptive positioning algorithm called SAP which includes two modes. Rough-precision mode uses DV-Hop I algorithm and high-precision mode uses DV-Hop II algorithm. In SAP, the first mode saves the energy consumption, the second mode which use RSSI to measure distances between nodes is enabled when target incidents happen, so the algorithm strikes a good balance between positioning accuracy and energy consumption. The simulation results demonstrate that SAP is more efficient in precision and robust than DV-Hop.

Huang Hao et al [15] proposed an enhanced DV-Hop Algorithm. First, it analyzes the location error of a similar "block effect", and then adjusts the location result of unknown node for unknown nodes of around anchor node. Finally, through simulation, "block effect" has effectively been resolved; the algorithm improved positioning accuracy and robustness.

Based on the characteristics of the DV-Hop, an improved scheme was proposed by Chen Kai et al [16] for this typical range-free localization algorithm in wireless sensor network. Its main principle is estimating distance of the hops according to the number of neighbours in the same block. In order to reduce the localization error, it uses weighted node distances to calculate the node's final coordinate.

Yi Xiao et al [17] put forward an improved positioning algorithm based on the DV-Hop algorithm, it shows a differential error correction scheme, in which average per hop distance of the position network and modified value of distance error are introduced, which is proposed to reduce cumulative distance error and node location error accumulated over the multiple hops. Simulation results show that the proposed algorithm can increase location accuracy obviously.

To improve the poor locating performance of DV-Hop algorithm in calculating the distance of unknown nodes to beacon nodes, Bao Xirong et al [18] presented an improved DV-Hop localization algorithm on the basis of analyzing the error of beacon nodes' estimated distances and actual distances. It mainly focuses on two respects: Firstly, the average one-hop distance error of beacon nodes was modified. Secondly, considering several beacon nodes' average one-hop, the average one-hop distance used by each unknown node was modified by weighting the received average one-hop distances from beacon nodes.

To enhance positioning accuracy of wireless sensor network node, a differential error correction scheme was put forward based on the DV-Hop algorithm by Yi Xiao et al [19], in which average per hop distance of the position network and modified value of distance error are introduced, it can reduce cumulative distance error and node location error accumulated over the multiple hops. Simulation results show that the proposed algorithm can increase location accuracy obviously and is applicable to asymmetrical network scene.

Wu Yanhai et al [20] had done the theoretical analysis of the DV-Hop algorithm and gave an improved algorithm. To use correction value as estimated distance between anchor nodes and unknown nodes. Meanwhile, TLS is applied to node localization algorithm to make further localization accuracy. It is proved that the improved algorithm gives better localization accuracy results than original algorithm and other existing improved algorithms.

To improve the accuracy of localization, Li Jian et al [21] proposed an improved DV-Hop algorithm. It is derived from DV-Hop algorithm, and uses weight of anchors to improve localization accuracy without needing no additional hardware device. Simulation results show that the improved DV-Hop algorithm can provide more accurate location estimation than the DV-Hop algorithm.

Zhang Jia et al [22] had used adjusted value of multi-hops and corresponding correction value to reduce the distance error based on the analysis of error sources. Meanwhile, the unknown node coordinates are solved by introducing total least squares method, which effectively suppress the error accumulated to improve the positioning accuracy. However, the computation is significantly increased due to the introduction of multi-hop distance and the corresponding correction value.

3. DV-Hop Algorithm Analysis

3.1. Implementation Process

The core idea of DV-Hop algorithm is: the distance between the unknown nodes and the anchor nodes is indicated by the product of multiplication of the first received with the average distance per hop and the number of their hops. Then the location information of unknown nodes is achieved by using the estimated distance to all anchor nodes. DV-Hop algorithm consists of three phases:

1) Information broadcasting

Each anchor node will convey the location information to all neighbouring nodes. Broadcasting information format is $\{id_i, x_i, y_i, Hops_i\}$, which contains the identity id_i of the anchor node, location coordinates (x_i, y_i) and the number of hops of $Hops_i$ information, and $Hops_i$ is initialized to 0. Each node receives this data and records $Hops_i + 1$ to a table, and then continues to broadcast the new neighbours' node.

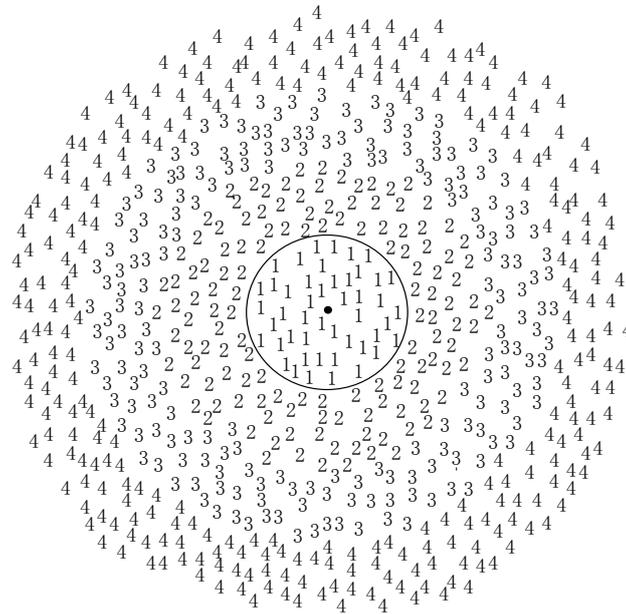


Fig. 1. Broadcasting process of DV-Hop algorithm information

If a node receives a packet with the same id, then it compares Hopsi with Hopsi of packet with the same id on the table. If the new hop count is less than the hop already existing in the table, the new hop will update the information of hop in the table; other wise the packet will be discarded and no longer be forwarded.

The information of each anchor node is broadcasted in the form of flooding on the entire network, thus the hops from each node to each anchor node is obtained. Meanwhile, the anchor node also accesses to the coordinates and hops of all other anchor nodes. So the distance of average per hop for anchor node is calculated by type (1):

$$c_i = \sum \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} / \sum h_{ij} . \tag{1}$$

Where, j is the other anchor node; h_{ij} is the number of hop between the anchor node i and the anchor node j .

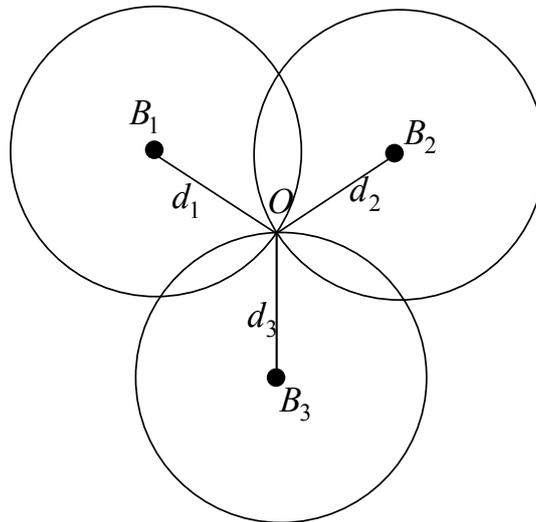


Fig. 2. Geometric representation of unknown node what is located by three anchor node

2) Distance calculation

As for every average distance of per hop calculated by each anchor node broadcast, its data packet format is $\{id_i, c_i\}$. The unknown node receives the data packets of each anchor node, and saves the average distance of per hop of each anchor node with a table, and then continues to broadcast to its neighbours. The data packet will be discarded in the event of meeting duplicate data packets. After the broadcast of information, the mean value of each anchor node's the average distance of per hop is calculated, then get the average distance per hop of entire network, here,

we use 'cc' to indicate it. Then each unknown node can calculate the distance of the node itself to each anchor node and save it into the table.

3) Localization calculation

According to the distance information of many anchor nodes that has been obtained, the unknown node calculates its coordinates using the maximum likelihood estimate of trilateration or multilateral measurement.

When the distance d between all anchor nodes and the location node O is known, according to (2) calculate:

$$\begin{cases} (x_1 - x) + (y_1 - y)^2 = d_1^2 \\ \dots \\ (x_n - x) + (y_n - y)^2 = d_n^2 \end{cases} \quad (2)$$

While (2) can be expressed as:

$$\begin{cases} x_1^2 - x_n^2 + 2(x_1 - x_n)x + y_1^2 - y_n^2 \\ -2(y_1 - y_n)y = d_1^2 - d_n^2 \\ \dots \\ x_{n-1}^2 - x_n^2 + 2(x_{n-1} - x_n)x + y_{n-1}^2 - y_n^2 \\ -2(y_{n-1} - y_n)y = d_{n-1}^2 - d_n^2 \end{cases} \quad (3)$$

The linear equation of (3) representation for:

$$AX = B \quad (4)$$

Where,

$$A = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ \dots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix} \quad (5)$$

$$B = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_n^2 - d_1^2 \\ \dots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_n^2 - d_{n-1}^2 \end{bmatrix} \quad (6)$$

According to (4), the coordinates of location node O can be obtained using estimation methods of the standard minimum mean square.

$$X = (A^T A)^{-1} A^T B \quad (7)$$

3.2. Error Source Analysis and Existing Optimization Strategies

From the execution process of DV-Hop algorithm, it is known that the error mainly comes from two aspects:

1) Algorithm makes the product of the number of hop and the average distance with per hop as the estimated distance between unknown nodes and the anchor nodes. This algorithm has a better result only in the situation when the true distance between nodes in the network is approximately close. The actual situation is different. When the location of unknown node is equated by applying the estimated distance of these anchor nodes, the position precise is low because of the accumulated error. Figure 3 show the generating process of error.

In the Fig.3, L_1 、 L_2 、 L_3 are all anchor nodes. A is the unknown node which need localization. The three anchor nodes know the distance to any other one, shown as Fig. 3.: 30, 30 and 40. The distance between A and L_1 is 15, the number of hop is 1. The hops between A and L_2 or L_3 is 3. Suppose the length of the other each edge is 10.

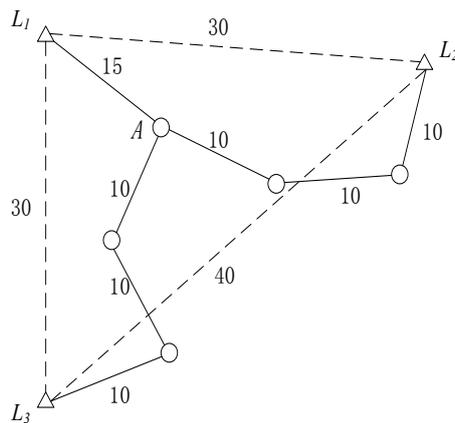


Fig. 3. Analysis diagram of DV-Hop localization error

Known by the DV-Hop algorithm, L_1 , L_2 and L_3 is calculated as follows:

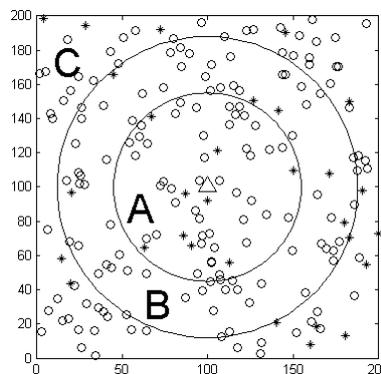
$$L_1: (30+30) / (4+4)=7.5;$$

$$L_2: (30+40) / (4+6)=7;$$

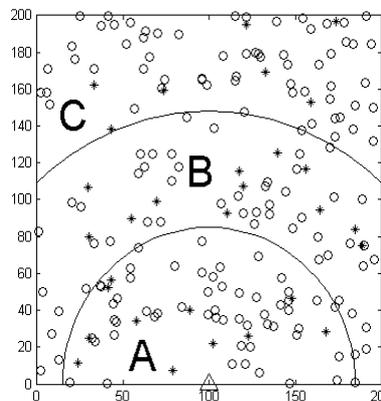
$$L_3: (30+40) / (4+6)=7;$$

After the average distance of per hop computed by the anchor node, the anchor nodes will broadcast this value in the network. The unknown node takes the first value of receiving as the average distance per hop. In the

example, L_1 , L_2 and L_3 will respectively broadcast the calculated value which is 7.5, 7 and 7. Because there is only one hop between node A and L_1 , 7.5 is the average distance of per hop of node A , then node A will calculate the distance between itself and three anchor nodes. The distance between A and L_1 is 7.5, and the distance between A and L_2 or L_3 is 22.5. In fact, the distance between A and L_1 is 15, but the calculated distance is 7.5, the error reaches 50%. It is clearly unacceptable. The error is generated from this point.



(a) The node locate centre



(b) The node locate edge

Fig. 4. Distribution of reference anchor node

2) The number of hop between the unknown node and this anchor node is larger when the anchor node is farther away from the unknown node. Error exists in the average distance of per hop itself, so if the hop count is larger, the error of estimate distance is much larger. If the coordinates of the

unknown node are solved by using the estimate distance of these anchor nodes, then the localization precise of node will reduce.

In Fig. 4, the area of $200m \times 200m$ randomly distribute 200 nodes (* stands for anchor node, o stands for the unknown node). Where, the proportion of the anchor node is 15%, the node communication radius is 50m.

To illustrate the simulation process, (a) and (b) are given the choice situation respectively for the reference anchor node of the unknown node in (100,100), (100, 0). Simulation process is divided into three parts:

1. Select the anchor node near to the unknown node as the reference anchor node, relative to the selected two nodes, shown as following, the area named A in Fig.4. At this time, most are one hop between the unknown nodes and anchor nodes, and only a few nodes own two hops.
2. Select the anchor node far from the unknown node as the reference anchor node, shown as following the area C in Fig.4. At this time, the hop count between the unknown nodes and anchor nodes are two hops at least.
3. Select all anchor nodes in the network as the reference anchor. The process is same to DV-Hop algorithm.

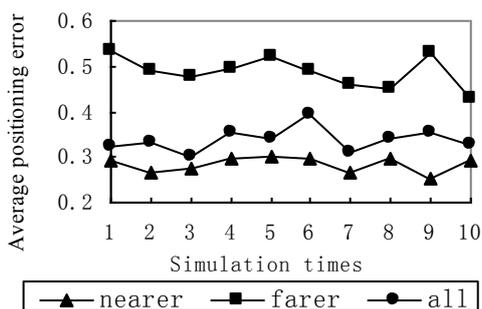


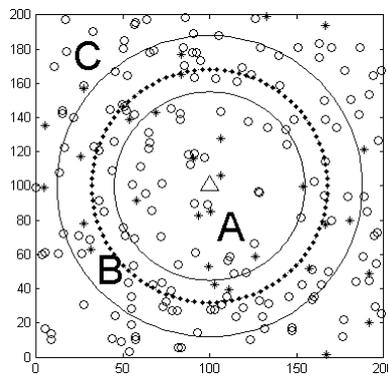
Fig. 5. Average location error of the different position anchor nodes in environment

Analysis of experimental data showed that, when select the anchor node far from the unknown node as the reference anchor node, the average location error obtained is about 48.70%. In the same condition, when select the anchor node near to the unknown node as the reference anchor node, the average location error obtained is about 28.94%, finally, when selecting all anchor nodes in the network as the reference anchor, the average location error obtained is about 33.88%. It is said that when select all anchor nodes in the network as the reference anchor, the average positioning error is far greater than the other two cases. The main source of error caused is the far away anchor nodes. Finally, we conclude that:

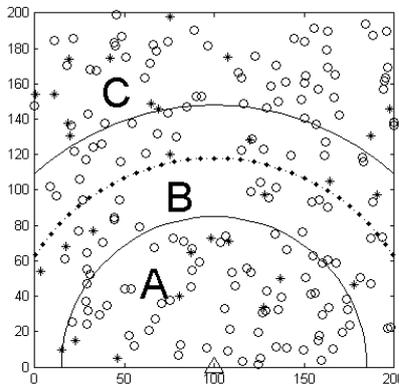
1. The average error of selecting all the anchor nodes in the network anchor node as a reference node is not the lowest.

2. The affect of localization error is larger when the anchor node is far away the unknown.

At the same time, in the simulation process we also found that, with the difference of the size of region A, the average location error is relatively larger when select from the unknown node closer to the different number of anchor nodes. Aimed at the two nodes selected, we change the size of A area when only considering the node itself position error, finally we got the anchor node selection condition up to the lowest location error through several simulation experiments. The area is indicated by dot line in Fig.6.



(a) The node locate centre



(b) The node locate edge

Fig. 6. Minimum location error distribution of reference anchor node

4. A New Location Algorithm NDV-Hop_Bon

4.1. The Location Idea

Under the premise of DV-Hop idea, the number of hops among nodes is generally fixed. In order to obtain the minimum error between the estimated distance and actual distance, it can be obtained by adjusting to the average distance of per hop. It was found that the average location error is not the smallest when selecting all the anchor nodes as reference anchor node of unknown nodes, but it has a better result when select the node near the unknown node. At the same time, the achieved product of average distance of per hop and the hops count is the true of closet. Experimental results show the number of optimal anchor nodes that can be up to the lowest average location error is involved with the network environment parameters, and different parameters of the network environment. Besides, the number of optimal anchor nodes is different.

Thus, a new localization idea is proposed. First of all, each unknown node initially sets a threshold named N . After exchanging information, we can get the estimate distance of the anchor node. Then select the information of N nodes which is received firstly, equate the coordinates of unknown nodes using least square method. Due to only selecting the anchor nodes with the short distance, the new localization algorithm not only reduces the computational in the process and improve efficiency, but also achieves a minimum average positioning by selecting the number of optimal reference anchor nodes.

4.2. Setting the Threshold N

Lots of simulation experiments showed that the number of optimal anchor nodes is involved with the following parameters of the network environment:

1. The area of the sensing region, we take A as the side of the sensing region, and assuming that a sensor area is always square.
2. Communication radius of sensor nodes, it is said that the nodes within the region of R can communicate with each other.
3. The proportion P of anchor node in the network, which means the number of anchor nodes in the network takes the proportion of the total number of nodes, the number of all nodes is denoted by TN .
4. The average θ of connectivity between the unknown nodes and the anchor nodes in the network. Define:

$$\theta = \frac{\sum_i Con_Anc_i}{U_n} \quad (8)$$

Where, U_n is the number of the unknown nodes; Con_Anc_i is the connectivity between the node i and anchor nodes in the network, that is to say, the number of anchor nodes that unknown nodes can communicate with.

The threshold N need to change in accordance with the changes of the network environment parameters, the specific change process is as follows:

1. The coverage of each node in the network is relatively smaller when the sensing area is greater and the communication radius of sensor nodes is smaller, that is to say, $A^2/\pi R^2$ is greater. Then the number of anchor nodes for locating the unknown nodes is larger; it is said that the value of N is directly proportional with $A^2/\pi R^2$.
2. When the connectivity between unknown nodes and anchor nodes is larger, that is θ has a larger value, the number of anchor nodes within one hop far away from unknown nodes is larger. At this time, the anchor nodes calculate the average distance of per hop. Because the number of per hop is smaller, it makes the average distance of per hop that is equated too large. So the number of optimal nodes should be increased and make it balance the length of the average distance of all per hop.
3. When P is very small (less than 5%), because the number of anchor nodes is very few in the network, at this time, better results can be gotten through selecting all anchor nodes. It's said that the value of N is fixed $1/P$. When P is larger than 5%, the number of all anchor nodes is increased with the accretion of the proportion P of anchor nodes in the network. As the relationship is not linear proportional relationship, we can set an option βP to adjust, which is a proportional factor. The value of βP is changed by adjusting the value of β . Thus it makes the value of βP tend to optimal.

The above analysis, the formula of the threshold N is defined as:

$$N = \begin{cases} \frac{A}{R} \frac{1}{\sqrt{\pi}} \times \theta \times (1 + 0.1P) & P > 25\% \\ \frac{A}{R} \frac{1}{\sqrt{\pi}} \times \theta \times (1 + \beta P) & 5\% \leq P \leq 25\% \\ \frac{1}{P} & P < 5\% \end{cases} \quad (9)$$

4.3. Determination of β Factor

The above formula β is the factor of adjusting the proportion P of the anchor nodes to the threshold. When the sensing area size and node communication radius is constant, with changes of the proportion for the anchor node in the network, the value will change accordingly to achieve the control of unknown node selecting the optimal number of anchor nodes. A large number of experiment data are obtained, and the final statistical analysis shows:

1. With the increase of the ratio P of anchor nodes, on the one hand, the number of optimal anchor nodes is growing with the increase of proportion of the anchor nodes. On the other hand, the proportion that the best anchor node relative to the total number of anchor nodes is constantly reducing. When the ratio of anchor nodes in the network increases, the number of the total anchor nodes increases and the anchor nodes of around unknown nodes are increased correspondingly. Then the increase of the number N of optimal anchor nodes ensures that there is enough anchor nodes to locate, which helps to improve the positioning accuracy. But due to the more overall anchor node, it can reach optimum positioning accuracy by selecting a relatively small proportion of the total number of anchor nodes anchor node. Reflected in the formula, the value of β is reduced as the constant decrease of the ratio P .
2. When the ratio of anchor nodes in a certain range of 1) (for example from 5% to 10%, from 10% to 15%). With the increase of the ratio of anchor nodes, the number of anchor nodes of around unknown nodes in the network can increase. At the same time, it can receive more high positioning accuracy when number of the optimal anchor nodes increases relatively. It is said that the value of β is relatively fixed which makes N grow with the increase of P in the process of the increase of P in this region.
3. With further increase of the ratio of anchor nodes (Greater than 25%). The number N of optimal anchor node will remain in a stable range. At this time, unknown nodes around the anchor node are enough, while the number of optimal anchor nodes selected can achieve a lower average position error. In theory, the value of β should continue to become smaller to meet the fixed value of N . But because the value of β is already smaller, the change of itself has little effect on the number of the optimal anchor nodes, that is, the value of β has stabilized. It means when P is greater than 25%, the fixed value of β is 0.1.

Finally, the relationship between β factor and the ratio of anchor nodes is given, as shown in Table 1.

Table 1. The relationship between β factor and the ratio of anchor nodes

P	5%~10%	10%~15%	15%~20%	20%~25%	>25%
β	20	5	2.5	0.5	0.1

5. Simulation and Analysis

To verify the performance of the new algorithm, we use Matlab to simulate, here, assuming the sensor area is square region. The definition of the average location error is shown as follows:

$$Error = \frac{\sum_i D_i / R}{Un} \quad (10)$$

Where, D_i is the difference of the estimated distance and the real distance; R is communication radius, UN is the number of the Unknown node.

Here, the prerequisite are the following supposition: (1) The node deployed region of wireless sensor network is the two-dimensional surface region, and the node has the function of adjusting the correspondence radius scope; (2) The node uses the model of free space radio wave propagation, that is to say, the node's communication range is take the node as a central concentric circle; (3) The node has the ability of symmetrical correspondence, and suppose the messages of all send can be received correctly; (4) The neighbour node can carry on the correspondence directly, namely the two nodes in the scope of correspondence radius can carry on the correspondence directly; (5) The part of nodes have realized own localization through installing the GPS instalment or deploying artificially beforehand, this kind of node is called as the anchor node; (6) There is only two kind of nodes in the network: the anchor node and the unknown node; (7) Besides the own position data known, the other attributes of the anchor node are the same as the unknown node.

Fig.7 and Fig.8 show the influence of the different proportion of anchor nodes to the average positioning error. Simulation environment are separately (a): 200 nodes are distributed randomly in the region of $200m \times 200m$; the communication radius is 50m; (b) 100 nodes are randomly distributed in the region of $100m \times 100m$; the communication radius is 50m.

The data in chart shows that the average positioning error of NDV-Hop_Bon algorithm is lower than that of DV-Hop algorithm, and will reduce with the increase of the proportion of anchor nodes. The positioning accuracy of DV-Hop algorithm tends to stabilize when the proportion of anchor nodes is larger than 15% in the environment of (b). The new positioning accuracy of the positioning algorithm can still be improved; the positioning accuracy is 22.3% when the anchor percentage is more than 25%.

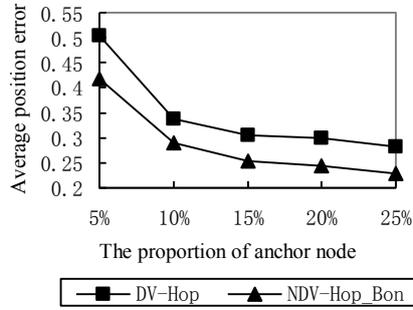


Fig. 7. The average location error of the proportion of the different anchor nodes in environment (a)

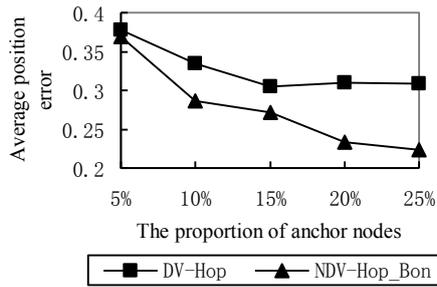


Fig. 8. The average location error of the proportion of the different anchor nodes in environment (b)

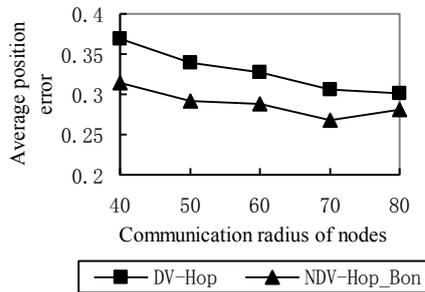


Fig. 9. The average location error of different communication radius in the environment (a)

Fig.9 and Fig.10 show the influence of the different communication radius to the average positioning error. Simulation environment are separately (a): 200 nodes are distributed randomly in the region of $200m \times 200m$; the ratio of anchor nodes is 10%; (b) 100 nodes are randomly distributed in the region of $100m \times 100m$; the ratio of anchor nodes is 10%.

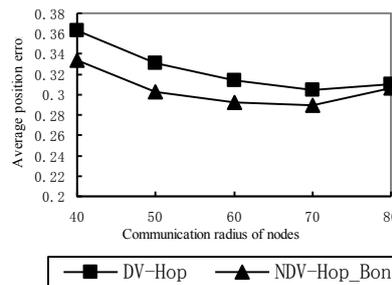


Fig. 10. The average location error of different communication radius in the environment (b)

We can see from the Fig.9 and Fig.10, NDV-Hop_Bon algorithm is superior DV-Hop Algorithm in both environments. The average position error of the new position algorithm is growing with the increase of communication radius when the communication radius is up to 80m. The reason is that the communication radius has reached 40% of sensing region side, and it is said that most of nodes can directly communicate with any other one. Generally, there is one hop between the anchor nodes and the unknown nodes, so the estimate distance of computation will be smaller, thus affecting the positioning accuracy. When the communication radius of the anchor node is 70m in the environment (b), at this time the proportion between the communication radius and the sensor region side is 70%. Almost all of the nodes in the network are one hop, and the error of estimated distance also is high, so there is a decline in positioning accuracy. This can explain that the new location algorithm has a lower requirement for the communication radius of nodes. It is said that the algorithm can achieve a high positioning accuracy when the nodes communication radius are set very small. From the energy point of view, the energy consumption is the lowest in the situation of the same position accuracy.

Fig.11 shows the influence of the average position error for the sensor region of difference size. Simulation experiment's environment is: the proportion of the anchor nodes is 10%, and the communication radius of nodes is 50m. The number of nodes becomes larger with the increase of sensing region and the number is increased from 100 to 500. It can be seen from the figure that the positioning error of NDV-Hop_Bon algorithm is lower than the effect of the DV-Hop Algorithm. However, the position error can increase with the growing of the area of sensor region. Through analysis, we can infer that it is the affect of communication radius of node. The number of

hops between most unknown nodes and position anchor nodes using is relatively large if the communication radius is too small. Due to the increasing number of hops resulted in cumulative error, thereby affecting the positioning accuracy.

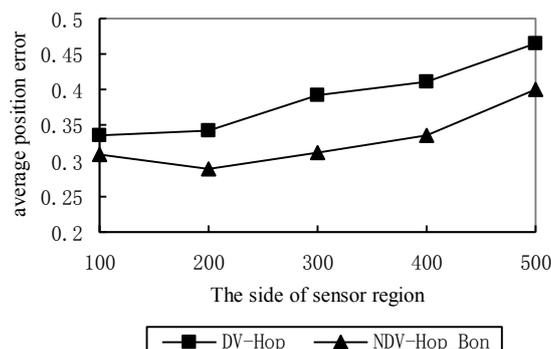


Fig. 11. The influence of the average position error for the sensor region of difference size

6. Conclusions

In order to improve the position performance of the node localization algorithm in the sensor network of distributing randomly, the paper summed up and analyzed the error of the position process of DV-Hop algorithm, then proposed a new position algorithm NDV-Hop_Bon in view of the disadvantage of all anchor node position. The new localization algorithm selects the reference nodes of a certain number to position the unknown nodes based on different environments. Finally, the results show that the positioning accuracy of the new positioning algorithm is better than DV-Hop. And the algorithm has a more prominent performance with the growing of the proportion of anchor nodes, the increase of sensor region area and the increase of the number of sensor nodes. Meanwhile, due to a lower requirement on the node communication radius, on one hand, it saves the energy; on the other hand, it can also be more suitable for the positioning of the large sensor networks.

As the newly proposed algorithm still has the limitation on the application, when the anchor node's energy is too low, the positioning accuracy can not meet the requirements. At the same time, communication expense is increased due to introduce of the computation for the number of optimal anchor nodes. Therefore, the key issues for further research and direction is to improve the scalability of the algorithm, optimization of anchor nodes when the proportion is too low, and ensuring the case of the lower positioning error with the reduction algorithm communication.

References

1. Chatterjee, A.: A Fletcher-reeves Conjugate Gradient Neural-Network-Based Localization Algorithm for Wireless Sensor Networks. *IEEE Transactions on Vehicular Technology*, Vol. 59, No.2, 823-830.(2010)
2. Li, M., Liu, Y.: Rendered Path: Range-Free Localization in Anisotropic Sensor Networks With Holes. *IEEE/ACM Transactions on Networking*, Vol. 18, No.1, 320-332. (2010)
3. Kwon, O., Song, H., Park, S.: The Effects of Stitching Orders in Patch-and-Stitch WSN Localization Algorithms. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 20, No.9, 1380-1391. (2009)
4. Katenka, N., Levina, E., Michailidis, G.: Robust Target Localization from Binary Decisions in Wireless Sensor Networks. *Technometrics*, 448-461. (2008)
5. Niculescu, D., Nath, B.: Ad-Hoc Positioning Systems (APS). In *Proceedings of the 2001 IEEE Global Telecommunications Conference [S1]: IEEE Communication Society*, 2926-2931. (2001)
6. Niculescu, D., Nath, B.: DV Based Positioning in Ad Hoc Networks. *Journal of Telecommunication Systems*, Vol. 22, No.4, 267-280. (2003)
7. Umer, M., Kulik, L., Tanin, E.: Spatial Interpolation in Wireless Sensor Networks: Localized algorithms for Variogram Modeling and Kriging. *Geoinformatica*, 101-134. (2010)
8. Xiong, K., Thunte, D.: Dynamic Localization Schemes in Malicious Sensor Networks. *Journal of Networks*, Vol. 4, No.8, 677-686. (2009)
9. Baouche, C., Freitas, A., Misson, M.: Radio Proximity Detection in a WSN to Localize Mobile Entities within a Confined Area. *Journal of Communications*, Vol. 4, No.4, 232-240.(2009)
10. Caballero, F., Merino, L., Gil, P., Maza, I., Ollero, A.: A Probabilistic Framework for Entire WSN Localization Using a Mobile Robot. *Robotics and Autonomous Systems*, Vol. 56, No.10, 798-806.(2008)
11. Pei, Z. M., Deng, Z. D., Xu, S., Xu, X.: A New Localization Method for Wireless Sensor Network Nodes Based on N-Best Rand Sequence. *Acta Automatica Sinica*, Vol. 36, No.2, 199-207. (2010)
12. Wang, J. C., Huang, L. S., Xu, H. L., Xu, B., Li, S. L.: A Novel Range Free Localization Scheme Based on Voronoi Diagrams in Wireless Sensor Networks. *Computer Research and Development*, Vol. 45, No.1, 119-125. (2008)
13. Wang, S. S., Yin, J. P., Cai, J. P., Zhang, G. M.: Localization Algorithm for Wireless Sensor Networks Based on RSSI. *Computer Research and Development*, Vol. 45, No.suppl, 385-388. (2008)
14. Zhang, Z. Y., Gou, X., Li, Y. P., Huang, S. S.: DV-Hop Based Self-Adaptive Positioning in Wireless Sensor Networks. *The 5th International Conference on Wireless Communications Networking and Mobile Computing*. 1-4.(2009)
15. Huang, H., Lu, W. K., Xu, C. H., Ye, M. Q.: An Enhanced DV-Hop Algorithm in WSN. *Periodical of Ocean University of China*, Vol. 38, 217-220. (2008)
16. Chen, K., Wang, Z. H., Lin, M., Yu, M.: An Improved DV-Hop Localization Algorithm for Wireless Sensor Networks. *IET International Conference on Wireless Sensor Network*, 255-259.(2010)
17. Yi, X., Liu, Y., Deng, L., He, Y.: An Improved DV-Hop Positioning Algorithm with Modified Distance Error for Wireless Sensor Network. *The 2th International Symposium on Knowledge Acquisition and Modeling*, Vol. 2, 216-218.(2009)
18. Bao, X. R., Bao, F. P., Shi, Z., Liu, L.: An Improved DV-Hop Localization Algorithm for Wireless Sensor Networks. *The 6th International Conference on Wireless Communications Networking and Mobile Computing*, 1-4.(2010)

Qingji Qian, Xuanjing Shen, and Haipeng Chen

19. Yi, X., Liu, Y., Deng, L., He, Y.: An Improved DV-Hop Positioning Algorithm with Modified Distance Error for Wireless Sensor Network. The 2th International Symposium on Knowledge Acquisition and Modeling. Vol. 2, 216-218.(2009)
20. Wang, Y. H., Zhang, J., Nan, W., Fang, G.: An Improved DV-Hop Localization Algorithm for Wireless Sensor Networks. International Conference on Intelligent Computation Technology and Automation. Vol. 2, 576-579.(2011)
21. Li, J., Zhang, J. M., Liu, X. D.: A Weighted DV-Hop Localization Scheme for Wireless Sensor Networks. The 8th International Conference on Embedded Computing. 269-272.(2009)
22. Zhang, J., Wu, Y. H., Shi, F., Geng, F.: Localization Algorithm Based on DV-Hop for Wireless Sensor Networks. Journal of Computer Applications, Vol. 30, No.2, 23-326. (2010)

Qingji Qian, female, born in 1962, associate professor in the College of Physics, Jilin University. Her main research interests include wireless sensor networks, embedded technology, and applied optics.

Xuanjing Shen, male, born in 1958. professor and PhD supervisor currently in the college of computer science and technology, Jilin University. His research interests are wireless sensor networks, multimedia technology, computer image processing, intelligent measurement system, optical-electronic hybrid system, and etc.

Chen Haipeng, born in 1978. Ph.D., a lecturer in the college of computer science and technology, Jilin University. Member of China Computer Federation (E200015167M). His main research interests include wireless sensor networks, digital image processing and pattern recognition, and computer network security.

Received: Februar 22, 2011; Accepted: April 22, 2011.

Energy-Efficient Opportunistic Localization with Indoor Wireless Sensor Networks

Feng Xia^{1*}, Xue Yang¹, Haifeng Liu¹, Da Zhang¹ and Wenhong Zhao²

¹ School of Software, Dalian University of Technology,
Dalian 116620, China
f.xia@ieee.org

² College of Mechanical Engineering, Zhejiang University of Technology
Hangzhou 310032, China

Abstract. Localization challenges researchers for its contradictive goals, i.e., how to tackle the problem of minimizing energy consumption as well as maintaining localization precision, which are two essential trade-offs in wireless sensor network systems. In this paper, we propose an Energy-Efficient Opportunistic Localization (EEOL) scheme to satisfy the requirement of positional accuracy and power consumption. We explore the idea of opportunistic wakeup probability to wake up an appropriate numbers of sensor nodes, while ensuring the high positional accuracy. Sensor nodes can be triggered by the opportunistic wakeup probability sent from the user. Through utilizing this method, the number of active sensors in the sensing range of the user is decreased, and the power consumption is significantly reduced. Theoretical analysis has been presented to evaluate the performance of EEOL. Simulation results show that EEOL confirms our theoretical analysis.

Keywords: Indoor Localization, Energy Efficiency, Wireless Sensor Network, Opportunistic Localization.

1. Introduction

As a new technology of information collection and dissemination, Wireless Sensor Network (WSN) has greatly extended our ability in target tracking, monitoring, intrusion detecting and other localization applications [1]. It has aroused great concern and been widely used in national defense, national security, environmental monitoring, traffic management, health care, manufacturing, antiterrorism and other disaster areas.

Localization is an inevitable challenge when we deal with sensor nodes. It is a problem that has been studied for many years. Without the nodes' position information, the monitoring information is always meaningless in the

* Corresponding author: Feng Xia (f.xia@ieee.org)

applications of WSNs. Consequently, a variety of WSN systems [2-4] for indoor localization have been developed and well tested.

In all these applications, energy consumption is one of the most important issues [5]. For sensor nodes in small size, energy is restricted and it is not convenient to charge or change batteries. As a result, it requires the power consumption to be as little as possible to extend the network's lifetime. And, how to take full advantage of the energy to maximize the network's efficiency has become one of the primary challenges for sensor networks. Nevertheless, power management in WSNs remains to be a difficult problem and it is unlikely to be solved in the near future, since the progress in battery capacity is slow. Several research issues remain to be solved before applications such as military tracking, environmental monitoring and positioning become economically practical.

Experimental results have shown that the energy consumption of wireless devices in the idle state is slightly less than that in transmitting and receiving states. As a result, an important approach to reduce power consumption is to switch the nodes into the low-power sleep mode as long as possible. There is an effective approach proposed for saving the power of sensor nodes using the duty cycling policy in [6]. It is true that many methods have been proposed in the literature, but most existing protocols need to wake up sensors periodically. Hence, as for localization, certain amount of energy is still wasted by using unnecessary sensors. Basically, we can use only two or three sensors to locate an object based on the technique of localization, and thus the power of other active sensor nodes could be saved. Therefore, it would be a proper approach to balance the requirements of positional accuracy and the power consumption restrictions by waking up an appropriate number of sensor nodes.

Opportunistic computing model motivates the basic idea of the relationship between the number of sensor nodes and the positional accuracy, which is the key to balancing power consumption and positional accuracy. Based on the opportunistic computing model, we propose an Energy-Efficient Opportunistic Localization (EEOL) scheme that regulates the sensors' on-off states through changing the wakeup probability of each sensor dynamically, according to the number of the sensors. Furthermore, it ensures that the number of active sensor nodes is large enough for the localization techniques, Trilateration algorithm for example.

Our proposed approach aims at obtaining the appropriate number of active sensors to locate an object in the process of localization, and avoiding unnecessary waste of energy. As for the total network efficiency, the power consumption significantly decreases and the network performance is improved. In a word, we achieve the purpose of reducing power consumption while ensuring the positional accuracy.

The rest of this paper is organized as follows. In Section 2, we discuss related work from the literature, presenting the context for our work. Section 3 provides our assumptions, scenarios and notation descriptions. Then, we describe our EEOL algorithm in Section 4. Simulation results are given in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

The basic concept of WSN and its localization algorithms have been the subject of many research studies recently. Here, we focus mainly on the WSN indoor localization systems and energy-efficient localization algorithms.

2.1. WSN indoor localization systems

Numerous works have already analyzed the performance of WSN indoor localization systems over the past decades, [7] for example, during which, new systems are being developed continuously.

Lots of the existing indoor localization systems are based on range-based schemes that exploit e.g. the Received Signal Strength (RSS) technology. The localization technique mentioned in [3] can achieve high accuracy and stability in indoor environment. MMSE proposed in [4] can minimize the localization errors. Additionally, the Dual-modal Indoor Mobile Localization System [8], which was implemented using the RSS approach and the unscented Kalman filter (SPKF) algorithm in active and passive dual-modal architecture, decreases the system cost and simplifies the sensor deployment. Further, WAX-ROOM in [9] incorporates three different localization techniques, namely a plain-RSS technique, a SA-RSS technique, and the range-free APIT algorithm, as well as an Optimal Fusion Rule (OFR) in order to leverage localization accuracy.

There are many other techniques for indoor localization. Localization of Sensor Nodes by Ultra-Sound (LOSNUMS) [10] offers a high accuracy of 10 mm, a locating rate up to 10 cycles/s, and it is applicable for both tracking mobile and locating static devices. The Crickets motes use the Time Difference of Arrival (TDoA) between the RF and the ultrasound signals to estimate the distance of the object. In [2] a system consisting of Cricket wireless sensor motes, a camera and a Pan/Tilt gimbal was proposed to solve the indoor localization and surveillance problems.

2.2. Energy-efficient localization

Energy consumption is one of the most important issues we concern in recent years. A number of methods have been proposed to address the energy efficiency problem.

A localization algorithm based on particle filtering for sensor networks was proposed in [11]. It is assisted by multiple transmit-power information, which outperforms the existing algorithms that do not utilize multiple power information.

Given a specified positional error tolerance in an application, the sensor-enhanced, energy-efficient adaptive localization system [12] dynamically sets sleep time for the sensors, adapting the sampling rate of target's mobility

level. It achieves better energy saving while conforming to application's error tolerance. However, the process of error estimation dynamically depends on several factors in the environment.

LPL (Low-Power Scheme for Localization) [13] and OLP (Optimized Listening Period) [14] explore decreasing the idle listening and an optimized allocation of the localization tasks on the nodes. In LPL, the mobile nodes (MNs) transmit packets and the anchor nodes (ANs) take RSSI measurements. During the transmissions, the MN goes into sleep mode to save energy. While in the OLP, the ANs are synchronized and transmit packets to the MNs in a short time. OLP was designed to keep the inter-arrival time of the transmitted packets as short as possible and reduce the idle listening time in the MN. Nonetheless, for large scale sensors networks, the energy consumption is still significant.

In [15], the presented scheduling algorithm selects a subset of active reference nodes to be used in localization, which serves to reduce the message overhead, increase network lifetime, and improve localization accuracy in dense mobile networks. The key for the decision of reference nodes is the design parameter which describes the average density required to ensure localization accuracy with high probability. Reference nodes remain active for several seconds. After that, all devices wake up and new references can be selected. However, maximizing the nodes' sleep time that the nodes never wake up until the reception of wakeup messages is much more energy efficient.

In the algorithms mentioned above, the duty cycle of the sensor nodes is fixed in advance. While in [6], an innovative probabilistic wakeup protocol is proposed for energy-efficient event detection in WSNs, the central idea of which is to reduce the duty cycle of every sensor via probabilistic wakeup, exploiting the dense deployment of sensor networks.

Our idea is initially inspired by the selection of the subset of active nodes and the probabilistic wakeup protocol. We have studied existing localization techniques [16]. According to the Trilateration algorithm, three sensors can locate a target, and the more the sensors, the more accurate the localization will be. In our study, we aim at reducing the number of the active sensors by employing opportunistic wakeup probability to save power consumption as well as satisfying the requirement of positional accuracy. Inspired by the idea of opportunistic computing as well, we manage to reduce the message communication between the sensor nodes and the mobile target, which also contributes to energy saving.

3. Problem Description

In this paper, we consider a wireless network composed of n randomly deployed nodes, and each of them is aware of its own position. All sensor nodes in the network are distributed randomly in a two-dimensional indoor environment, which are initialized to sleep state at the beginning. Assume

that one mobile node, called *User*, is moving freely, in need of localization service. However, the *User* can only be located by the active sensor nodes. The sensors will be waked up by the wakeup probability. Nodes are equipped with radio transmitters/receivers for communications. By using RSS or time of arrival (TOA) of radio signals, nodes can estimate the distance to the *User*.

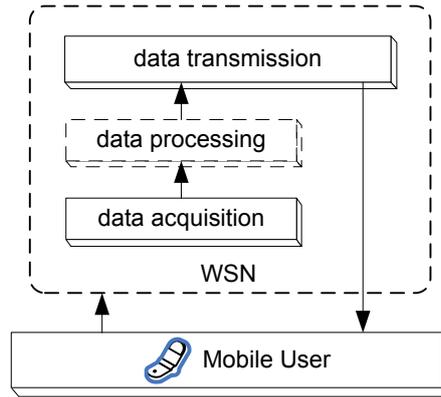


Fig. 1. A localization system

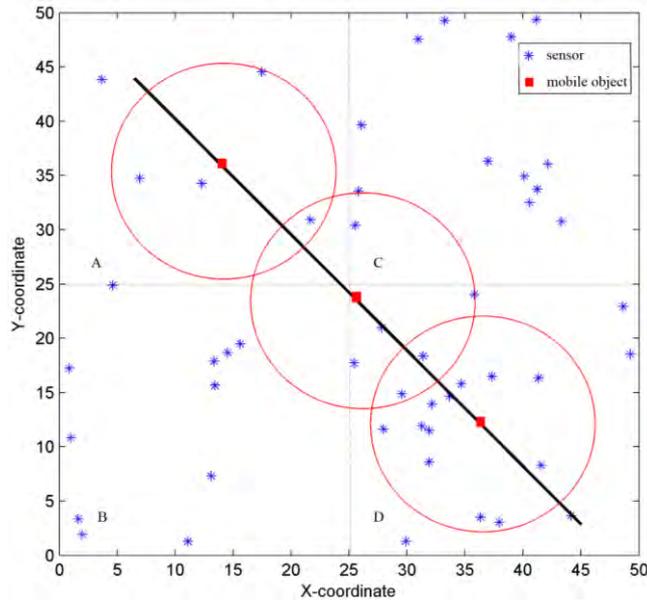


Fig. 2. A scenario of locating a mobile object with 50 distributed sensor nodes

Our localization system is shown in Fig.1. The mobile *User* (who uses a PDA or mobile phone with wireless sensors) comes into the network and calls for the localization service. After the process of data acquisition and data

transmission, the localization algorithm is implemented to compute the required location. Then the User gets its own location information, using the returned data from the network.

In the Data Acquisition Phase, we need to measure the distance between the object and the nodes. Then, here comes the problem. As shown in Fig. 2, when the mobile object is moving in places like Region D, since there are more sensor nodes around than Region A, some nodes are redundant according to the Trilateration algorithm, which means a waste of energy. If we shut down the unnecessary sensor nodes while ensuring the number of sensor nodes $n \geq 3$, it will be a significant contribution for reducing the power consumption of the entire sensor network.

In this paper, we try to develop a method to control the number of sensor nodes in the sensing range of the mobile object to achieve our goal of minimizing power consumption. Our proposed method is based on the following assumptions:

- The number of active sensors around the mobile object n is large enough ($n \geq 3$), then the object can be accurately positioned without considering other environmental factors;
- Once the sensor node wakes up, all parts of the sensor are active, including the sensor data processing unit, sensing devices, data transmission equipment, data receiving device and system clock ;
- In the sleep state, the sensor nodes can exchange opportunistic data, so that it can regulate its on and off states to complete localization;
- The sensor nodes, in the sensing range of the User, are independent, in other words, waking up a sensor will not affect the on or off rate of other sensors.
- The sensing range of the sensor nodes is the same with that of the sensor embedded in the User;
- The wakeup probability sent to the sensor nodes within the sensing range of the mobile object, is the same;
- There is only one User.

4. Energy-Efficient Opportunistic Localization

In this section, we will propose the Energy-Efficient Opportunistic Localization (EEOL) scheme. The work flow of the algorithm is depicted in Fig. 3. Our scheme uses the opportunistic wakeup probability based on the opportunistic computing model to schedule the sensors' on-off states and to obtain an appropriate number of active sensor nodes to locate the User. Naturally, the Data Acquisition Phase is the focus of this paper, and the appropriate number of sensors is the key to the problem.

We will first explain how the opportunistic wakeup probability makes contribution to power saving, as well as the relationship between opportunistic wakeup probability and the number of sensor nodes.

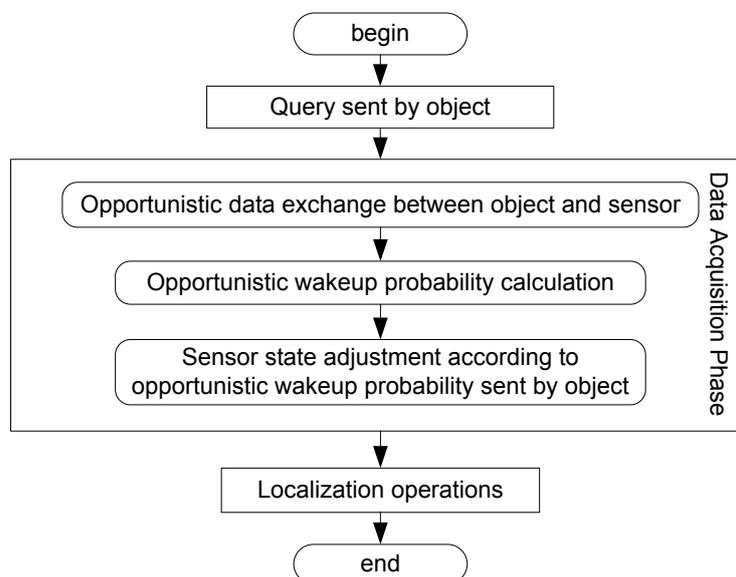


Fig. 3. Work flow of the localization system

Researchers have developed lots of protocols and algorithms in order to extend lifetime of the system and to maintain the positional accuracy. Typically, sensors can be divided into three modes according to their working status: active mode, sensing mode and sleep mode [6]. Previous research shows that for the sake of maximizing power savings, an optimal way is to set duty cycle for the system in which sensors turn on or off periodically.

Setting a fixed value of duty cycle ratio δ to turn each sensor node on or off periodically is fairly easy to implement. However, a defect of this method is that a sensor will still periodically turn on or off even when there is no event occurs, which is not flexible enough and might lead to a waste of energy.

According to the localization techniques, the more the sensors, the more accurate the localization will be. When more sensor nodes are detected in the User's sensing range (hereinafter referred to as in-sensors), the wakeup probability is smaller, provided that the positional accuracy can be ensured. Hence, we exploit the opportunistic wakeup probability instead of static duty cycle, so that the sensor nodes can decide by themselves whether to wake up or to continue to sleep.

4.1. Energy-efficient localization

Every node in the network is equipped with a common wireless communication interface that is used for (opportunistic) data exchange. The radio transmission is always correctly received within a distance R (coverage range) from the transmitter, whereas it might not be correctly received from

longer distances. Therefore, opportunistic data exchange requires the nodes to be mutually in range. Assume that opportunistic data exchange immediately takes place as soon as both conditions are met. Such an event is coined rendezvous [17].

When the User goes into the coverage range of the nodes, messages will be exchanged between the nodes and the object. The object becomes aware of the number of sensor nodes within its sensing range by the messages received from the nodes, which will be used for the calculation of opportunistic wakeup probability.

4.2. Opportunistic wakeup probability calculation

According to the Trilateration localization algorithm, in two-dimensional space, we need at least three reference nodes' location information to locate an object. Because we assume that the nodes are independent, when the number of in-sensors $n \geq 3$, the probability of successfully locating a User obeys the binomial probability $B(n, p)$, where n represents the number of sensor nodes in the sensing range of the mobile object, and p represents the wakeup probability of sensor node. The probability of accurately locating a moving object can be obtained from (1).

$$P(K \geq 3) = 1 - P(K = 0) - P(K = 1) - P(K = 2) \quad (1)$$

$$P(K \geq 3) = 1 - C_n^1 p (1-p)^{n-1} - C_n^2 p^2 (1-p)^{n-2} - (1-p)^n \quad (2)$$

where P represents the probability of accurately locating the mobile object, and K represents the number of sensor nodes to wake up. According to (2), we can map out the quantitative relation between the number of in-sensors and the probability of accurately locating a User, under certain preconditions of each sensor's wakeup probability.

From Fig. 4 we can easily see that, given a certain number of in-sensors, the higher the wakeup probability, the higher the accurate positioning probability. On the other hand, given a certain positional accuracy, the more in-sensors, the lower the wakeup probability and the more the power can be saved. Consequently, setting the number of in-sensors reasonably for the User will achieve energy-efficient localization while ensuring accuracy.

For different applications, the positional accuracy is different. At any time, the probability of positioning a User must satisfy the lowest positional accuracy λ , as Eq. (3) shows.

$$P(K \geq 3) \geq \lambda \quad (3)$$

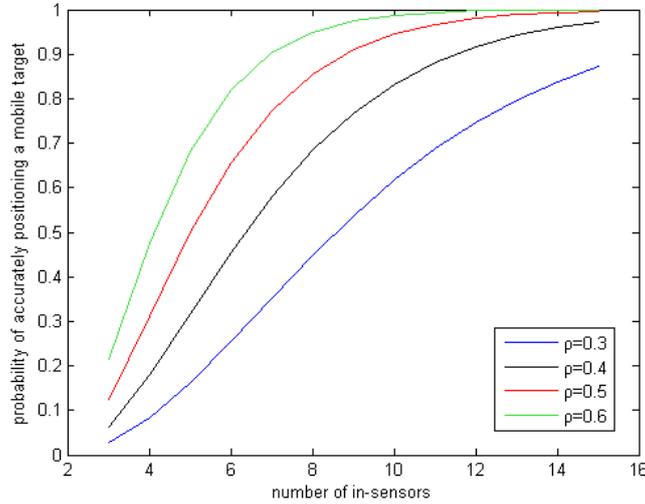


Fig. 4. Relationship between probability of accurate positioning and number of in-sensors

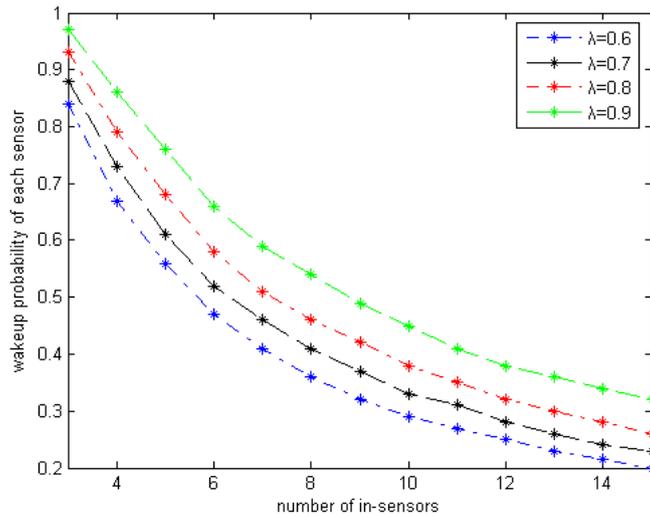


Fig. 5. Relationship between sensor wakeup probability and number of in-sensors

Based on (2) and (3), we get the numerical relations between the number of in-sensors and the wakeup probability which is shown in Fig. 5. In Fig. 5, with the increase of the number of in-sensors, the wakeup probability of each sensor decreases. It can be found that the relationship between the number of in-sensors and the wakeup probability fits the features of the exponential distribution function. In order to make this wakeup probability easier to be applied in indoor localization systems, we conduct some simulations to obtain

the functional relationship between the wakeup probability and the number of in-sensors.

We use MATLAB curve fitting technique to get the relation between the wakeup probability and the number of in-sensors when $\lambda = 0.8$. After fitting, it gives a very satisfactory fitting result with the exponential distributions of 95% confidence intervals. The residual is the difference between the real value and the estimated value. In Fig. 6, we can see that the basic residual is in (-0.005, 0.005), which has already achieved a good degree of fitting.

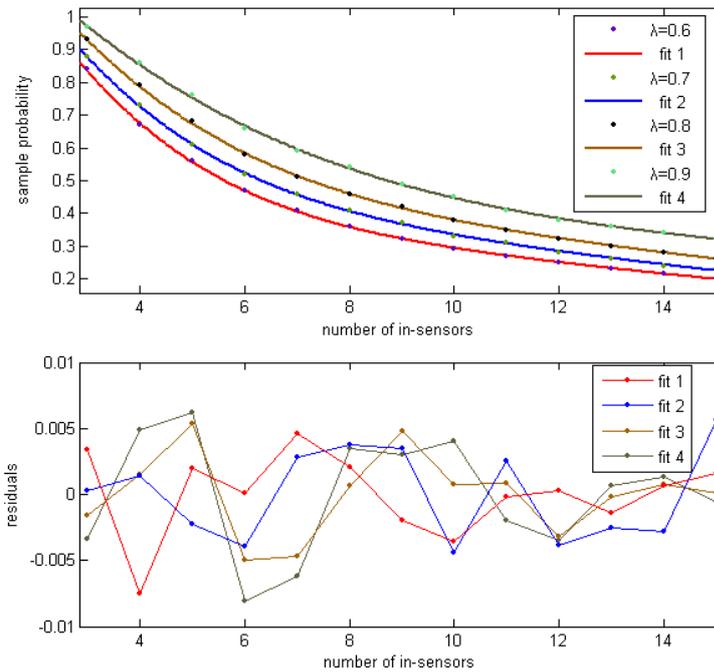


Fig. 6. Results of fitting

Table 1. Coefficients VS positional accuracy

Coefficients	Positional Accuracy			
	$\lambda = 0.6$	$\lambda = 0.7$	$\lambda = 0.8$	$\lambda = 0.9$
a_1	3.274	3.299	3.893	2.866
b_1	-7.647	-8.713	-10.66	-12.46
c_1	7.459	8.19	9.229	11.76
a_2	0.5796	0.6304	0.6999	0.7464
b_2	-12.8	-12.65	-14.05	-29.78
c_2	26.83	27.16	29.09	47.62

Accordingly, the wakeup probability can be computed as follows:

$$\eta = a_1 e^{-\frac{(n-b_1)^2}{c_1^2}} + a_2 e^{-\frac{(n-b_2)^2}{c_2^2}} \quad (4)$$

We derive (4) from MATLAB, where a_1 , a_2 , b_1 , b_2 , c_1 , c_2 are coefficients which might vary with λ . Simulation results show that Equation (4) is the most acceptable fitting, compared with other fitting formulas. Therefore, we use the same formula to fit others. Fig. 6 shows the results of fittings.

Using the data in Table 1, we can dynamically calculate the wakeup probability of the sensor nodes according to the number of in-sensors. That is, if positional accuracy λ is known, we can check Table 1 to derive a_1 , a_2 , b_1 , b_2 , c_1 , c_2 , and then we can get the wakeup probability using (4).

4.3. Sensor state adjustment

Sensor nodes are initialized to sleep state at the beginning. They will never wake up until the reception of wakeup probability from the User. The decision of wakeup is defined as the comparison between the wakeup probability and a random number between (0, 1).

When nodes wake up, they begin to measure the distance between the User and themselves and transmit it back to the User. After that, the User calls the localization algorithm, such as Trilateration Algorithm, Maximum Likelihood Estimation Method, etc. to compute the required location, using the returned data from the sensors, and then obtain its own location information.

5. Performance Evaluation

The performance of the EEOL scheme will be evaluated through a series of simulations. We use a bounded region of 100 x 100 m², in which nodes are placed using a uniform distribution. Additionally, we use an average value, 24mW, as the working power. A number of experiments have been conducted to assess the effectiveness of the proposed algorithm by using the MassMobility model. Table 2 lists the parameters used for simulation.

We compare our EEOL scheme with the existing method using duty cycle (DC) policy, in terms of the average energy consumption and the ratio of effective positioning.

We conduct simulations to locate the target using 100 sensor nodes whose communication range is set to 20m. As shown in Fig. 7 and Fig. 8, the energy consumption increases linearly with time. We can observe the energy consumption of DC is significant, which is nearly 80% larger than that of EEOL. This is mainly because the sensor nodes keep sleeping when the target is outside its sensing range. In Fig. 8, it is clear that the smaller the duty cycle, the less the energy consumption. Fig. 9 depicts the effective

positioning ratios of both EEOL and DC. As we can see, the ratio of effective positioning of EEOL is as large as 60%~90%. In contrast, when the duty cycle is 20%, the network of DC is only capable to locate the target in 20% of the total time. When the duty cycle is 2%, 5% or 10%, the ratio will be less than 5%.

Table 2. Simulation settings

Parameter	Value
Number of Nodes	25,50,75,100
Area Size	100 × 100 m ²
Simulation Time	1000s
Communication Range	20,30,40,50
Accuracy (λ)	0.6,0.7,0.8,0.9
Duty Cycle (δ)	2%,5%,10%,20%
Cycle	10s
Power	24mW
Speed	5
Change Angle	5

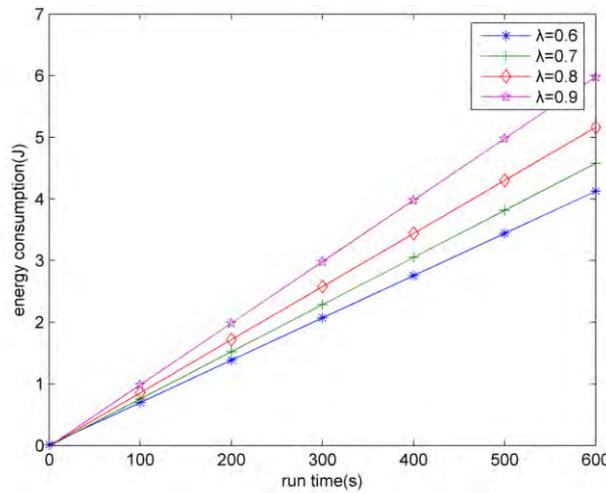


Fig. 7. Energy consumption with EEOL using 100 sensor nodes

5.1. Impact of number of sensor nodes

To examine the impact of different numbers of sensor nodes, we set the number of nodes to 25, 50, 75 and 100 respectively. We take $\lambda=0.8$ and

$\delta = 10\%$ as an example for EEOL and DC. Fig. 10 and Fig. 11 show the results of our simulation, where the communication range is 20m.

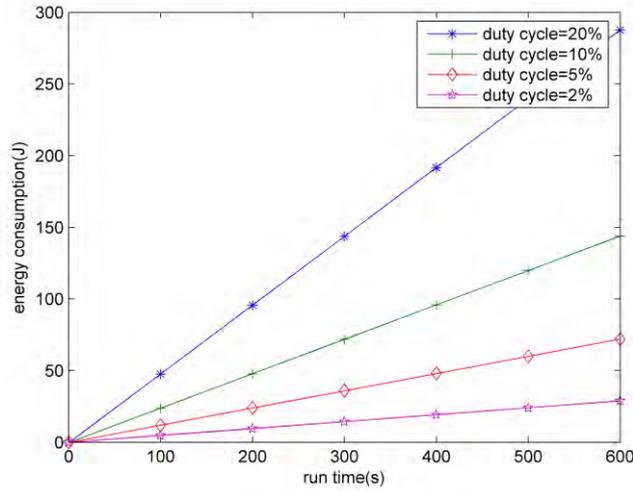


Fig. 8. Energy consumption with DC using 100 sensor nodes

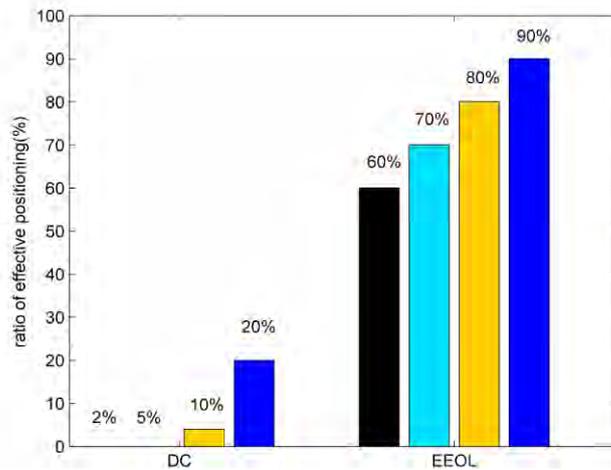


Fig. 9. Ratios of effective positioning for DC and EEOL

As shown in Fig. 10, we focus on the energy consumption at the moment of 200s and 400s. Compared with EEOL, DC performs worse, which can also be proved in Fig. 7 and Fig. 8. We can easily observe that with the increase of the number, the energy consumption drops down.

Fig. 11 shows the ratio of effective positioning with different number of sensors. Our EEOL algorithm significantly outperforms the DC method. With 25 to 75 nodes, the ratio of effective positioning of DC is close to 0, while it become up to 80% under EEOL. A low density of sensor nodes may lead to a low ratio of effective positioning.

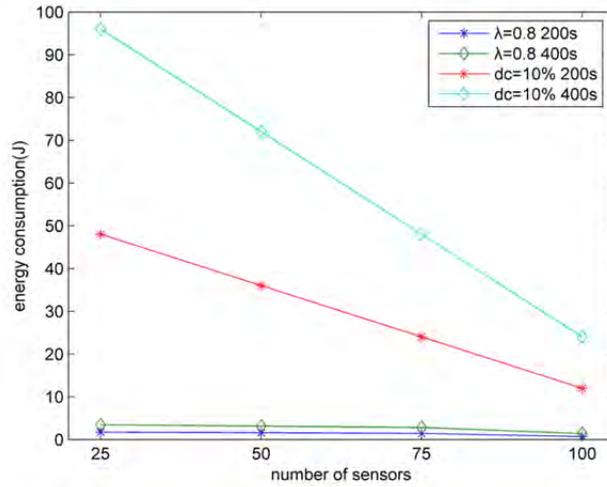


Fig. 10. Energy consumption versus number of sensor nodes

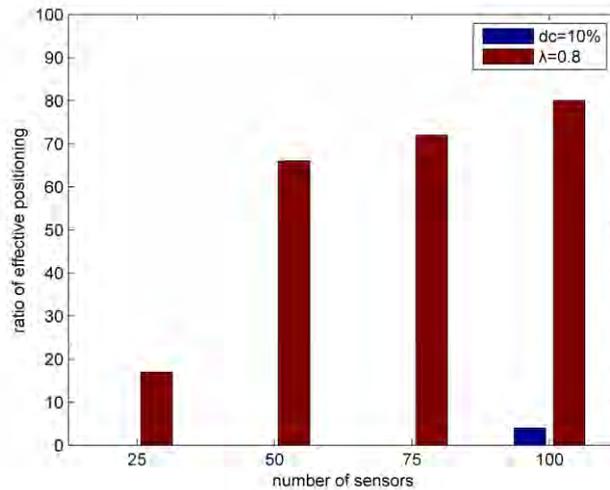


Fig. 11. Ratio of effective positioning versus number of sensor nodes

5.2. Impact of communication range

We conduct simulations with 100 sensor nodes, using different communication ranges. Here we also take $\lambda = 0.8$ as an example for EEOL, and focus on the energy consumption at the moment of 200s and 400s. Fig. 12 and Fig. 13 show the simulation results.

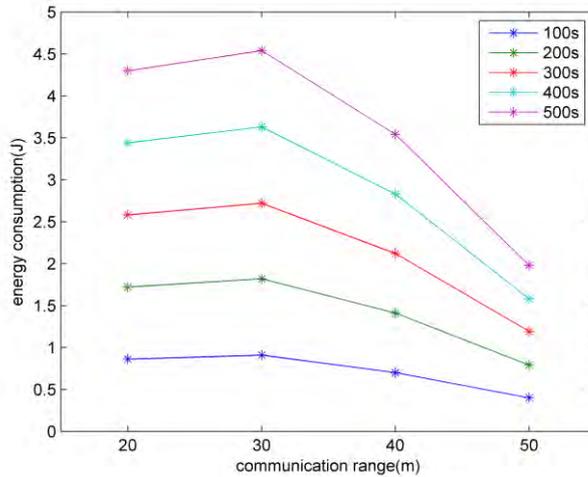


Fig. 12. Energy consumption versus communication range

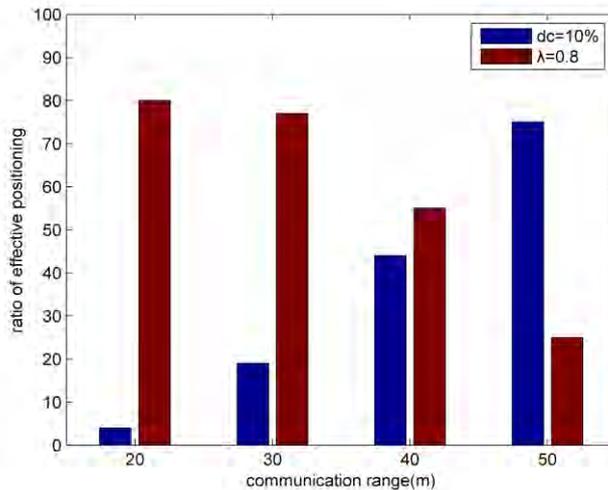


Fig. 13. Ratio of effective positioning versus communication range

It is clear that the communication range can not affect the energy consumption of DC, so we do not show it in Fig. 12. As we can see, the

energy consumption decreases as the communication range increases from 30m to 50m. Fig. 13 shows the ratios of effective positioning for different communication ranges. With the increase of the communication range, the ratio of effective positioning for EEOL will decrease. As for DC, the larger the duty cycle, the larger the ratio of effective positioning will be. This is reasonable since there are more sensor nodes in the sensing range of the target if the communication range is larger.

6. Conclusions

In this paper, an energy-efficient method for indoor localization (namely EEOL) has been proposed to reduce energy consumption, which can be easily applied in indoor localization systems. To balance the energy efficiency and the requirement of positional accuracy, we use opportunistic wakeup probability policy to wake up an appropriate numbers of sensor nodes. However, since most existing protocols need to wake up sensors periodically to perform positioning, certain amount of energy is still wasted by unnecessary sensors. As for localization, only two or three sensors are necessary based on the technique of localization to locate an object. Sensor nodes that are not triggered by opportunistic wakeup probability provided by the User keep sleeping. We avoid the communications between sensors, so that the sensor nodes are independent from each other, which will also make contribution to reducing the power consumption. We used MATLAB to simulate the relation between the number of sensor nodes and positional accuracy. By switching more sensors to sleep mode, the power is saved, and the lifetime of the localization network is prolonged. Simulation results are presented to demonstrate the performance of EEOL.

Possible further work includes the following topics: firstly, to produce a more practical investigation to satisfy the need of the positional accuracy; secondly, how to adapt our algorithm if there are multiple Users in a more complex environment; thirdly, how to tackle the issue of locating a mobile User if there are insufficient sensor nodes or network disconnection occurs.

Acknowledgement. This work was partially supported by the Natural Science Foundation of China under Grant No. 60903153, Zhejiang Provincial Natural Science Foundation of China under Grant No. Y108685, the Fundamental Research Funds for Central Universities (DUT10ZD110), the SRF for ROCS, SEM, and DUT Graduate School (JP201006).

References

1. Gu, Y., Lo, A., Niemegeers, I.: A survey of indoor positioning systems for wireless personal networks. *IEEE Communications Surveys & Tutorials*. vol.11, 13-32. (2009)

2. Desai, P., Rattan, K. S.: Indoor Localization and Surveillance using Wireless Sensor Network and Pan/Tilt Camera. Proceedings of the IEEE National Aerospace & Electronics Conference, 1-6. (2009)
3. Ahn, H., Rhee, S.: Simulation of a RSSI-Based Indoor Localization System Using Wireless Sensor Network. In Proc. 5th International Conference on Ubiquitous Information Technologies and Applications (CUTE), 1-4. (2010)
4. Huang, Y., Jheng, Y., Chen, H.: Performance of an MMSE based indoor localization with wireless sensor networks. In Proc. 6th International Conference on Networked Computing and Advanced Information Management (NCM), 671-675. (2010)
5. Soreanu, P., Volkovich, Z.: Energy-Efficient Circular Sector Sensing Coverage Model for Wireless Sensor Networks. 3rd International Conference on Sensor Technologies And Applications (Sensorcomm), 229-233. (2009)
6. Zhu, Y., Lionel M. Ni: Probabilistic Wakeup: Adaptive Duty Cycling for Energy-efficient Event Detection. Proc. International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), pp. 592-600, Crete, Greece, 360-367. (2007)
7. Mitilineos, S. A., Kyriazanos, D. M., Segou, O. E., Goufas, J. N., Thomopoulos, S.: Indoor Localization With Wireless Sensor Networks. Progress In Electromagnetics Research, Vol. 109, 441-474. (2010)
8. Wang, L., Hu, C., Wang, J., Tian, L., Meng, Q.: Dual-modal Indoor Mobile Localization System based on Prediction Algorithm. International Conference on Information and Automation (ICIA), 225-230. (2009)
9. Mitilineos, S. A., Goufas, J. N., Segou, O. E., Thomopoulos, S.: WAX-ROOM: an Indoor WSN-based Localization Platform. Proc. XIXth SPIE Conference on Signal Processing, Sensor Fusion and Target Recognition - SPIE Defense Security and Sensing 2010 Symposium, pp.1-5. (2010)
10. Schweinzer, H., Syafrudin, M.: LOSNUS: An ultrasonic system enabling high accuracy and secure TDoA locating of numerous devices. In Proc. International Conference on Indoor Positioning and Indoor Navigation (IPIN), 1-8. (2010)
11. Ren, H., Meng, Q.: Power Adaptive Localization Algorithm for Wireless Sensor Networks Using Particle Filter. IEEE Transactions on Vehicular Technology, Vol.58, 2498-2508. (2009)
12. You, C., Chen, Y., Chiang, J., Huang, P., Chu, H., Lau, S.: Sensor-Enhanced Mobility Prediction for Energy-Efficient Localization. In Proc. 3rd Annual IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON), 565-574. (2006)
13. Robles, J. J., Tromer, S., Quiroga, M., Lehnert, R.: A Low-Power Scheme for Localization in Wireless Sensor Networks. In Proceedings of EUNICE'2010, 259-262. (2010)
14. Robles, J. J., Tromer, S., Quiroga, M., Lehnert, R.: Enabling low-power localization for mobile sensor nodes. In Proc. International Conference on Indoor Positioning and Indoor Navigation (IPIN), 1-10. (2010)
15. Gribben, J., Boukerche, A., Pazzi, R.: Scheduling for Scalable Energy-Efficient Localization in Mobile Ad Hoc Networks. In Proc. 7th Annual IEEE Conference on Sensor Mesh and Ad Hoc Communications and Networks (SECON), 1-9. (2010)
16. Zhang, D., Xia, F., Yang, Z., Yao, L., Zhao, W.H.: Localization Technologies for Indoor Human Tracking. In Proc. 5th International Conference on Future Information Technology (FutureTech), 1-6. (2010)

Feng Xia, Xue Yang, Haifeng Liu, Da Zhang and Wenhong Zhao

17. Francesco, Z., Kang G., Tanguy, P., Andrea, Z.: Opportunistic localization scheme based on Linear Matrix Inequality. In Proc. IEEE International Symposium on Intelligent Signal Processing, 247-252. (2009)

Feng Xia is an Associate Professor and Ph.D Supervisor in School of Software, Dalian University of Technology, China. He is the (Guest) Editor of several international journals. He serves as General Chair, PC Chair, Workshop Chair, Publicity Chair, or PC Member of a number of conferences. Dr. Xia has authored/co-authored one book and over 100 papers. His research interests include cyber-physical systems, mobile and social computing, and intelligent systems. He is a member of IEEE and ACM.

Xue Yang received B.E. degree from Dalian University of Technology, China, in 2011. Currently she is a Master student. Her research interests include mobile computing and wireless sensor networks.

Haifeng Liu received B.E. degree from Dalian University of Technology, China, in 2009. Currently he is a Master student in School of Software, Dalian University of Technology. His research interests include mobile computing and green computing.

Da Zhang received B.E. degree from Dalian University of Technology, China, in 2010. Currently she is a Master student. Her research interests include mobile computing and wireless sensor networks.

Wenhong Zhao is a Professor in College of Mechanical Engineering, Zhejiang University of Technology, China. His research interests include embedded systems, wireless sensor networks, and automatic control.

Received: April 6, 2011; Accepted: June 1, 2011.

An Energy Efficient and Load Balancing Routing Algorithm for Wireless Sensor Networks

Jin Wang^{1,2}, Tinghuai Ma¹, Jinsung Cho², and Sungoung Lee²

¹ Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, 210044, Nanjing, China
{wj0514, ma_tinghuai}@hotmail.com

² Department of Computer Engineering, Kyung Hee University,
449-701, Yongin City, South Korea
chojs@khu.ac.kr; sylee@oslab.khu.ac.kr

Abstract. Many energy aware routing algorithms and protocols have been proposed for wireless sensor networks recently to achieve aims like minimum energy consumption, maximized network lifetime, reduced communication latency and overhead etc. The problem of hotspot can not be well addressed under many routing algorithms since some nodes which are on the shortest path or close to the base station tend to deplete their energy quickly and consequently cause network partition. In this paper, we propose a Ring-based Energy Aware Routing (REAR) algorithm for wireless sensor networks which can achieve both energy balancing and energy efficiency for all sensor nodes. Our algorithm considers not only the hop number and distance but also the residual energy of the next hop node during routing process. Simulation results validate that our algorithm outperforms some other routing algorithms in the aspects of energy consumption and network lifetime etc.

Keywords: wireless sensor networks, hop number, energy efficiency, energy balancing, network lifetime.

1. Introduction

Wireless sensor networks (WSNs) are composed of huge number of sensor nodes which can monitor the environment by collecting, processing as well as transmitting collected data to the remote sink node through direct or multi-hop transmission. WSNs have attracted lots of attention in recent years due to their wide applications such as battlefield surveillance, inventory and wildlife monitoring, smart home and healthcare etc [1].

Since the tiny sensor nodes are powered by limited battery resources, energy efficiency is one of the primary challenges to the successful application of WSNs. Usually energy is consumed during three processes which are sensing, processing and communication process. Here, we only

focus on energy consumption during communication process since it prevails over the other two processes.

Many energy aware routing algorithms or protocols have been proposed for WSNs in recently years [2-22]. Among these routing algorithms, techniques like data aggregation [3-5], clustering [6-11], evolutionary based algorithms [12-15] are adopted to achieve better energy efficiency. However, many of these algorithms aim to minimize metrics such as energy consumption, latency during routing process, which will cause certain hotspot nodes as well as partitioned network area due to the overuse of certain nodes on shortest path or close to the BS. Since the network lifetime is usually defined as the time when the first node dies from lack of energy, huge amounts of energy will be wasted by the remaining sensor nodes when the first node dies.

To efficiently mitigate the hotspot problem which is caused by imbalanced consumption of energy among sensors, the network metrics like hop number, hop distance as well as remaining energy need to be carefully considered. In fact, these network metrics have very important impact on network performance such as energy consumption, network lifetime, routing overhead, latency and interference etc [16]. Intuitively, if the hop number is too large, the energy consumption can be reduced at the cost of long end-to-end latency and large control overhead. If the hop number is too small (e.g., direct transmission), the latency will be very small while the energy consumption can be very large due to the nature of long distance wireless communication. Thus, an optimal hop number with suitable individual distance can be deduced to achieve energy reduction and energy balancing. The authors in [17-19] study the energy consumption from a new viewpoint by studying the transmission manner which proves to be effective.

Based on our previous work, we try to achieve both energy efficiency and energy balancing from hop and distance based point of view in this paper. We first deduce a suitable hop number with individual distance based on our theoretical analysis of energy and traffic models. Then, we propose our Ring-based Energy Aware Routing (REAR) algorithm which considers not only hop number and hop distance but also the residual energy during routing process.

The remainder of the paper is organized as follows. Section 2 presents some related work. Section 3 gives the theoretical deduction of hop number and hop distance based on the energy and traffic models. In Section 4, the REAR algorithm is proposed and Section 5 provides extensive simulation results. Section 6 concludes this paper and gives some future work.

2. Related Work

A survey about different routing protocols in sensor networks is given in [2] which classifies traditional routing algorithms into three types, namely data-centric, hierarchical and position-based routing algorithms. Direct Diffusion [3] is viewed as a representative data-centric routing protocol for flat structure

WSNs. The data generated by sensor nodes is named by attribute-value pairs. Once a sink node inquires certain type of information, it will send a query and the observed data will be aggregated and transmitted back to the sink node. Rather than always using the lowest energy paths, the authors in [4] use sub-optimal paths occasionally so that the network lifetime is increased by 40% compared to [3]. In [5], the authors propose a centralized and decentralized routing protocols named UBERP by carefully selecting the transmission path with residual energy larger than certain threshold.

Hierarchical routing protocols [6-11] are very suitable for WSNs since they can not only provide good scalability but also perform data fusion by each cluster head. LEACH [6] can prolong network lifetime to 8-fold more than other ordinary routing protocols. However, 5% of cluster head nodes are randomly chosen and cluster head nodes use direct transmission. PEGASIS [7] is viewed as an improved version of LEACH. It is a chain based routing protocol which can save more energy compared to LEACH. HEED [8] considers the residual energy as the primary parameter and a secondary parameter like node's degree etc. ERA [9] is similar to [6] during route setup phase while energy balancing is achieved during cluster head association phase since each node selects its cluster head with maximum residual energy. DHAC [10] provides a simple six-step bottom-up clustering method rather than traditional top-down methods with better network lifetime performance. The authors in [11] try to distribute energy load among all sensors in order to achieve both energy efficiency and lifetime maximization.

An improved version of LEACH is presented in [12] to improve energy efficiency and system stability by using genetic algorithm (GA) during the selection of cluster heads. In [13], each swarm agent can carry and exchange the residual energy information during route selection process to maximize network lifetime in ad hoc and sensor networks. In [14], an improved ant colony optimization (ACO) method is applied to the communication network routing problem with better performance in terms of hop number. An energy balanced unequal clustering protocol is proposed in [15] with particle swarm optimization technique so that the hot-spot problem is avoided and network lifetime is prolonged.

The authors in [17] present some pioneering work by studying different energy models under general wireless network environment. In [18], the authors use a probability of P_i to transmit data through multi-hop manner and a probability $(1 - P_i)$ to transmit through single hop to sink node. The authors in [19] also study the energy consumption under both single hop and multi-hop transmission. They claim that the preference of multi-hop routing to single hop routing depends on source to sink distance and reception cost. In [20,21], the authors study the energy consumption from hop and distance point of view and propose a hop-based energy aware routing algorithm which can reduce energy consumption and prolong network lifetime effectively. The authors in [22] consider the hot-spot phenomenon and propose a load balancing data gathering algorithm which classifies sensors into different layers based on their distance to sink node.

In this paper, we aim to achieve both energy efficiency and balancing by building BS oriented ring structure in a centralized manner from BS side. The BS determines the ring size as well as final route from source node to BS.

3. Theoretical Analysis

3.1. Network Model

The traditional WSNs can be viewed as an undirected graph $G = \langle V, E \rangle$ where V represents the set of vertices and E represents the set of edges. We assume there are N nodes randomly placed in an area $[X, Y]$. There exists a link $E(i, j)$ between node i and node j if the Euclidean distance $d(i, j)$ is not larger than the radio transmission radius R . Here, undirected graph means bi-directional communication link. In other words, if node j can receive packet from its neighboring node i , it is believed that node i can receive packet from node j in a reverse way. The objective in this paper is to find a set of optimal or sub-optimal individual distances during routing process so that the energy is consumed at similar rate for all involved sensors.

3.2. Energy Model

The first order radio model is commonly used as an energy consumption model [6, 8, 17, 20]. Based on this model, radio consumes E_{Tx} amount of energy to transmit a l bits message over a distance of d :

$$E_{Tx}(l, d) = \begin{cases} l \cdot E_{elec} + l \cdot \epsilon_{fs} \cdot d^2, & \text{if } d < d_0 \\ l \cdot E_{elec} + l \cdot \epsilon_{mp} \cdot d^4, & \text{if } d \geq d_0 \end{cases} \quad (1)$$

and E_{Rx} amount of energy to receive this message:

$$E_{Rx}(l) = l \cdot E_{elec} \quad (2)$$

and E_{Fk} amount of energy to forward this message:

$$E_{Fk}(l, d) = E_{Tx}(l, d) + E_{Rx}(l) = \begin{cases} 2l \cdot E_{elec} + l \cdot \epsilon_{fs} \cdot d^2 \\ 2l \cdot E_{elec} + l \cdot \epsilon_{mp} \cdot d^4 \end{cases} \quad (3)$$

Definition of radio parameters above are the same as [6, 8, 20] etc.

3.3. Theoretical Analysis

For simplicity, we first study energy consumption under one dimensional linear network which can be used in linear applications such as highway monitoring, congestion control etc.

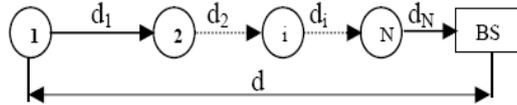


Fig. 1. One dimensional linear network

In Fig.1., there are N sensors randomly placed along a line from source node 1 to the BS with individual distance $\{d_1, d_2, \dots, d_i, d_N\}$, and $\sum_{i=1}^N d_i = d$.

Now, we suppose each node takes turn to transmit its l bits message via direct or multi-hop transmission to the BS. In Fig. 1. we find that node N will forward the data for $(N-1)$ times and node i will forward $(i-1)$ times. If we let $d_1 = d_2 = \dots = d_N = d/N$, node N will become hotspot node and dies quickly. Here, our objective is to find the set of individual optimal distance d_1, d_2, \dots, d_N as well as the optimal or sub-optimal hop number N so that $E_1 = E_2 = \dots = E_N$. In the mean time, we try to let each involved nodes consume the least energy in order to achieve energy efficiency.

For certain forwarding node i , the energy consumed for both receiving and then transmitting the data will be:

$$E_i = l \cdot (E_{elec} + \varepsilon_{amp} \cdot d_i^\alpha) + l \cdot (i-1)(2E_{elec} + \varepsilon_{amp} \cdot d_i^\alpha) = l \cdot (2i-1)E_{elec} + l \cdot i \cdot \varepsilon_{amp} \cdot d_i^\alpha \quad (4)$$

Here, $\varepsilon_{amp} = \varepsilon_{fs}$ when $\alpha = 2$ and $\varepsilon_{amp} = \varepsilon_{mp}$ when $\alpha = 4$.

Let $E_{i+1} = E_i$, we can finally get:

$$d_{i+1} = \sqrt[\alpha]{\frac{-2E_{elec} + i\varepsilon_{amp}d_i^\alpha}{\varepsilon_{amp}(i+1)}} = \sqrt[\alpha]{\frac{-2iE_{elec} + \varepsilon_{amp}d_1^\alpha}{\varepsilon_{amp}(i+1)}} \quad (5)$$

Since $d_n > 0$, it must satisfy:

$$d_1 > \sqrt[\alpha]{\frac{2(n-1)E_{elec}}{\varepsilon_{amp}}} \quad (6)$$

Given multi-hop number n , we can get the lower bound of distance value d_1 as well as the minimal source to sink node distance $d = \sum d_i$. Here, we let $E_{elec} = 50nJ/bit$, $\alpha = 4$ and $\varepsilon_{amp} = 0.001pJ/bit/m^4$.

On the other hand, given the source to sink node distance d , there might be several multi-hop routes with different hop number n . For example, when $d = 300$, we can use either 2-hop or 3-hop route to achieve energy balancing. If the hop number $n \geq 4$, the minimal $d = \sum d_i \geq 307.6$ which is contrary to $d = 300$. Thus, there exists a highest hop number route with minimal energy consumption for each sensor node and this is the multi-hop route we need. For example, when $d = 800$, we can either choose an 8-hop route with $d_1(8) = 164.8$ or choose one 7-hop route with $d_1(7) = 170.5$. The corresponding individual distance d_i can be deduced from Equation (5).

From the analysis of energy consumption above, we can see that the suitable multi-hop route with hop number n as well as corresponding individual distance d_i can be achieved to gain energy balancing with $E_1 = E_2 = \dots = E_N$ when the source to BS distance d is given. In the mean time, we try to find multi-hop route with more intermediate nodes involved in order to gain energy efficiency. Thus, our REAR algorithm below can be both energy balancing and efficient if the routing algorithm is carefully designed.

4. Our REAR Algorithm

4.1. Basic Assumptions

We make the following basic assumptions:

- 1) All sensor nodes are static and homogeneous after deployment.
- 2) The communication links are symmetric.
- 3) Each sensor node has several power levels which they can adjust.
- 4) Each sensor node can know the distance to its neighbors and to the BS.
- 5) There is no obstacle between nodes.

In this paper, we do not consider mobile sink nodes or mobile sensor nodes and all the homogeneous sensors are randomly deployed and left unattended after deployment. We assume the link is symmetric which means if node i can receive a message from its neighbor node j then node j can also get a message from node i . As is shown in many papers, current sensor nodes can have several transmission power levels so that they can dynamically adjust its power to its neighboring node to save energy. By adopting some localization or positioning techniques, the sensors can know

its relative distances to its neighbors as well as to the BS. Also, we do not consider obstacles between nodes even though our algorithm can avoid this phenomenon by choosing another alternative neighboring node as a solution. It is worth noting that we make no assumption of the uniform distribution of sensor nodes or the knowledge of global network topology here.

4.2. Route Setup Phase

Once some source node has data to send, it will select its next hop node and try to set up a route from itself to the BS as follows.

Route Setup Phase Algorithm

Begin

- 1: Source node i has data to send to BS
- 2: *if* $d_{i,BS} < \sum d_i(1) = 100$ *then*
- 3: direct transmission
- 4: *else*
- 5: Source node i broadcasts multi-hop request to BS
- 6: BS determines optimal hop number n and distance $[d_1 : d_n]$
- 7: BS builds ring structure with
- 8: $r_1 = d_1$
- 9: $r_2 = d_1 + d_2$
- 10: *until* $r_n = d_1 + \dots + d_n = d_{i,BS}$
- 11: BS classifies sensors into different levels based on ring size
- 12: *if* $d_{k,BS} \leq r_1$ *then*
- 13: node $k \in$ Level 1
- 14: *elseif* $r_{k-1} < d_{k,BS} \leq r_k$ ($k \geq 2$)
- 15: node $k \in$ Level k
- 16: *end*
- 17: BS determines the final route of node i as follows
- 18: i chooses its neighbors set A with $d_{i,j} \in (d_n, d_n + \Delta]$, $j \in A$
- 19: i chooses its neighbors sub-set $B \in A$ where node $k \in$ Level $(n-1)$
- 20: i chooses its final neighbor $j^* \in C$ with maximal residual energy
- 21: node j^* finds its final next hop in an iterative way like node i until BS
- 22: BS send the final multi-hop route with individual nodes to source node i
- 23: *end - if*
- 24: Source node i start sending its data to BS based on the route table

End

Fig. 2. Route setup phase algorithm

As is shown in Fig. 2., the source node will first determine the transmission manner. Namely, if the source to BS distance $d < \sum d_i(1)$, source node will use direct transmission (line 3) to send its data to the BS. Or else, it will broadcast a multi-hop request to BS (line 5).

When BS receives the multi-hop request from source node, it will determine the final multi-hop route with the optimal number n and individual distance $\{d_1 \dots d_N\}$ (line 6), based on the source to BS distance. Then, it will build a ring structure with different ring size (line 8-10). Next, it will classify sensor nodes into different levels based on ring size (line 11-16).

Once the sensors are associated with different levels, BS will determine the final multi-hop route as follows. First, it will choose some candidate next hop nodes of source node with distance $d_{i,j} \in (d_n, d_n + \Delta]$ (line 18). Here, Δ is used under practical random network topology. Within these candidates, BS will choose those which belong to level $(n-1)$ to make progress from source to BS (line 19). Finally, BS will choose the one from level $(n-1)$ with maximal remaining energy as the final next hop node (line 20).

BS will perform the same process in a similar way for the next hop node until a multi-hop route from source to BS is built (line 21). Finally, BS will send the complete multi-hop route information to the source node (line 22). Source node will start the transmission of its data when it receives the complete multi-hop route information (line 24).

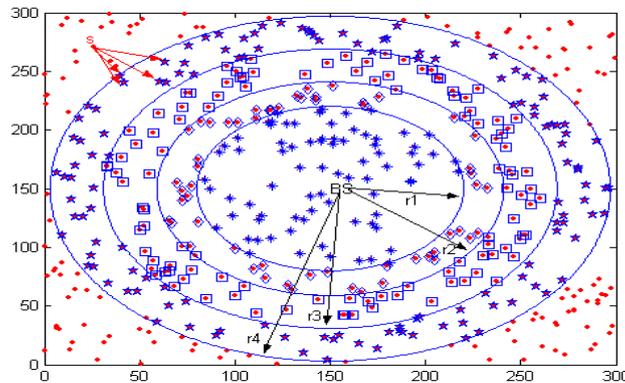


Fig. 3. BS oriented ring structure in WSNs

It is worth noting that the multi-hop route is built by BS in a centralized way due to the fact that BS has more powerful resources such as memory, computation and communication etc. We treat the hop number and individual distance as the primary metric during the selection of next hop. In the mean time, we treat the remaining energy as the secondary metric during the selection of the final nodes from many candidates (line 20).

From Fig. 3. we can see the building of BS oriented ring structure multi-hop route more clearly. BS will determine the multi-hop route from source node until itself as follows. First, source nodes selects its candidate neighbors which belong to level $(n-1)$ with distance $d_{i,j} \in (d_n, d_n + \Delta]$. Then, it will choose the one with maximal remaining energy as the final next hop node. This process will continue in an iterative way until the BS.

4.3. Route Maintenance Phase

A link failure is usually caused by reasons like the depletion of energy, physical damage, and mobility of certain nodes or BS etc. Due to the centralized routing nature in our REAR algorithm, BS will always choose the node with high remaining energy to ensure the multi-hop route reliability. In other words, the link failure probability is relatively low until most of the nodes run out of energy. From the simulation part, we can also validate this point.

5. Experimental Results

5.1. Simulation Environment

There are N nodes randomly deployed in a WSNs area. The BS is placed either inside or outside the monitoring area. In each round, each sensor node takes turn to transmit their 2000 bits message to the BS with either direct transmission or multi-hop transmission. Some of the simulation parameters are listed in Table 1.

Table 1. Simulation Environment

Parameter	Value
Network size	$300 \times 300 m^2$
Number of nodes	300
Transmission radius R	[50, 140] m
Initial energy	2 J
BS location	inside and outside
Data size	2000 bits
Δ	[20, 40] m

We compare our REAR algorithm with three other existing algorithms which are direct transmission, greedy algorithm and max-remaining energy (MRE) algorithm. In direct transmission algorithm, all sensor nodes simply

transmit their message directly to the BS. This algorithm is simple and energy efficient when the network scale is small. In greedy algorithm, each node prefers to choose its neighbor which is closest to the BS as its next hop neighbor. This algorithm can be energy efficient when the transmission radius is carefully designed. In MRE algorithm, the neighboring node with high residual energy will be chosen as next hop in order to prolong network lifetime. It is worth noting that some of these factors such as distance and residual energy can be jointly considered which can further reduce energy consumption and prolong network lifetime.

5.2. Study of Hop Number

Fig. 4. shows the average hop number for 4 algorithms where BS is placed in the middle of the area. The transmission radius R is varying from 50 to 140 meters and $\Delta = 40$.

We can see from Fig. 4. that the average hop number decreases as the transmission radius R increases. Greedy and REAR algorithms have almost the same performance when $R \leq 100$. When $220 \leq R \leq 140$, our REAR algorithm is better than greedy algorithm and when $R \geq 140$, greedy algorithm has a shorter hop number than REAR at the cost of more energy consumption.

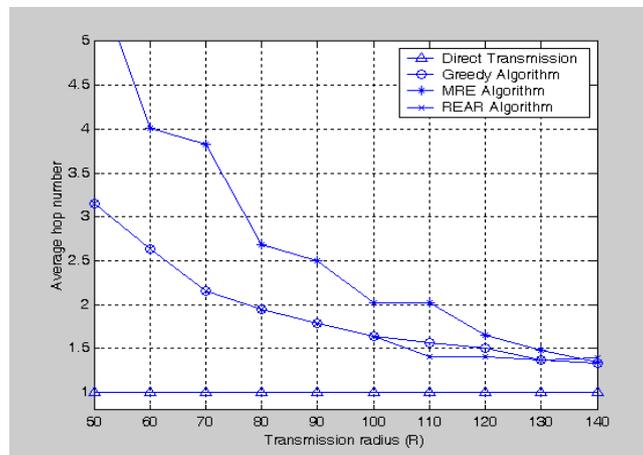


Fig. 4. Average hop number under different R

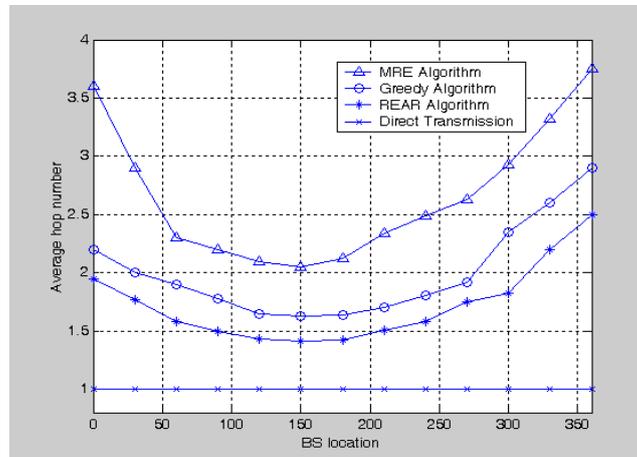


Fig. 5. Average hop number under different BS location

Fig. 5. shows the average hop number for 4 algorithms under the same simulation environment as Fig. 4., where BS moves along the diagonal line from position (150, 0) to (150, 360) with step size 30.

From Fig. 5. we can see that it is nearly symmetric based on line $x = 150$ as BS moves from (150, 0) to (150, 360). When BS moves from (150, 150) to (150, 360), the average hop number increases since the average source to BS distance is getting larger. It is worth noting that the x label value in Fig. 5. means the y coordinate of BS location here.

5.3. Study of Energy Consumption

Fig. 6. shows the energy consumption under different source to BS distance d with similar simulation environment to Fig. 4 and 5. Here, the BS is placed in the middle of the area. The transmission radius $R = 110$ and $\Delta = 20$.

From Fig. 6. we can see that when $d \leq R = 110$, direct transmission manner can be chosen by all these 4 routing algorithms and they have almost the same energy consumption which is also very small. When $d > 110$, direct transmission is not possible for the other 3 routing algorithms except direct transmission. Direct transmission consume the largest energy since multi-path model is used under which power attenuates in the fourth order of distance. The performance of MRE and greedy algorithms are in the middle while our REAR algorithm consumes the least energy.

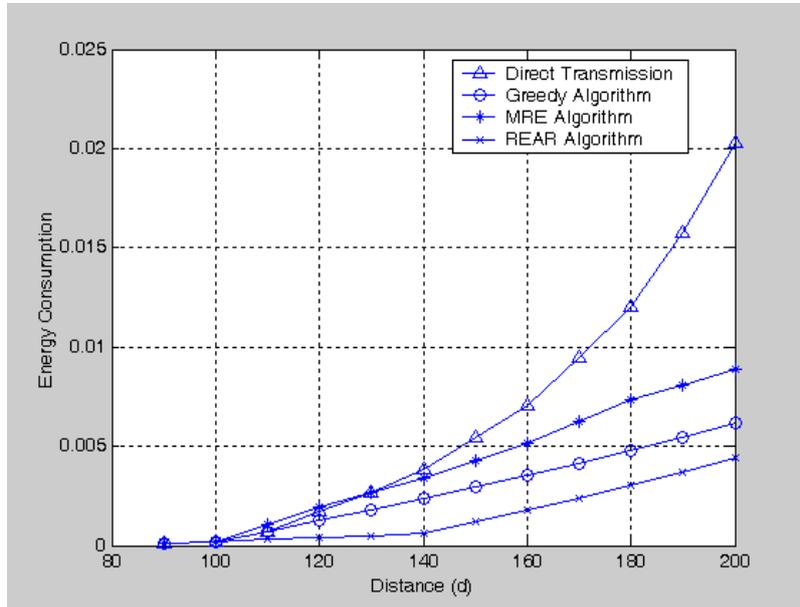


Fig. 6. Energy consumption under different d

In Fig. 7. we study the energy consumption under different BS location. The simulation environment is also similar to Fig. 4 to 6. where BS moves along the diagonal line from position (0, 0) to (300, 300).

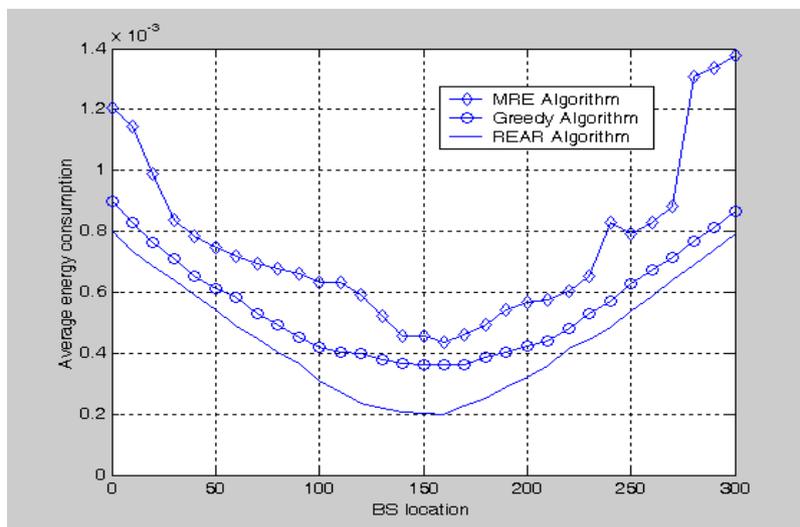


Fig. 7. Energy consumption under different BS location

From Fig. 7. we find that the distribution of energy consumption for the 3 algorithms is almost symmetric based on line $x = y$ and the minimal energy consumption can be achieved if BS is placed at (150, 150) (middle of WSNs). It is easy to understand the symmetry property from energy consumption model since the average energy consumption tends to get the minimum value when BS is located at the center of network area. It is worth noting that we do not compare with the direct transmission algorithm since it is relatively large and the symmetry property is not very clear as BS moves.

5.4. Study of Average Network Lifetime

We define the network lifetime as the time when the first sensor node dies out of energy. We compare the average network lifetime under similar network environment to Fig. 4. to 7. The simulation is done under 100 different network topologies to see an average performance and $R = 110$ and $\Delta = 20$.

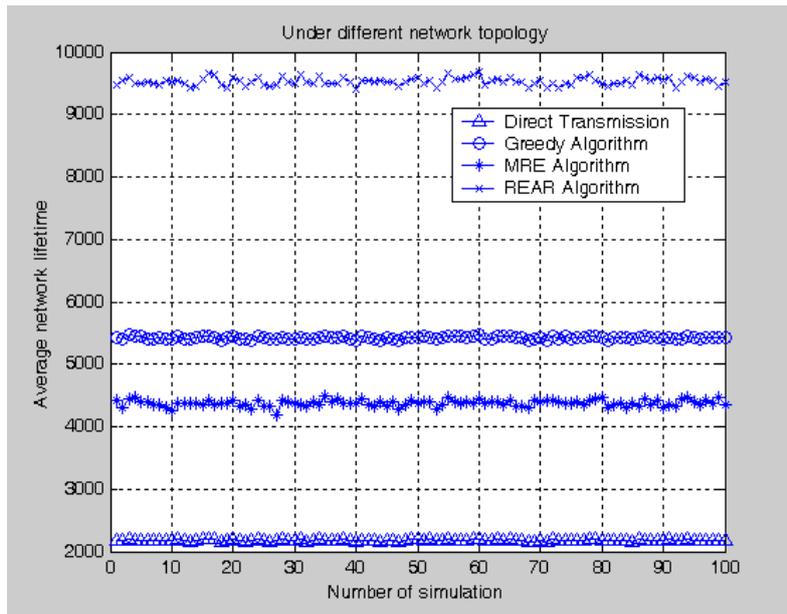


Fig. 8. Average network lifetime under different topology

From Fig. 8. we can see that our REAR algorithm has the longest lifetime while direct transmission algorithm has the worst average network lifetime. The reason lies in the average energy consumption mechanisms of each algorithm, as we have explained. It is worth noting that our REAR algorithm

has a factor of 2 to 4 times longer network lifetime than the other 3 routing algorithms on average.

From the simulations results in Fig. 4 to 7, we can see that our REAR algorithm can achieve both energy balancing and energy efficiency as comparing with other routing algorithms. We only focus on studying energy efficient routing mechanism from appropriate hop number and distance point of view in this paper and do not consider clustering and data fusion here.

6. Conclusions and Future Work

We propose a ring-based energy aware routing (REAR) algorithm for WSNs in this paper which can achieve both energy efficiency and balancing from hop number and distance point of view. Given the source to sink node distance, the multi-hop number and corresponding individual distance can be determined so that all sensor nodes can consume energy at a similar rate. During the routing process, we consider the hop number and distance as the primary factor and the residual energy as the secondary factor. Simulation results show that our REAR algorithm is superior to some existing routing algorithms in terms of energy hop number, consumption as well as network lifetime on average.

For future research, we plan to extend our work by exploring the effect of hop number and hop distance on other network metrics such as latency, communication overhead etc. Also, we plan to further extend network lifetime by combining clustering mechanism with our REAR algorithm in the future.

Acknowledgement. This work was supported by a post-doctoral fellowship grant from the Kyung Hee University Korea in 2011 (KHU-20110219). This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency)" (NIPA-2011-(C1090-1121-0003)). Professor Sungyoung Lee is the corresponding author. The authors are very thankful to the anonymous reviews for their constructive comments and suggestions which greatly improve the quality of this paper.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: A survey. *Comput. Netw.* Vol. 38, 393-422. (2002)
2. Akkaya, K., Younis, M.: A Survey of Routing Protocols in Wireless Sensor Networks. In Elsevier Ad Hoc Network, Vol. 3 (3), 325-349. (2005)
3. Intanagonwiwat C., Govindan R., Estrin D.: Directed diffusion: A scalable and robust communication paradigm for sensor networks. In Proceedings of the 6th International Conference on Mobile Computing and Networking, 56-67. (2000)

4. Shah, R.C., Rabaey, J.M.: Energy Aware Routing for Low Energy Ad hoc Sensor Networks. In Proceedings of IEEE Wireless Comm. and Netw. 350-355. (2002)
5. Zytoune, O., El aroussi, M., Aboutajdine D.: A Uniform Balancing Energy Routing Protocol for Wireless Sensor Networks. *Wireless Personal Communications*, Vol. 55, 1477-161. (2010)
6. Heinzelman W., Chandrakasan A., Balakrishnan H.: Energy-efficient communication protocol for wireless microsensor networks. In Proceeding of the Hawaii International Conference System Sciences, 8020-8026. (2000)
7. Lindsey S., Raghavendra C. S.: PEGASIS: Power Efficient GATHERing in Sensor Information Systems. In Proceedings of the IEEE Aerospace Conference. (2002)
8. Younis, O., Fahmy, S.: HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad hoc Sensor Networks. *IEEE Trans. on Mobile Comput.*, 366-379. (2004)
9. Chen, H., Wu, C.S., Chu, Y.S., Cheng, C.C., Tsai, L.K.: Energy residue aware (ERA) clustering algorithm for leach-based wireless sensor networks. In the 2nd international Conference on Systems and Networks Communications, 40. (2007)
10. Lung, C.H., Zhou, C.: Using hierarchical agglomerative clustering in wireless sensor networks: An energy-efficient and flexible approach. *Ad Hoc Networks*, 328-344. (2010)
11. Bajaber, F., Awan, I.: Energy efficient clustering protocol to enhance lifetime of wireless sensor network. *J. Ambient Intell Human Comput*, Vol 1, 239-248. (2010)
12. Li, X., Xu, L., Wang, H., Song, J., Yang, S.X.: A Differential Evolution-Based Routing Algorithm for Environmental Monitoring Wireless Sensor Networks. *Sensors*, Vol. 10, 5425-5442. (2010)
13. Yang, H., Ye, F., Sikdar, B.: A Swarm-Intelligence-Based Protocol for Data Acquisition in Networks with Mobile Sinks. *IEEE Trans. on Mobile Computing*, Vol. 7, 931-945. (2008)
14. Zhao, D., Luo, L., Zhang K.: An improved ant colony optimization for the communication network routing problem. *Mathematical and Computer Modelling*, Vol. 52, 1976-1981. (2010)
15. Jiang, C. J., Shi, W.R., Xiang, M., Tang X.L.: Energy-balanced unequal clustering protocol for wireless sensor networks. *The Journal of Universities of Posts and Telecommunications*, Vol. 17, 94-99. (2010)
16. Haenggi, M.: Twelve Reasons not to Route over Many Short Hops. In Proceedings of the 60th IEEE Vehicular Technology Conference (VTC), 3130-3134. (2004)
17. Stojmenovic, I., Lin, X.: Power-aware Localized Routing in Wireless Networks. *IEEE Transaction on Parallel and Distributed Systems*, Vol.12, 1121-1133. (2001)
18. Efthymiou, C., Nikolettas, S., Rolim, J.: Energy Balanced Data Propagation in Wireless Sensor Networks. *Wireless Networks*, Special Issue: Selected Papers from WMAN'04, Vol. 12, 691-707. (2006)
19. Szymon, F., Martin, C.: On the Problem of Energy Efficiency of Multi-hop vs One-hop Routing in Wireless Sensor Networks. In the 21st International Conference on Advanced Information Networking and Applications, 380-385. (2007)
20. Wang, J., Cho, J., Lee, S. L., Chen, K.C., Lee, Y. K.: Hop-based Energy Aware Routing Algorithm for Wireless Sensor Networks. In the IEICE Transactions on Communications, Vol.E93-B, No. 2, 305-316. (2010)

Jin Wang, Tinghuai Ma, Jinsung Cho, and Sungoung Lee

21. Wang, J., Kim, J.U, Shu, L., Niu, Y., Lee, S.L.: A Distance-Based Energy Aware Routing Algorithm for Wireless Sensor Networks, *Sensors*, ISSN: 1424-8220, 9493-9511. (2010)
22. Guan, X., Guan, L., Wang, X.G., Ohtsuki, T.: A new load balancing and data collection algorithm for energy saving in wireless sensor networks. *Telecommunication Systems*, Vol. 45, 313–322. (2010)

Jin Wang received the B.S. and M.S. degree in the Electrical Engineering from Nanjing University of Posts and Telecommunications, China in 2002 and 2005, respectively. He received Ph.D. degree in Ubiquitous Computing laboratory in the Computer Engineering Department of Kyung Hee University Korea in 2010. Now, he is a professor in Nanjing University of Information Science & technology. His research interests include routing protocol and algorithm design, analysis and optimization, and performance evaluation for wireless ad hoc and sensor networks.

Tinghuai Ma is an associate professor in Computer Sciences at Nanjing University of Information Science & Technology, China. He received his Bachelor (HUST, China, 1997), Master (HUST, China, 2000), PhD (Chinese Academy of Science, 2003) and was Post-doctoral associate (AJOU University, 2004). From Nov.2007 to Jul. 2008, he visited Chinese Meteorology Administration. From Feb.2009 to Aug. 2009, he was a visiting professor in Ubiquitous computing Lab, Kyung Hee University. His research interests are in the areas of Data Mining and Privacy Protected in Ubiquitous System, Grid Computing. His research interests are data mining, grid computing, ubiquitous computing, privacy preserving etc. He has published more than 70 journal/conference papers. He is principle investigator of several NSF projects. He is a member of IEEE.

Jinsung Cho received his B.S., M.S, and Ph.D. degrees in computer engineering from Seoul National University, Korea, in 1992, 1994, and 2000, respectively. He was a visiting scholar at IBM T.J. Watson Research Center in 1998 and a research staff at SAMSUNG Electronics in 1999~2003. Currently, he is an assistant professor of Department of Computer Engineering at Kyung Hee University, Korea. His research interests include mobile networking & computing and embedded systems & software.

Sungyoung Lee received his B.S. from Korea University, Seoul, Korea. He got his M.S. and Ph.D. degrees in Computer Science from Illinois Institute of Technology (IIT), Chicago, Illinois, USA in 1987 and 1991 respectively. He has been a professor in the Department of Computer Engineering, Kyung Hee University, Korea since 1993. He is a founding director of the Ubiquitous Computing Laboratory, and has been affiliated with a director of Neo Medical ubiquitous-Life Care Information Technology Research Center, Kyung Hee

An Energy Efficient and Load Balancing Routing Algorithm for Wireless Sensor
Networks

University since 2006. Before joining Kyung Hee University, he was an assistant professor in the Department of Computer Science, Governors State University, Illinois, USA from 1992 to 1993. His current research focuses on Ubiquitous Computing and applications, Context-aware Middleware, Sensor Operating Systems, and Real-Time Systems etc.

Received: February 28, 2011; Accepted: April 25, 2011.

A Redistribution Method to Conserve Data in Isolated Energy-harvesting Sensor Networks

Peng Liu*, Song Zhang, Jian Qiu*, Xingfa Shen, and Jianhui Zhang

Institute of Computer Application Technology, Hangzhou Dianzi University
310018 Hangzhou, China

*{perryliu,qiujian}@hdu.edu.cn

Abstract. In ambient monitoring applications, the sensing field may be so far away from the data center that causes the direct relay routes between the sensor network and the data center impossible. Typically, in such isolated sensor network, data is stored in a distributed manner and collected by data mule. To improve the efficiency, sensed data is normally stored near the area where the mule will pass by with respect to storage limitation. However, previous researches didn't consider the energy constraint and energy harvesting capability of nodes. The purpose of this paper is to design a solution for fair data storage under space and energy limitation only based on local information. We propose a heuristic Distributed Energy-aware Data Conservation method (DEDC), which considers following two issues: i) where to store data with respect to energy and space storage, ii) how to prioritize the transmission of important data. Simulation has shown that the method is effective, energy efficient and robustness.

Keywords: isolated wireless sensor networks, energy-harvesting, data conservation.

1. Introduction

A Wireless Sensor Network (WSN) is normally composed of lots of sensor nodes which gather environmental data in a continuous or discrete manner. Nodes communicate with their neighbors and forward data to the sink through multi-hop routes. Many aspects, such as routes, channel access, locating, energy efficiency, coverage, network capacity, data aggregation and QoS have been explored extensively. Nowadays, problems studying of practical WSN applications has been a topic of great interest rather than theoretical research. For example, considering WSNs deployed in ambient environment such as habitat monitoring of Great Duck Island [13] and Landslide Prediction in India [16], how to ensure systems work sustainably are more important. These WSNs are more likely to be isolated from outside world as mentioned in [12] due to many reasons such as high cost of building base stations, sink node failure or adverse condition.

In this paper, the work is motivated by the project of GreenOrbs [11], which is targeted at using mobile wireless sensors for monitoring canopy density in

wild mountain areas which are difficult to access. Collecting data from such isolated WSNs may rely on vehicles passing by or a person carrying some special equipments. Furthermore, energy-saving or harvesting strategies can extend the lifetime of these WSNs. We consider data conservation issues in an isolated energy harvesting WSN, in which a data mule is used to collect stored data periodically. In the WSNs, each sensor node has limited RAM and flash storage space and faces the challenge that its battery tend to be flat. In such case, there are many issues have to be considered: i) where to store the received data to prevent data loss with respect to the limited energy shortage as well as space storage, ii) how to reduce communication overheads caused by data redistribution, iii) how to prioritize data by the importance and forward higher priority data as fast as possible. In this paper, a Distributed Energy-aware Data Conservation method (DEDC) is brought forward which allows nodes to decide where to send or to reserve data, and how to exchange with neighbors only based on the local information.

Another motivation of this paper is the project named Cyber IVY [9] which has been proposed and carried out for supplying building surveillance functions. Previously, it has been implemented with HDU Mote using normal rechargeable batteries. Recently, Cyber IVY has moved to a new stage. Nodes are powered by outdoor solar-power and wind power from external units of air conditioners. Data will be collected periodically by a staff(Mule) using handheld device. Therefore, it could be regarded as an isolated energy-harvesting sensor networks.

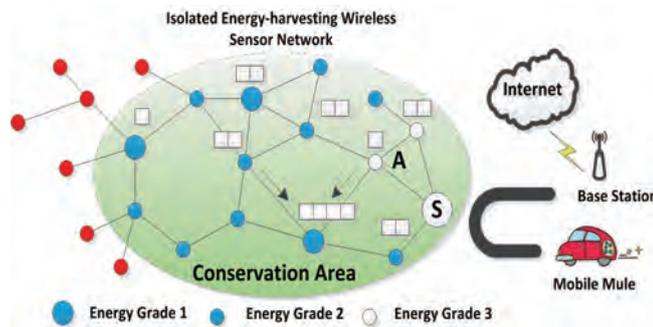


Fig. 1. Typical architecture of DEDC

In this paper, we try to find a method of data redistribution to maximize the network robustness. To make things simple we define conservation area near the sink node to store generated data packets. As shown in Fig.1, red nodes represent those outside the conservation area. The size of a node stand for its energy level. The whole network is isolated and can only be accessed by an external vehicle which will upload sensory data to the Internet via the base station. Inside the network, each node tries to forward packets to neighboring

nodes which are one hop closer to the sink and have more energy than itself. That is to say, data packets will be redistributed in the conservation area according to energy level and location. In our method, data packets will always be forwarded to nodes closer to the sink unless the node itself and its closer neighbors will deplete their energy very soon. For node A, it has to fall back its data because its energy grade is near critical level and it cannot find any node suitable to receive the data that are closer than node A.

The remainder of the paper is organized as follows. In Section 2, the related work is introduced. In section 3, we introduce the system model. Then the proposed DEDC method is described in detail in section 4, followed by the simulation and experimental evaluations in section 5. Finally, section 6 conclude the paper.

2. Related work

Different from researches on energy efficiency of battery-powered sensor networks, there are two main concerns in the energy-harvesting solution [7]. One is that rather than a limit on the maximum energy, it has a limit on the maximum rate at which the energy can be used. The other is that different nodes may have different harvesting opportunity as well as harvested energy availability. Therefore, besides minimizing the energy consumption and maximizing network operational time, maximizing the utility of the application subject to the harvested energy via routing protocols are evaluated in [6]. They compare the protocols under realistic scenarios and show how parameters of the MAC protocol can be optimized for a given harvesting scenario and network topology. Furthermore, Sharma et al. [15] consider generated energy together with generated data packets to achieve the largest possible data rate. They find conditions for energy neutral operation of the system, while keeping the data queue stable. They also obtain energy management policies which minimize the mean delay of the packets in the queue.

In this paper, we study issues of energy-harvesting sensor networks from the point of view of data conservation, especially in an isolated energy-harvesting sensor network. The main concern is that how and where to store sensory data regarding constrained storage space and useable energy. Data mule, isolated sensor network and energy-harvesting are three main attributes of this paper. Therefore we evaluate related works from three points of view as shown below.

2.1. Mobile sink

The use of mobile sink(s) is regarded as one of the most successful means of load balancing and efficient data collection. Finding an appropriate route that minimizes energy consumption for data dissemination from source to mobile sink is a major concern. Kim et al. [8] propose SEAD, a Scalable Energy-efficient Asynchronous Dissemination protocol which considers the distance and the

packet traffic rate among nodes, to minimize energy consumption in both building the dissemination tree and disseminating data to mobile sinks. In [20], mobile relays are employed not only to carry data packets but dynamically distribute network resources such as energy, computational power, sensing, and communication abilities. The trajectory of the mobile sink is another concern. Alnabelsi et al. have a mobile sink node patrol and collect the data from all the fragments across the network [1]. Dynamic Programming and Integer Linear Programming are introduced to find the optimal route of the mobile sink such that the energy consumption at the sensor nodes and inter-visit time within the fragment are minimized.

2.2. Data collection in isolated or sparse WSN

Isolated WSNs have some features similar to sparse WSNs. The difference is that for isolated WSNs the whole network is isolated from outside world while in sparse WSNs only single node are far from each other. The latter concerns mule discovery process, the data transfer process, data transfer rate and so on. In [14], an architecture to connect sparse sensor networks at the cost of higher latencies is proposed. The main idea is to utilize the motion of the existing entities to provide a low power transport medium for sensory data. Chakrabarti, et al. prove that using mules with predictable mobility [4] can significantly reduce communication power in WSNs. Ref [3] analyzed the optimal ARQ-based data-transfer protocol and provided an upper bound for any ARQ-based data-transfer protocol. In [2], an integrated evaluation of mule discovery and data transfer performance is provided. The results show that low duty cycle is actually feasible for most common environmental monitoring applications. Other means like using mobile ferries to conduct routing in a highly disconnected ad hoc network is discussed in [21]. *Message Ferrying (MF)* [21] is proposed for data delivery in sparse networks which utilizes a set of special mobile nodes to provide communication services for nodes in the network. The main idea is to introduce non-randomness in the movement of nodes. In [5], the authors developed a hybrid routing approach in which both MANET routing and message ferrying are used to explore available connectivity in clusters via gateway nodes. Yu-Chee Tseng et al. brings forward a *Distributed Storage Management Strategy (DSMS)* [19] to buffer data in an isolated WSN which concerns on space limit and data priority. By keeping higher-priority packets closer to the sink area, DSMS can reduce data loss probability and achieve higher efficiency. However, how to buffer packets in isolated energy harvesting WSNs to avoid data loss due to battery being flat remains unsolved.

2.3. Data collection in energy-harvesting sensor networks

There are many related work about energy-efficient data gathering or aggregation methods in wireless sensor networks. However, there are not much researches suitable for energy-harvesting sensor networks. In [10], a solution for

fair and high throughput data extraction from all nodes is designed in the presence of renewable energy sources. Specifically, the authors seek to compute the lexicographically maximum data collection rate and routing paths for each node such that no node will ever run out of energy. A centralized algorithm and two distributed algorithms are proposed. The centralized algorithm jointly computes the optimal data collection rate for all nodes along with the flows on each link, the first distributed algorithm computes the optimal rate when the routing structure is a given tree; and the second distributed algorithm, although heuristic, jointly computes a routing structure and a high lexicographic rate assignment that is nearly optimal. They prove the optimality for the centralized and the first distributed algorithm, and use real testbed experiments and extensive simulations to evaluate both of the distributed algorithms. In [17], data preservation problem is studied in the intermittently connected sensor networks under energy constraints at sensor nodes. By distributing the data items from low energy nodes to high energy nodes, data can be preserved for maximum possible time. However, the distribution process is uncoordinated as the data item could be put anywhere in the network regardless of the location of sink nodes, which will result in great transmission cost. The centralized algorithm is also hard to be implemented in large scale WSNs.

3. System model

3.1. Data redistribution problem

Large scale wireless sensor network for long-term environmental monitoring as GreenOrbs [11] has more than 700 nodes deployed in a reserved natural forest area for eight months. In such network, changing batteries or building a base station nearby which can directly access Internet is not feasible. So we consider an isolated energy-harvesting wireless sensor network in which data can only be collected by scheduled or non-scheduled data mules. The network is composed of a *sink node* and *static sensor nodes*. Regarding the route of mobile mules, single sink is more generality. Issues related with multi-sink will be addressed in future work. Static sensor nodes are all identical and have same initial energy and storage space. They can continuously monitor the environment and generate data packets periodically or according to events. In this paper, storage limit is not fully taken into account as in such a low duty cycle network, data packets are generated every 15 minutes. As shown in table. 1, a lot of space is still available beyond system and user needs. Each node has means of energy harvesting and the energy source will be flexible, such as wind or solar power. However, the node will be *dead* when its battery is flat. So how to avoid loss of sensory data is more important.

Before data mules collect sensory data, static nodes will store data packets in a distributed manner. Data mules will only visit the specified node *sink* for a period of time. Only during this fixed or random period can the sink forward the

¹ There is also an onBoardFlash free to use which has capacity of 1MB.

Table 1. Flash and Ram usage of HDU Mote (MSP430F1611) of some typical applications

HDU Mote	Flash(total 48KB) ¹	Used Per.	Ram(total 10KB)	Used Per.
Blink	2732B	5.7%	55B	0.5%
HotelMon	19072B	39.7%	1522B	1.5%
MultihopOscilloscope	31228B	65%	3906B	39%
TestNetwork	32374B	67.4%	3218B	32%

data packets to the mule. After that, the mule will move to a base station and upload all the data packets.

3.2. Network model and problem formulation

The principle idea of the paper is to store data safely and reduce transfer latency when data mules arrive. We propose a distributed energy-aware data reservation method to coordinate node action to avoid control packet overheads. To reduce the cost, as mentioned in [19], we set up a *Conservation Area* which is a predefined region adjacent to the sink. Nodes in CA are responsible for storing sensory data of the whole network. The scale of CA is defined by the nodes within the maximum hop-count to the sink so that it can scale from 3 hops to 10 hops. Our proposed method will focus on data dissemination scheduling among nodes in CA. By defining the CA part of the network instead of the whole network we can significantly reduce overhead caused by centralized method. Nodes not in CA will forward their data packets to those in CA. With respect to the single sink strategy the action of other nodes (direction of forwarding data) is accordance with non-energy-constraint system. Therefore, the communication cost can be ignored. Three facts should be considered such as i) Data loss should be minimized all over the network, ii) Packets with higher priority should be delivered earlier, iii) Transfer latency should be minimized in CA. For simplicity, we defined some terms as follows.

Definition 1. Energy threshold T We define energy threshold to justify node energy status. Typically, $T1$ equals 65% of total energy. $T2$ equals 30% of total energy which represents the lower bound.

Definition 2. Energy harvesting amount $A(u)$ Let $A(u)$ be the total amount of energy that has been collected by node u for a certain period of time Δt . $H(u)$ denotes the energy harvesting capability of node u which will be a function of input currency. We use historical knowledge to predict current and future energy harvesting rate. So that $A(u)$ could be computed as $\int_0^{\Delta t} H(u)dt$. To make things simple, we classify every 5% percent energy as one step S . Δt varies from system to system. In the Cyber IVY project, we set Δt as 5 minutes. The energy harvesting speed can be seen in Fig. 2 (note that functional voltage is from 1.4 to 2.6).

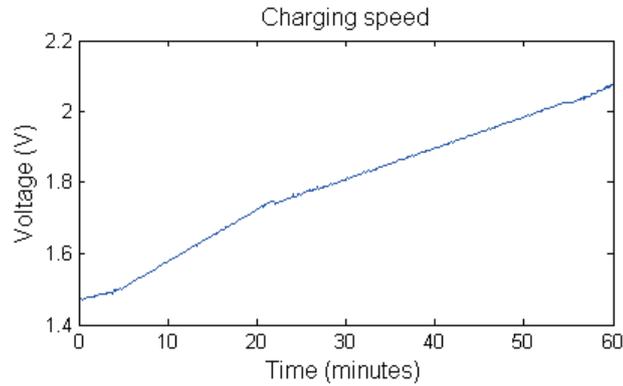


Fig. 2. Energy harvesting speed of node in Cyber IVY

Definition 3. Energy grade $E(u)$ Let $B(u)$ indicate the residual energy of node u and E_{max} indicate the maximum energy capacity of a node. We assume all nodes have the same energy draining rate c so that the energy grade of a node is $E(u) = B(u) + A(u) - c\Delta t$. To normalize the equation, we use:

$$E(u) = (B(u) + A(u) - c\Delta t) / (E_{max} \times 0.05) \quad (1)$$

It is decided by three facts: residual energy, energy consumption and energy harvesting rate, which will seriously affect the safety of the system. To simplify the node action, We model node energy state into three main grades (levels) which will be sperate by T . $G1 : T1 \leq E(u)$ means the node have ample power that is greater than threshold $T1$. $G2 : T2 \leq E(u) \leq T1$ means energy residua locates between $T1$ and $T2$ and maintains a balance. $G3 : E(u) \leq T2$ means energy locates lower than $T2$.

Definition 4. Sink distance $D(u)$ Each node or packet is some hops away from sink. $D(u)$ means the minimum hop count of node u to the sink, while $D(i)$ stands for the minimum hop count of packet i to the sink.

Definition 5. Survive probability of packet S We define survive probability of packet i located on node u as $S_{i \rightarrow u}(i) = \frac{E(u)}{\lambda(u)D(u)}$ in which $\lambda(u)$ is a function of three factors, namely, communication cost, communication latency and hop count from u to the sink. Normally, we let $\lambda(u) \equiv 1$. The greater the S is the higher chance the data delivered to mobile mules.

Definition 6. System robustness R We define the whole system robustness as

$$R(CA) = \sum_{u \in CA, i \rightarrow u} S_{i \rightarrow u}(i)P(i) \quad (2)$$

in which $P(i)$ denotes the priority of packet i . Thus, the sum of all survive probability of all packet times by their priority is maximized.

$P(i)$ can be defined using many methods. Therefore, the main goal of our proposed method is to maxim Global System Robustness function R in CA.

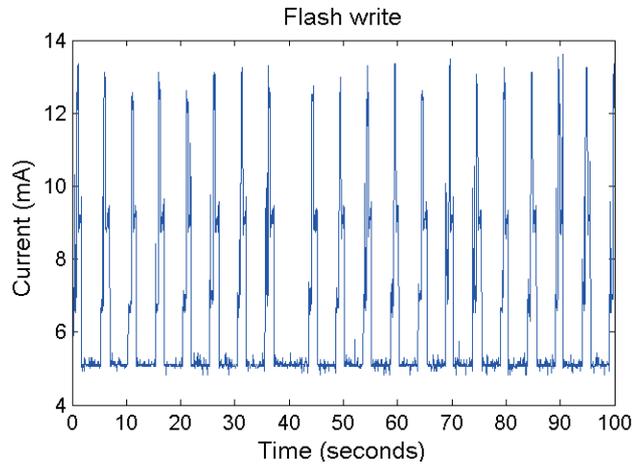


Fig. 3. The energy consumption factor of writing action of onChipFlash

There's no detailed information of currency consumption of OnChipFlash so that we did some experiments and got the following result. Fig. 3 indicates that writing operation takes place on onChipflash for 2.5 seconds every 5 seconds. This is the only routine running on HDU Mote. No radio, no sensing, so the lower current bound is around 5mA. Average flash writing currency is approximately 10mA which is not much compared with external flash. Therefore, node can move data to the flash when necessary while energy cost is reasonable. The final result is shown in table 2.

Table 2. Currency consumption factor

Action	HDU Mote OnChipFlash	OnBoardFlash
Active current(READ)	5mA	4mA
Active current(WRITE/ERASE)	10mA	15mA

4. Distributed energy-aware data conservation method (DEDC)

In isolated sensor networks, such as GreenOrbs, data will be sampled every 15 minutes. The full size of each sample data is 12 bytes. It can support more than 150 hours continuous work only relying on ram of nodes. Storing data in

onChipFlash or onBoardFlash are both feasible, which are 1.25 Mbytes in total. Therefore, it is believed that energy is more important than storage space. Communication is the major part of energy consumption. However, because CA is close to the sink, each node out of CA will forward their packets to CA which has the same goal compared with other energy efficient routing protocols. In CA, each node will try to forward data packets to the nodes that are closer to the sink. That is to say, we can use any existing routing protocol to improve the efficiency so that we don't take communication cost into account. In this paper we propose a Distributed Energy-aware Data Conservation method to maximum the maxim Global System Robustness function R in CA. The assumptions are summarized as follows:

- Each node u knows its distance $D(u)$ to the sink and the neighbor set $N(u)$ within one hop; These parameters are easy to obtain when the network was setup and update by broadcasting. We try to make the method distributed so we build our method on local knowledge. One hop is a balance of accuracy and overhead. Centralized method or distributed method needing whole network knowledge is not feasible.
- Each node has sufficient space storage for the application.

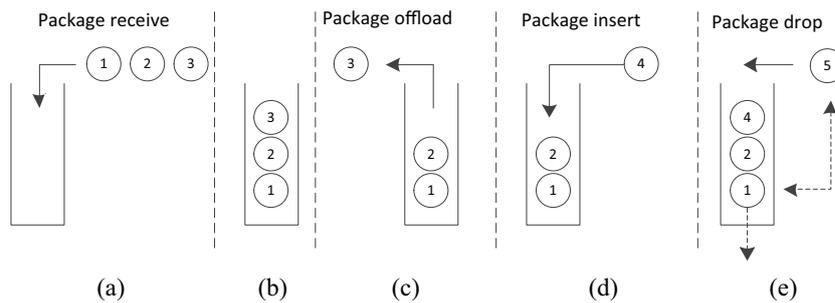


Fig. 4. Organization of storage structure

In practice, data packets always have different priorities. Packets of higher priority deserve greater chance to survive in terms of storage and energy limitation. Normally, besides preassigned priorities, the newer the packet is the higher the priority is. Therefore, DEDC method uses a stack-like data structure to store packet in their memory to ensure newer packet will always be sent earlier. As shown in Fig.4(a), when the buffer is empty three data packets arrive respectively and are stored as shown in Fig.4(b). Fig.4(c) shows that packet '3' will be offloaded first when data is passing on. Newer packet will be inserted from the top of the stack as seen in Fig.4(d). If dropping data is inevitable the newer packet will be compared with the packet at the bottom of the stack. The one with lower priority will be discarded. The structure is easy to implement and

has less number of comparison which results in a better time complexity and lower overhead.

For data redistribution, nodes of three different energy grades follow different actions and algorithms, but they all obey a fundamental rule:

R1: Denote $E(i)$ to be the energy grade of the last relay node where packet i came from. For each node u , a packet i stops and locates on u only when $\forall v \in N(u), E(v) < \min\{E(u), E(i)\}$ where $D(v) < D(u)$.

Each node will attempt to deliver packets to nodes closer to the sink. R1 describes when this approach will stop under certain conditions. It implies that packets are forwarded to nodes which have more energy and closer to the direction of data center. In the case that there are some *strong* nodes in the CA, if we only consider a node's one hop neighbors, packets will be stuck in these nodes because no nodes around them have more energy. Meanwhile, if there are nodes of less energy standing in main data route they will also become bottle neck and block packet transmission. We call both situations the fat energy wall phenomenon and the thin energy wall phenomenon respectively. As shown in Fig.5, there are three nodes in a line and the size of circle represents the energy grade. Fig.5(a) shows two packets i, j are not able to pass through node v because any neighbor nodes of node v have less energy. If node v has least energy among three node two packets i, j can not pass through node v ether as shown in Fig.5(b). To avoid the existence of energy walls that block packets from going to the sink and store too many packets in its memory, we consider two hop situation in a distributed manner to balance packets load in CA while each node still needs one hop information of its neighbors and packets. At the same time, each node has two actions, namely *PUSH* and *PULL*, against the fat energy wall and the thin energy wall respectively. However, *PULL* is a conditional action, only when node raises a request will it be performed. To support two actions, node u not only considers next hop neighbors but also previous hop ones.

To formulate our method we bring forward some basic rules as indicated below:

C1: For each node u , let $v = \max EDN(u)$ be the node with highest energy grade of all neighbors of u where $D(v) < D(u)$, node u moves packet to v when $E(v) \geq E(u)$.

C2: For node $v = \max EDN(u)$, if C1 is not met, node u moves packet i to v if $E(v) \geq E(i)$.

C3: For node $u, v \in N(u)$ and $D(v) > D(u)$. On receiving a packet transferring request from v , if \exists node $w = \max EDN(u)$ and $E(w) > E(v)$, Node u *PULLs* the packet from v and *PUSHs* it to w .

C4: If C1, C2, C3 could not be satisfied, node u tries to move packet to $v, v \in N(u)$ if and only if $E(v) > E(u)$ and $E(u) \in G3$ where $D(v) > D(u)$.

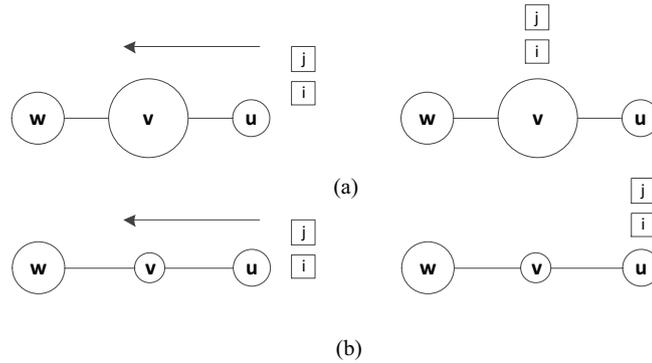


Fig. 5. Two examples of energy wall phenomenon

C5: If C1, C2, C3, C4 could not be satisfied, node u will store the packet in its flash memory.

Whenever a node receives or generates a packet, it will determine its action according to the above conditions. A node will examine these five conditions in a sequence of C1 to C5. Therefore, packets are forwarded to nodes as close to the sink as possible. Nodes use RTS/CTS like protocol to negotiate the process of packet exchanging. Communication cost is reduced as only one hop neighbors are involved.

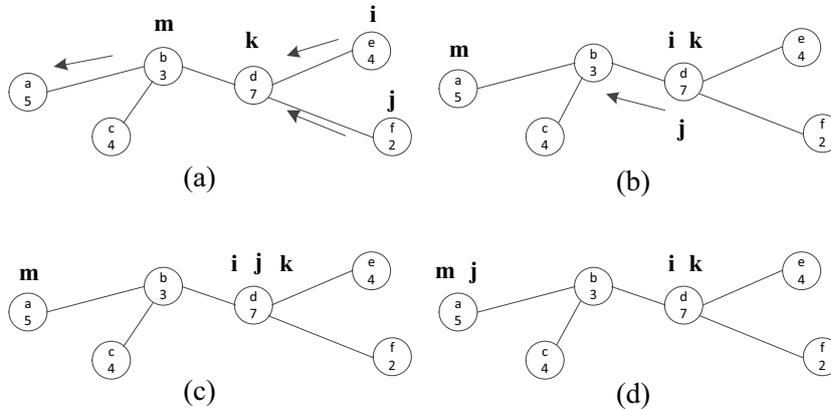


Fig. 6. An example of DEDC method

As shown in the example of Fig. 6(a), there are six nodes in part of the network. Each of which has a name and energy grade in the circle. Four packets i, j, k, m are generated from node b, d, e, f respectively. Regarding condition

C1, node b will forward packet m to node a and node e will forward packet i to node d . Node f will forward packet j to node d and node d chooses to keep packet k . Greedy method that only perform condition C1 will finally stop at the snap as shown in Fig.6(c). Thus, node d becomes a wall which blocks packets going to node a, b, c . Regarding DEDC, node d will try to forward packet j to node b because the energy grade of its last relay node (source node) is less than that of node b as indicated in Fig.6(b). Finally, packet j will be moved to node a and stop there as can be seen in Fig.6(d). Therefore, DEDC has better balance on packet delivery and Survive Probability S in CA.

The Distributed Energy-aware Data Conservation method for each node when receiving or generating a packet is described as following:

Algorithm 1 DEDC method

```
1: when a packet  $i$  is received or generated by node  $u$ 
2: FOR each node  $v \in N(u)$ 
3:   IF C1 is satisfied
4:     move packet  $i$  to node  $v$ ;
5:   ELSE IF C2 is satisfied
6:     move packet  $i$  to node  $v$ ;
7:   ELSE IF C3 is satisfied
8:     move packet  $i$  to node  $w$ ;
9:   ELSE
10:    CASE energy grade of  $E(u)$ 
11:    G1: broadcast its hop count and energy information;
12:    G2: null operation
13:    G3: FOR each node  $w \in N(u)$ 
14:      IF C4 is met
15:        data fall back to node  $w$ 
16:      ELSE
17:        C5: store data into flash memory.
18:      ENDIF
19:    ENDCASE
20:  ENDIF
```

5. Performance evaluation

To verify the performance of proposed DEDC method we conducted some simulations. First, we test data storage structure of DEDC and compare it with OPT and FIFO algorithms. The simulation environment contains 100 sensor nodes each of which has storage space of 20 data packets (ram together with flash). Each node will generate a data packet every 10 seconds and each packet has a random priority between 0 and 3. Each time when the data mule comes it will fetch all data and the time of this process is ignored. Two algorithms are chosen as competitors. When the storage space of a node is full, FIFO drops

data in a First-in-First-Out manner. OPT algorithm will search out the entire storage space to find a packet with lowest priority for substitution. Fig.7 compares the average priorities of the packets collected by data mule when we vary the visiting interval of the mule and CA size. Fig.7(a) shows that as the visiting interval increases, the average priority also increases. It means OPT can collect more high priority packets while the increasing of visiting interval results loss of more and more data packets. Fig.7(b) shows that the average priority decreases as the CA size is enlarged which means increasing of successful transmitting packets. DEDC has similar performance compared with Optimal algorithm in both experiments while it needs less computation power and comparison.

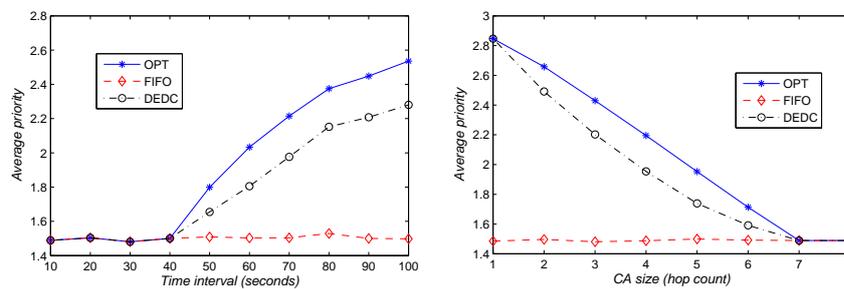


Fig. 7. The comparison of average priorities of packets collected by mules by varying (a) the visiting interval and (b) the different size of CA.

The metrics of evaluating algorithms in related work [12][18] is total redistribution cost which implies energy cost for data transfer. Data packets are stored distributively in the whole network without consider of cost of future collection. In our work, data mule will only visit the sink node of the network for a short period of time so that data should be offloaded to buffer area near the sink node through energy-efficient route. Our goal is to ensure data is distributed near the sink node regarding priority and storage safety. As we define $\lambda(u) \equiv 1$ so that Survive Probability of packet is seriously affected by network size. Two methods are chosen as competitors, Greedy method only forward packets to nodes which have more energy and closer to the sink, Non-schedule method will keep packets where they are generated. As shown in Fig.8, all three methods turn to achieve less System Robustness when the number of nodes increases. This is because that packets are more likely to be stored far away from the sink. However, DEDC outperforms Greedy method and Non-schedule method because it has better balance between data safety and data transmission overhead. When network scale increase Greedy method tends to encounter energy wall phenomenon and Non-schedule method will lead high data transmission latency.

We randomly generate 500 packets and let the data mule collect data only for a short period of time and evaluate the successfully delivered packets while the network size increases. As shown in Fig.9, transmitted packets decrease

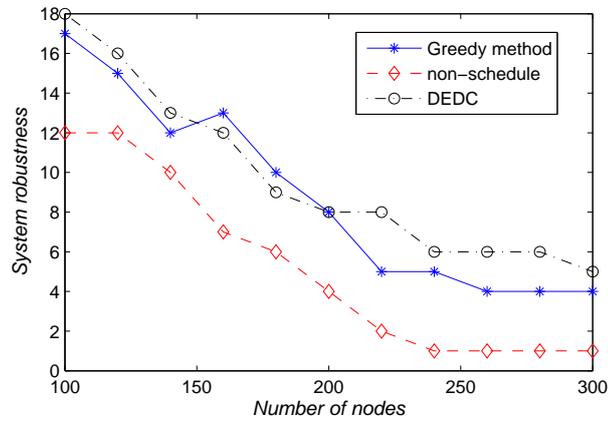


Fig. 8. System robustness of different network size

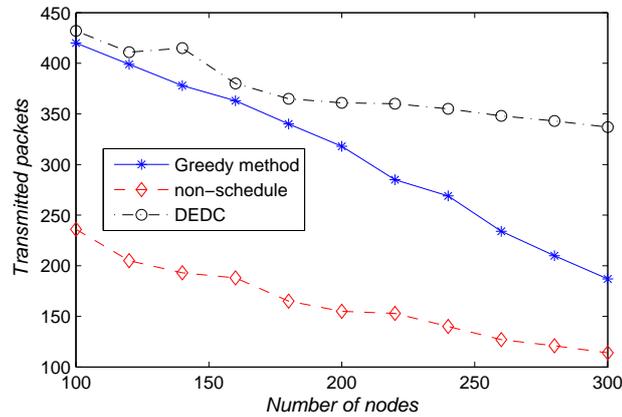


Fig. 9. Transmitted packets of different network size

because more and more packets are stored far from the sink when the number of nodes increase. However, DEDC has a lower drop rate so that it achieves best performance because more packets are closer to the sink.

6. Conclusions

In this paper, a Distributed Energy-aware Data Conservation method (DEDC) has been introduced which helps to extend the lifetime and achieve reasonable packet delivery performance in isolated wireless sensor networks. This scheme allows nodes to decide where to send, to reserve on itself or exchange with neighbors based only on local information. The method is eval-

uated through theoretical analysis and simulation. It has been proved to be efficient, which has achieved the following advantages: i) an improved data packet storage scheme has been provided with respect to limited energy, ii) the communication overheads caused by data redistribution has been reduced, iii) high priority data can be delivered earlier which helps in network balancing.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (Grant No. 60903167), the Pre-research Scheme of National Basic Research Program of China (973 Program) under Grant No. 2010CB334707, and the Doctoral Scientific Research Foundation of Hangzhou Dianzi University (Grant No. KYS055608107).

References

1. Alnabelsi, S.H., Almasaeid, H.M., Kamal, A.E.: Optimized sink mobility for energy and delay efficient data collection in FWSNs. In: ISCC. pp. 550–555. IEEE (2010)
2. Anastasi, G., Conti, M., Francesco, M.D.: Data collection in sensor networks with data mules: An integrated simulation analysis. In: ISCC. pp. 1096–1102. IEEE (2008)
3. Anastasi, G., Conti, M., Monaldi, E., Passarella, A.: An adaptive data-transfer protocol for sensor networks with data mules. In: WOWMOM. pp. 1–8. IEEE (2007)
4. Chakrabarti, A., Sabharwal, A., Aazhang, B.: Using predictable observer mobility for power efficient design of sensor network. In: IPSN. pp. 363–385 (2003)
5. Chen, Y., Zhao, W., Ammar, M., Zegura, E.: Hybrid routing in clustered dtns with message ferrying. In: MobiOpp. pp. 75–82. ACM (2007)
6. Hasenfratz, D., Meier, A., Moser, C., Chen, J.J., Thiele, L.: Analysis, comparison, and optimization of routing protocols for energy harvesting wireless sensor networks. In: SUTC/UMC. pp. 19–26. IEEE Computer Society (2010)
7. Kansal, A., Hsu, J., Zahedi, S., Srivastava, M.B.: Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems* 6(4), 32:1–32:38 (Sep 2007)
8. Kim, H.S., Abdelzaher, T.F., Kwon, W.H.: Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. In: SenSys. pp. 193–204. ACM Press (2003)
9. Liu, P., Zhang, S., Qiu, J., Dai, G.: Building surface mounted wireless sensor network for air conditioner energy auditing. In: Proceedings of 2011 INFOCOM workshop: The First International Workshop on Cyber-Physical Networking Systems. pp. 749–754 (2011)
10. Liu, R.S., Fan, K.W., Zheng, Z., Sinha, P.: Perpetual and fair data collection for environmental energy harvesting sensor networks. to appear in *IEEE/ACM Transaction on Networking* pp. 1–14 (2011)
11. Liu, Y., Zhou, G., Zhao, J., Dai, G., Li, X.Y., Gu, M., Ma, H., Mo, L., He, Y., Wang, J.: Long-term large-scale sensing in the forest: recent advances and future directions of greenorbs. *Frontiers of Computer Science in China* 4(3), 334–338 (2010)
12. Luo, L., Huang, C., Abdelzaher, T.F., Stankovic, J.: Envirostore: A cooperative storage system for disconnected operation in sensor networks. In: INFOCOM. pp. 1802–1810. IEEE (2007)

Peng Liu, Song Zhang, Jian Qiu, Xingfa Shen, and Jianhui Zhang

13. Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., Anderson, J.: Wireless sensor networks for habitat monitoring. In: Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA-02). pp. 88–97. ACM Press (2002)
14. Shah, R.C., Roy, S., Jain, S., Brunette, W.: Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks* 1(2-3), 215–233 (2003)
15. Sharma, V., Mukherji, U., Joseph, V., Gupta, S.: Optimal energy management policies for energy harvesting sensor nodes. *IEEE Transactions on Wireless Communications* 9(4), 1326–1336 (2010)
16. Sheth, A., Thekkath, C.A., Mehta, P.: Senslidea distributed landslide prediction system. *Operating Systems Review* 41(2), 75–87 (2007)
17. Takahashi, M., Tang, B., Jaggi, N.: Energy-efficient data preservation in intermittently connected sensor networks. In: Proceedings of 2011 INFOCOM workshop: The 3rd International Workshop on Wireless Sensor, Actuator and Robot Networks. pp. 590–595 (2011)
18. Tang, B., Jaggi, N., Wu, H., Kurkal, R.: Energy-efficient data redistribution in sensor networks. In: MASS. pp. 352–361. IEEE (2010)
19. Tseng, Y.C., Lai, W.T., Huang, C.F., Wu, F.J.: Using mobile mules for collecting data from an isolated wireless sensor network. In: Proc. 2010 International Conference on Parallel Processing (39th ICPP'10). pp. 673–679 (Sep 2010)
20. Wang, W., Srinivasan, V., Chua, K.C.: Extending the lifetime of wireless sensor networks through mobile relays. *IEEE/ACM Trans. Netw* 16(5), 1108–1120 (2008)
21. Zhao, W., Ammar, M.H., Zegura, E.W.: A message ferrying approach for data delivery in sparse mobile ad hoc networks. In: *MobiHoc*. pp. 187–198. ACM (2004)

Peng Liu received his B.S. and M.S. in Computer Science and Technology from Hangzhou Dianzi University in 2001 and 2004, and D.Eng in Computer Science and Technology from Zhejiang University in 2007, China. Currently he is an associate Professor at Hangzhou Dianzi University. His research interests include embedded system, wireless sensor networks and distributed computing, etc.

Song Zhang received his B.E. degree in Software Engineering from Hangzhou Dianzi University, China in 2010. He is currently a graduate student at Hangzhou Dianzi University. His research interests include wireless sensor networks, embedded systems, etc.

Jian Qiu received his B.E. degree in Information Engineering from Zhejiang University in 2005. He received his MSc in Communication Engineering and PhD in Electrical Engineering in the department of Electronics from University of York in 2006 and 2010 respectively. His current research interests include sensor networks, wireless communications, mobile networks, etc.

Xingfa Shen received his B.E. degree in Electrical Engineering and Ph.D. in Control Science and Engineering from Zhejiang University, China in 2000 and 2007. He is currently an associate Professor at Hangzhou Dianzi University. His research interests include wireless sensor networks, embedded systems, etc.

Jianhui Zhang received the PhD degree and the BS degree from Zhejiang University and Northwestern Polytechnical University, China in 2008 and 2000, respectively. He is an assistant professor of computer science at the Hangzhou Dianzi University now. His research interests include algorithm design and analysis, distributive systems, sensor networks and wireless networks. He is a member of IEEE.

Received: April 20, 2011; Accepted: May 4, 2011.

A Packet Buffer Evaluation Method Exploiting Queueing Theory for Wireless Sensor Networks

Tie Qiu^{1,2}, Lin Feng², Feng Xia^{1*}, Guowei Wu¹, and Yu Zhou¹

¹ School of Software, Dalian University of Technology,
116620 Dalian, China
qjutie@dlut.edu.cn; f.xia@ieee.org

² School of Innovation Experiment, Dalian University of Technology,
116024 Dalian, China
fenglin@dlut.edu.cn

Abstract. In large-scale wireless sensor networks (WSNs), when the consumption of hardware is limited, how to maximize the performance has become the research focus for improving transmission quality of service (QoS) of WSNs in recent years. This paper presents a new evaluation method for packet buffer capacity of nodes using queueing network model, whose packet buffer capacity is analyzed for each type node, when it is in the best working condition. In order to evaluate congestion situation in the queueing network, and to get real effective arrival rates and transmission rates in the model, holding nodes were added in the queueing network model, and equivalent queueing network model is expanded. We establish an M/M/1/N type queueing network model with holding nodes for WSNs and design approximate iterative algorithms. Experimental results show that the model is consistent with the real data.

Keywords: wireless sensor networks, queueing network model, blocking, packet buffer capacity, node utilization.

1. Introduction

Wireless sensor networks (WSNs) are successfully applied in intelligent transportation, monitoring environment, location and other fields. They consist of tiny sensing devices that have limited possessing and computation capabilities, and can collaborate real-time monitoring, sensing, collecting network distribution of the various environments within the region or monitoring object information [1,2,3]. WSNs of distribution regions are composed of sink nodes [4,5,6], transmission nodes and boundary nodes [7]. The performance of each type node will affect the overall network performance in WSNs. Throughput and utilization [8,9,10] of the nodes in the lifetime [11,12] are the main evaluation performance indicators of WSNs. The

* Corresponding author

packet buffer capacity of nodes is an important factor in utilization of network nodes [13]. If a node of WSNs is blocked, and packet buffer set too small, the entire network data transmission and processing efficiency is not high. Therefore, when the consumption of hardware is limited, how to optimize the node packet buffer size and maximize the performance for WSNs has become a research focus for improving the Quality of Service (QoS) of WSNs transmission in recent years.

In this paper, we consider that packet buffer capacity corresponds to the length of the waiting queue in the established limited capacity of the queueing network model. When the length of the waiting queue reaches the maximum, the node is blocked in the queueing network model. Therefore, a typical WSN is modeled by M/M/1/N type queueing network model. The method of modeling based on topology of nodes in WSNs and performance analysis of the packets buffer capacity have been proposed. According to the topology of WSNs and operational characteristics, arrival, transferring and leaving relationships of transmission nodes, boundary nodes and sink node are analyzed, and data flow balance equations are obtained. In order to evaluate the congestion situation in the queueing network, and get real effective arrival rates and transmission rates in the model, holding nodes were added in the queueing network model and equivalent queueing network model is expanded. By analyzing the queueing model with blocking probability, to obtain the performance index of system when it is in steady state, approximate iterative algorithms are designed. The performance parameters of nodes model in the WSNs are calculated using limited iteration times. The optimal values for packets buffer sizes settings are obtained for transmission nodes, boundary nodes and sink nodes.

The remainder of this paper is organized as follows. The related work and problem statement are introduced in Section 2 and Section 3. Section 4 describes the modeling of WSN and analysis. Section 4.1 describes the modeling method of using open queueing network model for WSNs; the balance equations of data flow are established. In Section 4.2, the equivalent queueing network model is obtained, that holding nodes are added in the may be blocking nodes. The model parameters of blocking probability of data packets, the arrival rate and node transfer rate, all are analyzed in Section 4.3 using the equivalent queueing model. Section 5 designs iterative approximation algorithms for total arrival rate of nodes and effective arrival rate and transfer rate of nodes with blocking probability. Section 6 gives the numerical calculation and analysis of experimental results. The performance parameters of WSN nodes are calculated using iterative algorithms given in Section 5. According to the relationship curves between utilization and buffer size, the packets size of the optimal buffer settings are obtained for transmission nodes, boundary nodes and sink node, respectively. The correctness of modeling and analysis method is verified by experimental data of the WSNs. Section 7 contains the conclusion and future work.

2. Related Work

When the hardware has been implemented, it is difficult to adjust the node's hardware resources in accordance with specific needs. Therefore, researchers have proposed the need for large-scale WSN nodes modeling method [14]. Through performance evaluation of pre-setting nodes, optimal parameters of allocation for the hardware nodes are obtained. The current modeling method based on Petri nets [15,16,17] is suitable for macro-modeling, but it is not a specific modeling technique for large-scale WSNs. Queueing network is an effective system-level modeling method, which is widely used in the modeling and performance analysis of computing and communication systems [18,19]. It has many advantages that include a highly abstract and rich theory for modeling.

In recent years, researches have made some progress on analyzing and improving network performance in the application of finite capacity queueing networks. Bisnik et al. [20] modeled random access multi-hops wireless networks as open G/G/1 queueing networks and used diffusion approximation in order to evaluate closed form expressions for the average end-to-end delay. In [21,22], Kouvatsos and Awan described the priorities and blocking mechanisms with open-loop queueing network performance analysis, and queueing network parameters on the approximation and error estimates. Özdemira et al. [23] presented two Markov chain queueing models with M/G/1/K queues, which have been developed to obtain closed-form solutions for packets delay and packets throughput distributions in a real-time wireless communication environment using IEEE 802.11 DCF. Mann et al. [24] developed a queueing model for analyzing resource replication strategies in WSNs, which can be used to minimize either the total transmission rate of the network or to ensure that the proportion of query failures does not exceed a predetermined threshold. In [20], Liehr et al. introduced enhancements to the standard of extended queueing network models, which allow the modeling and the simulation of inter-process communication and highlight the benefits granted by their enhanced EQN approach. However, these researches don't address the packet buffer capacity of nodes and how to set the buffer size to derive the optimal performance of the nodes in WSNs.

3. Problem Statement

Data packets are transmitted and processed in collaboration by the sink nodes, transmission nodes and boundary nodes. For a large scale WSN, a queueing network model can be used to analyze its performance [25, 26]. But how to configure resources to find the best value hardware using trends of changing the parameters of performance is an important reference for node design. The definition of the threshold of node buffer capacity is given below:

Definition 1. When the queueing network system is stable, node's hardware buffer capacity just accommodate the maximum length of the queue

to be processed. Buffer size value at the moment is called the node threshold, denoted by N_T .

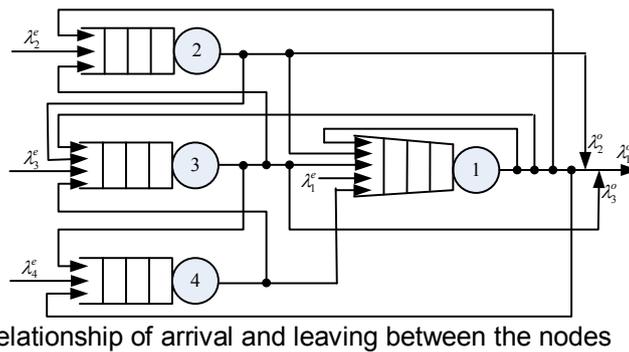
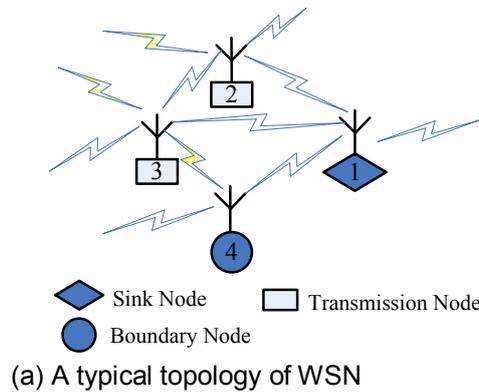


Fig. 1. Topology and node transfer of the WSN.

In WSNs, for any node buffer size, we make the following discussion:

(i) If $N_i \geq N_T$, node queue length will never be processed over the buffer capacity when a system is in a steady state. Therefore, the packets that have not been timely processing data will be placed in the packet buffer. Newly arrived packets will not cause the blocking node server.

(ii) If $N_i < N_T$, when the packet buffer of node is full, the link paths that include the nodes are blocked, and lead to the processing efficiency of the whole WSN down. On the other hand, when the link path is blocked, all the nodes are in an active state in the link path. Therefore, energy consumption of the node is larger, and individual nodes are invalidated due to energy exhaustion.

Figure 1(a) shows a typical topology of WSN, which is composed of sink node 1, transmission nodes 2 and 3, and boundary node 4. According to the modeling method of open queueing network, it describes the transmission

relationship between the node queues as shown in Figure 1(b), where λ_i^e is the independent external Poisson arrival rate of node i and λ_i^o is the leaving rate after the completed service of node i . (Note: Boundary node does not include λ_i^o).

In practice, the task content of sink nodes, transmission nodes and boundary nodes is different. Therefore, its consumption of hardware resources will be different. For example, in Figure 1(b), the arrival rate of sink node 1 packets is the maximum. Therefore, finding a way to properly evaluate its performance becomes very important. That is, how to set the packet buffer capacity of wireless sensor nodes, in order that each node has the highest speed of data processing and throughput. Thus, the best parameters between the utilization of node and consumption of hardware buffer capacity will be found. In order to facilitate the analysis of queueing network model for WSNs, some symbols are defined in Table 1.

Table 1. Definition of symbols

Symbol	Description
i, j, k, d	Node No. in the queueing network.
M	Total number of paths in the queueing network.
N	Size of packet buffer capacity
λ_i	Arrival rate of node i .
μ_i	Service rate of node i .
ρ_i	Utilization of node i .
p_i	Steady-state probability of arrival node i .
p_i^o	Leaving probability from node i .
p_{ij}	Probability of from node i to node j .
S	State of node.
T_i	Monitoring cycle of i -th times in WSN.
A	Aggregate of all the holding nodes.
pb_{ij}	Blocking probability from node i to node j .
pb_i^e	Independent external Poisson arrival blocking probability of node i .
pb_i^a	Total arrival blocking probability of node i .

4. Open Queueing Network Analysis for WSN Model

In this section, we describe the modeling method of using open queueing network model for WSNs and the balance equations of data flow are established. In order to analyze blocking of the queueing network model, holding nodes were added in the may be blocking nodes, and the equivalent

queueing network model is obtained. Therefore, analysis of the blocking queueing network is possible.

4.1. Open Queueing Network

A typical open queueing network composed of WSN is consistent with a flow balance equation. According to the theorem for flow balance equation [25,27], we can give Corollary 1 and Corollary 2 for any WSN.

Corollary 1. For the transmission node, arrival node and leaving node in the queueing network of WSN, the number of all possible packets leaving the node i equals that of the arrival number in the state.

$$\lambda_i^e + \sum_{k=1}^{m-1} \lambda_k p_{ki} = \sum_{f=j}^d \lambda_i p_{if} + \lambda_i p_i^o \quad (1)$$

Proof: Using reductio ad absurdum, suppose the number of packets entering the node i is not equal to the number of packets leaving the node i , then according to the flow balance equation, there must be packets with probability $p_{ii} \neq 0$ in the self-loop. This is in contradiction with the fact that transfer node, arrival node and leaving node service is the order of one-way services in WSNs. Therefore, the Corollary 1 is established. \square

Corollary 2. For the sink node in the queueing network of a WSN, the sum of the arrival packet and self-loop packet equals that of the sum of the leaving packet for node i in the state.

$$\lambda_i^e + \sum_{k=1}^{m-1} \lambda_k p_{ki} + \lambda_i p_{ii} = \sum_{f=j}^d \lambda_i p_{if} + \lambda_i p_i^o \quad (2)$$

Proof: For the sink node in the queueing network of a WSN, data transmission between nodes requires a very high accuracy. When the data packets validation is not correct, we should have re-transmission processed until the correct calibration data is received. Therefore, the phenomenon of self-loop appears in the node i . Increased the number of arriving packet is equivalent to $\lambda_i p_{ii}$. Therefore, we can get the sum of arrival packet as equation (3).

$$\lambda_i = \lambda_i^e + \sum_{k=1}^{m-1} \lambda_k p_{ki} + \lambda_i p_{ii} \quad (3)$$

According to flow balance equation [25,27], we can deduce that the flow balance equation (2) is established. \square

The Nodes are used in environmental monitoring and control in WSNs, due to factors such as limited power consumption, which are not always in running state [28,29,30]. The states of nodes in WSNs are alternating between

monitoring and sleeping, as a premise in meeting the required monitoring conditions. Switching between states of nodes is shown in Figure 2.

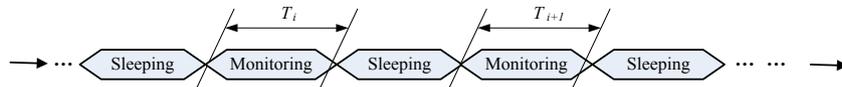


Fig. 2. Switching between states of nodes.

Next, we give the definition of the longest monitoring cycle of node i .

Definition 2. For all nodes in WSNs, the wake-up from a sleeping state into the monitoring state, and then entry into another sleep experienced by far the longest time is known as the longest monitoring cycle of WSN nodes.

$$T_i^m = \max\{T_1, T_2, \dots, T_i, \dots, T_N\} \quad (4)$$

In WSNs, each node has a packet capacity. If the packet buffer size is set too large, would be a waste of resources. Conversely, if the data packet buffer size is set too little, block of system will be increased. We give the theorems for packet queue length of sink nodes and transmission nodes.

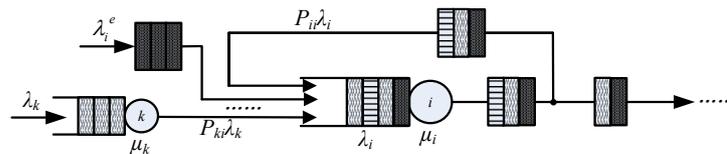


Fig. 3. Transmission relation of packet queue for sink node i .

Theorem 1. For any sink node i in WSNs, packet queue length of sink node is L_i^m , then the following relationship is obtained.

$$\lambda_i^e + \sum_{i=1}^{m-1} \lambda_k p_{ki} = \frac{L_i^m}{T_i^m} + (1 - p_{ii})\mu_i \quad (5)$$

Transmission relation of packet queue for sink node i in WSNs is shown in Figure 3. We give the proof of Theorem 1 below.

Proof: For sink node i in WSNs, according to Corollary 2, the data packets arrival rate is obtained by equation (6).

$$\lambda_i = \lambda_i^e + \sum_{i=1}^{m-1} \lambda_k p_{ki} + \lambda_i p_{ii} \quad (6)$$

In the time period, the arrival number is X_i given by equation (7).

$$X_i = T_i^m \lambda_i \quad (7)$$

At this time, the service rate is μ_i . The number of the data packets is Y_i given by Equation (8) after completion of the service.

$$Y_i = T_i^m \mu_i \tag{8}$$

Therefore, packets queue length of node i is L_i^m , which is the difference between the number of effective arrival and leaving, and minus the number of packets being processed.

$$L_i^m = X_i - Y_i - 1 \tag{9}$$

Putting Equations (6), (7), and (8) into Equation (9) will result in Equation (5). Therefore, Theorem 1 is proved. \square

Theorem 2. For any transmission node i in WSNs, if packets queue length of transmission node is L_i^m , then the following relation is obtained.

$$\lambda_i^e + \sum_{k=1}^{m-1} \lambda_k p_{ki} = \frac{L_i^m}{T_i^m} + \mu_i \tag{10}$$

Proof of Theorem 2 is relatively simple, using Corollary 1, and with reference to the proof of Theorem 1.

4.2. Equivalent Queueing Network Model

The packet buffer size should be consistent with the length of the queue in the queueing model of WSNs. When the queue length reaches the maximum, the packet streams are stopped, resulting in queueing network being blocked [31,32]. When a data packet transfers from one queue to another queue and if the path is full, the packet will be blocked in by the just completed service in the queue. Then the blocked node cannot handle any other data packets until the destination node services, where there is a free packet buffer before they can lift the blocking. This situation is called Transfer Blocking.

Transfer blocking makes the internal arrival process and service process of node complicated. The blocking rule is that blocking should occur after service. Therefore, the queue will not be cached, only waiting on the link path. Some researchers [33,34,35,36] have discussed about thinking of adding holding nodes in the queueing networks, that adds the imaginary limitless capacity nodes, which may occur on the blocking path in the infinite queueing networks. The basic idea is to remove the blocking server, that is unblocked and save it in the holding nodes. As shown in Figure 1, the queueing network model for the topology of WSN is expanded to include holding nodes. Equivalent queueing network models are shown in Figure 4. M/M/1/ ∞ -type queues are added holding nodes. (Note: When added to holding nodes, assuming that the original node does not exist, blocking is established.)

Intervals between the arrival time accord with the exponential distribution. Thus, we calculate the real effective arrival rate nodes, and its performance evaluation is possible.

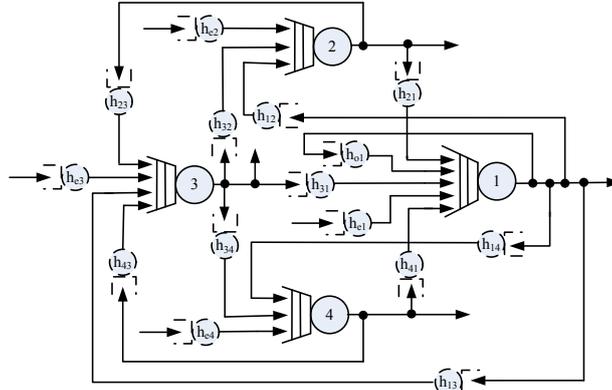


Fig. 4. Queueing network model with holding nodes.

4.3. Queueing Model Analysis with Blocking Probability

When the holding nodes are added in queueing network, the packets that did not receive timely services are stored in the queue of holding nodes, as waiting for an empty target node. Total effective arrival rate is equal to the external arrival rate λ_j^e and the internal arrival rate of nodes i^1, i^2, \dots, i^A after considering blocking nodes. Then, the queueing network model with blocking probability is shown in Figure 5. Below we discuss flow balance of arrival and leaving data packets in the queueing network model.

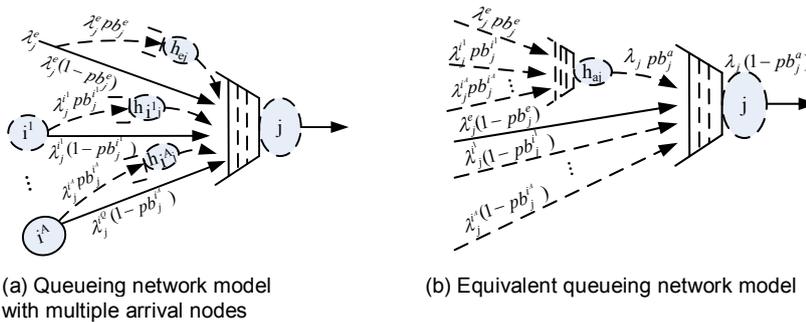


Fig. 5. Queueing network model with blocking probability.

Figure 5(a) shows that multiple arrival nodes are blocked in node j , and Figure 5(b) shows an equivalent model of Figure 5(a).

Therefore, the external effective arrival rate $\bar{\lambda}_j^e$ of node j is obtained as in Equation (11).

$$\bar{\lambda}_j^e = \lambda_j^e (1 - pb_j^e) \quad (11)$$

The effective data packets stream from node i to node j as shown in Equation (12).

$$\lambda_{ij} = \lambda_i p_{ij} (1 - pb_{ij}) \quad (12)$$

Let $\bar{\lambda}_j$ be the effective internal arrival rate of node j , which is equal to the sum of effective internal arrival rate from independent internal nodes i^1, i^2, \dots, i^A .

$$\bar{\lambda}_j = \sum_{i \in A} \lambda_{ij} \quad (13)$$

Total effective arrival rate with probability pb_j^a of node is obtained by Equation (14).

$$\bar{\lambda}_j = \lambda_j (1 - pb_j^a) \quad (14)$$

According to Corollary 1 and Corollary 2, we can obtain a flow balance equation of queueing network with blocking.

$$\bar{\lambda}_j = \bar{\lambda}_j^e + \bar{\lambda}_j \quad (15)$$

Equation (16) is derived from applying equations (11) (12) (13) (14) into Equation (15).

$$\lambda_j (1 - pb_j^a) = \lambda_j^e (1 - pb_j^e) + \sum_{i \in A} \lambda_i p_{ij} (1 - pb_j^i) \quad (16)$$

In order to obtain the effective arrival rate of node j , the calculations of blocking probability pb_{ij} , pb_j^a and pb_j^e are needed. The three blocking probabilities are calculated, the specific derivation is shown as in [37].

Now we can determine the performance parameters of each type node according to the connection between nodes. In this paper, we use the approximate calculation. First, the initial values of blocking probabilities are given, then the algorithm performs in a limited iteration times. When the effective arrival rate and departure rate tends to reach equilibrium, the iterative algorithm is finished.

5. Total Arrival Rate of Nodes and Approximate Algorithm

Blocked nodes are released by adding holding nodes of infinite capacity in the queueing network model. Processing time of blocked node is the blocked time, and thus we can describe arrival and service process of nodes in the equivalent queueing network model. In a lot of practical application and engineering experiments, we found that when WSN node communication enters into a stable state, the average arrival rate of node tends to be a constant value. We designed an iterative method, such as shown in Algorithm 1. We set initial values to the network status, and then gradually revised the last time arrival rate by our iterative method. In the end, a system was approaching to reach equilibrium. The reduction algorithm is as follows.

Algorithm 1
<p>Begin</p> <p>Step 1. According to transition probability, each node connection in the queueing network model is obtained. External arrival rate λ_j^e (j is the number of nodes) is determined.</p> <p>Step 2. Initialize n nodes with the total arrival rate λ_j^0 ($1 \leq j \leq m$).</p> <p>Step 3. For queue j in queueing network, calculate the arrival rate λ_j of node j:</p> <p>Step 3.1. If node j is a transmission node or boundary node,</p> $\lambda_j^1 = \lambda_j^e + \sum_{i=1}^m \lambda_i p_{ij} \quad (17)$ <p>go to Step 3.3. Otherwise, it is executed orderly.</p> <p>Step 3.2. node j is sink node,</p> $\lambda_j^1 = \lambda_j^e + \sum_{i=1}^{m-1} \lambda_i p_{ji} + \lambda_i p_{ii} \quad (18)$ <p>Step 3.3. The outputting rate of node j is calculated by Corollary 1 and Corollary 2.</p> <p>Go to Step 3.1, until the difference of the internal arrival rate for two computing (before and after) is less than a certain value (error limit of our calculations is 10^{-4}).</p> <p>Step 4. After calculating the arrival rate of all nodes, if the difference of the internal arrival rate for two computing values is less than a value (10^{-4}), go to Step 5. Otherwise, use λ_j^1 instead λ_j^0, and go to Step 3.1, continue by</p>

iterative calculation.

Step 5. Return the total arrival rate λ_j^n of each node.

End

The time complexity of Algorithm 1 is $O(n*m)$, where m is the number of queueing network nodes and n is the number of iterative algorithms.

The total arrival rate of each node was obtained by Algorithm 1, but it is not the effective arrival rate, because the blocking probability of the nodes is not considered. Next we will have the numerical results of Algorithm 1 as the initial value of Algorithm 2, and then solve the blocking probability and system performance indicators. The reduction Algorithm 2 is as follows.

The calculation of Algorithm 2 mainly focused on the loop in Steps 3 to 6 of the cycle. The time complexity of Algorithm 2 is $O(n*m*N)$, where N is the packet buffer size of node. When the iterative algorithm converges, the WSN performance parameters are outputted. According to that we can predict the actual operation of WSNs. Thus, the hardware design for the WSN node is guided by the performance parameters.

Algorithm 2

Begin

Step 1. The equivalent queueing network model is expanded by adding holding nodes in wireless sensor queueing networks. The total arrival rate of Algorithm 1 is the initial input value of Algorithm 2 for each node.

Step 2. Blocking probabilities of each node are initialized, and means and variances of internal arrival time are calculated.

Step 3. Calculate the utilization and steady-state probability of the nodes, where $n_i = \{1, 2, \dots, N\}$, that means the number of data packet buffer of each node. When the system reaches a steady state, we assume that the probability of queue i in state n_i is $p_i(n_i)$, which is obtained by:

$$p_i(n_i) = \frac{\rho_i^{n_i} (1 - \rho_i)}{1 - \rho_i^{N+1}} \quad (19)$$

where $\rho_i = \frac{\lambda_i}{\mu_i}$. Specific derivations of Equation (19) can be found in [38,39]. According to Jackson's theorem [30], the status of node i and the status of all other nodes are independent. Thus, we can get the steady-state probabilities of any node in the link path.

Step 4. Calculate the blocking probabilities.

Step 5. Correct means of the arrival time interval and variance of nodes

using blocking probabilities.

Step 6. If the difference of the internal arrival rate for two computing (before and after) is less than a certain value (error limit of our calculations is 10^{-4}), then go to Step 7. Otherwise go to Step 3 followed by iterative calculation.

Step 7. Return to the node utilizations with a blocking when system is stable.

End

6. Numerical Calculation and Experimental Results

According to the preceding analysis, we studied a WSN topology for temperature monitoring. The performance parameters of WSN nodes are calculated using the iterative algorithm proposed in Section 5. By setting different packet buffer capacity sizes of nodes, the relationship curves between utilization (ρ_i) and data packet buffer size (N_i) for transmission nodes, boundary nodes and sink node are obtained. In order to rationally allocate resources, the maximum utilizations of nodes are ensured within limited resources. According to the relationship curves between utilization (ρ_i) and data packet buffer size (N_i), the values of packet buffer capacity size for transmission nodes, boundary nodes and sink nodes are set.

6.1. A WSN Topology

We have designed the WSNs for temperature monitoring. Its typical topology is shown in Figure 6. It consists of many clusters, each of which comprises a mixed structure from the ring and star network topologies. Information between clusters is communicated by sink nodes.

From the WSN we can conclude that all the clusters can be divided into two categories. (i) Boundary cluster: It is located in the boundaries of the network by the cluster nodes, transfer nodes and boundary nodes; Cluster 1 is shown in Figure 6. (ii) Internal cluster: it is located within the network only by the sink nodes and transfer nodes. Cluster 2 is shown in Figure 6. According to the packets statistics in the engineering practice, amount of tasks for transmission node is the same in the boundary and internal clusters. Therefore, we studied Cluster 1, in which performance of internal nodes will be analyzed. Cluster 1 is comprised of sink node 1, transmission nodes 1–4, and boundary nodes 5–6. Ring structure is comprised of nodes 1–6, and node 7 is in the center of the other six nodules that shows a similar star, which is with 6 nodes to communicate. In practical engineering in applications, how to set the buffer

size to obtain the optimal performance of the nodes for hardware design of WSNs is a practical application problem that needs to be solved.

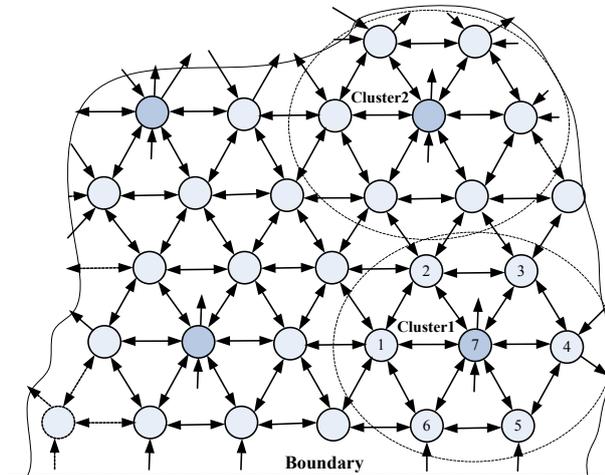


Fig. 6. A WSN topology for temperature monitoring.

6.2. Node Transition Probabilities and Total Arrival Rate

The node transition probabilities for the topology of Figure 6 are obtained from packet statistics in the engineering practice of WSNs for temperature monitoring. Thus the transition probability matrix is as follows.

$$P = \begin{bmatrix} 0 & 0.2 & 0 & 0 & 0 & 0.2 & 0.5 \\ 0.2 & 0 & 0.2 & 0 & 0 & 0 & 0.5 \\ 0 & 0.2 & 0 & 0.2 & 0 & 0 & 0.5 \\ 0 & 0 & 0.2 & 0 & 0.2 & 0 & 0.5 \\ 0 & 0 & 0 & 0.2 & 0 & 0.2 & 0.6 \\ 0.2 & 0 & 0 & 0 & 0.2 & 0 & 0.6 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0 \end{bmatrix} \quad (20)$$

In the distributed temperature acquisition process, for each sensor node placed in different positions, the amount of transmitted data is different. For example, node 1, node 2, node 3 and node 4 are transmission nodes within the WSN. Thus, the external arrival rates of the nodes are same. Node 5 and node 6 are the boundary nodes (do not allow other nodes from outside). Node 7 is the sink node. The external arrival rates of these nodes are different.

Table 2. Arrival rate for each node in the time T_i^m

Node number	External arrival rates λ_i^e (p/s)	Total arrival Rates λ_i (p/s)
1	10	29.6577
2	10	31.1161
3	10	31.1161
4	10	29.6577
5	3	22.3661
6	3	22.3661
7	2	89.6131

According to the input of the external arrival rates, the total arrival rate of node i can be calculated using Algorithm 1, as shown in Table 2.

6.3. Evaluation for Packet Buffer Capacity of Nodes

In WSNs, packet buffer capacity of nodes settings that affects the efficiency of the whole network system is an important factor. If packet buffer capacity is too small it will cause serious blocking to some link paths of the system, and lead to a low efficiency of data processing and transmission. If packet buffer capacity is too large, it will take up too much of the hardware resources, resulting in an increased cost of the hardware node. And the energy consumption will increase, causing a link failure to individual nodes due to energy depletion. Therefore, we design hardware nodes that make the data packet buffer capacity to reach the optimal settings.

Other input parameters of queueing network model for wireless sensor are shown in Table 3, where CA_i^e is the independent external Poisson arrival Variance of node i , μ_i is the service rate of node i , CS_i is the service variance of node i and N_i is the size of packet buffer capacity which ranges from 1 to 30.

Table 3. Input parameters for model

i	CA_i^e	μ_i	CS_i	N_i
1	10	40	14	1~30
2	12	40	16	1~30
3	12	40	16	1~30
4	10	40	14	1~30
5	7	30	10	1~30
6	7	30	10	1~30
7	16	100	30	1~30

According to of the approximate iterative Algorithm 2 in Section 5, packet buffer size (N_i) and node utilization (ρ_i) are calculated.

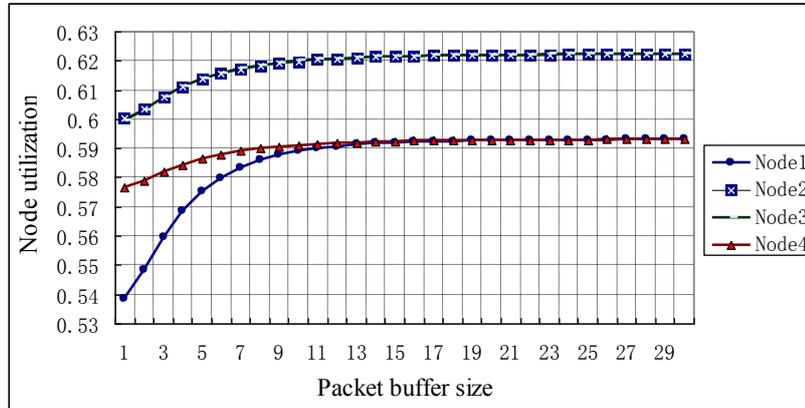


Fig. 7. Relationship curves between utilization and buffer size for transmission nodes.

The relationship curves between utilization and packet buffer size for transmission nodes are shown in Figure 7. In WSNs for temperature monitoring, node 1, node 2, node 3 and node 4 are the transmission nodes, in which the main function of the network is to collect data and transfer it to other nodes. With the increase of packet buffer size, increased speed of node 1 is the fastest, increased speed of node 4 has slowed down compares to other three nodes. When the packet buffer sizes are increased to $N_1 = N_4 = 14$, curves for nodes 1 and 4 coincide. Next, with the increase of packet buffer size, the curves in almost horizontal axis tend to become parallel. At this time, if we increase the packet buffer sizes, utilization for nodes will not have any impact. This point can be set as the value of packet buffer size then the node utilization is optimal. From point of view for the layout of the WSNs, nodes 1 and 4 are adjacent to the boundary nodes, transmission rates of data packets are almost the same. Therefore, when the system reaches equilibrium, packet buffer sizes in use become the same.

The curves between utilization and packet buffer size for nodes 2 and 3 are in coincidence in figure 8. When the packet buffer sizes are increased to $N_2 = N_3 = 11$, the curves in almost horizontal axis tend to become parallel. At this time, if we increase the packet buffer sizes, utilization for nodes will not have any impact. This point can be set as the value of packet buffer size then the node utilization is optimal. From the point of view of the layout of the WSNs, nodes 2 and 3 are internal nodes of the WSN, transmission rates of data packets are almost the same. Therefore, the relationship curves between utilization and buffer size for boundary nodes are the same.

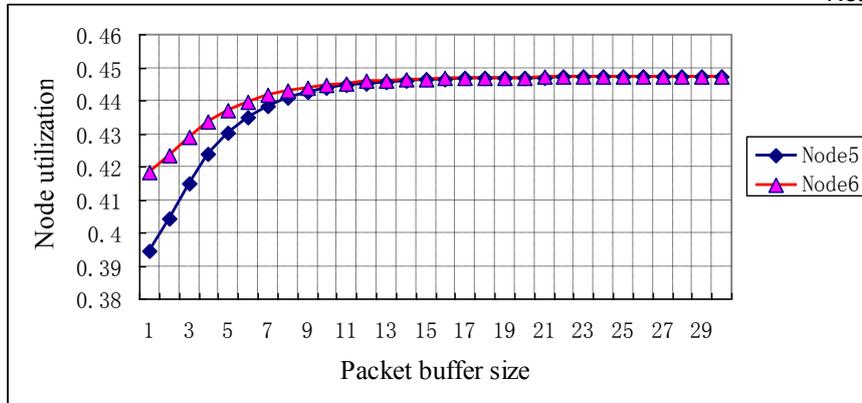


Fig. 8. Relationship curves between utilization and buffer size for boundary nodes.

The relationship curves between utilization and packet buffer size for boundary nodes are shown in Figure 8. With the increase of packet buffer size, beginning curves are relatively steep. The increased speed of curve for node 5 is the fastest; the increased speed of curve for node 6 has slowed down compared to curve for node 5. According to this situation, we can speculate that, when node 5 is set to a smaller packet buffer size, a more serious blocking occurred in the link paths for the node, resulting in relatively low node utilization. When the packet buffer sizes increased to $N_5 = N_6 = 13$, curves for nodes 5 and 6 are coincidence. And following the increase of packet buffer size, the curves in almost horizontal axis tend to become parallel. At this time, if we increase the packet buffer sizes, utilization for nodes will not have any impact. If this point can be set as the value of packet buffer size, then the node utilization is optimal. From the point of view of the layout of the WSNs, nodes 5 and 6 are located on the outside of the WSN, and they are boundary nodes. The transmission rates of data packets are almost the same. Therefore, when the system reaches equilibrium, the packet buffer sizes in use are the same.

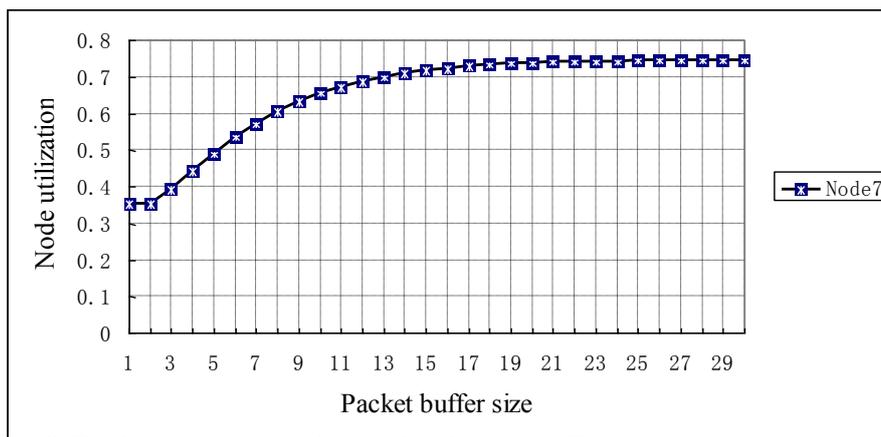


Fig. 9. Relationship curves between utilization and buffer size for sink node.

For sink node 7, on the one hand, it is responsible for collecting the data packets of adjacent transmission nodes and boundary nodes. On the other hand, information between clusters is communicated, and a partial data processing is completed by sink node. The relationship curves between utilization and packet buffer size for sink node are shown in Figure 9.

With the increase of packet buffer size, beginning curves are relatively steep. As per this situation, we can speculate that, when sink node 7 is set to a smaller packet buffer size, its utilization is relatively low. After the packet buffer sizes are increased to $N_7 = 19$, and following this, the curves in almost horizontal axis tend towards parallel. At this time, if we increase the packet buffer sizes, utilization for nodes will not have any impact. If this point can be set as the value of packet buffer size, then the node utilization is optimal.

6.4. Comparison of Experimental Data and Model Calculation Data

According to the size of the nodes through the analysis of packet buffer size in Section 6.3, the experimental environment was designed. The packet buffer sizes for transmission nodes are set as $N_1 = N_4 = 14$, $N_2 = N_3 = 11$, respectively. The packet buffer sizes for boundary nodes are set as $N_5 = N_6 = 13$, respectively. The packet buffer size for sink node is set as $N_7 = 19$. The simulation environment using NS2 software combined with random arrival derived the algorithm [39]. The experimental simulation for arrival rate of node with holding nodes was carried out. Figure 10 shows the comparison of nodes between the effective arrival rates.

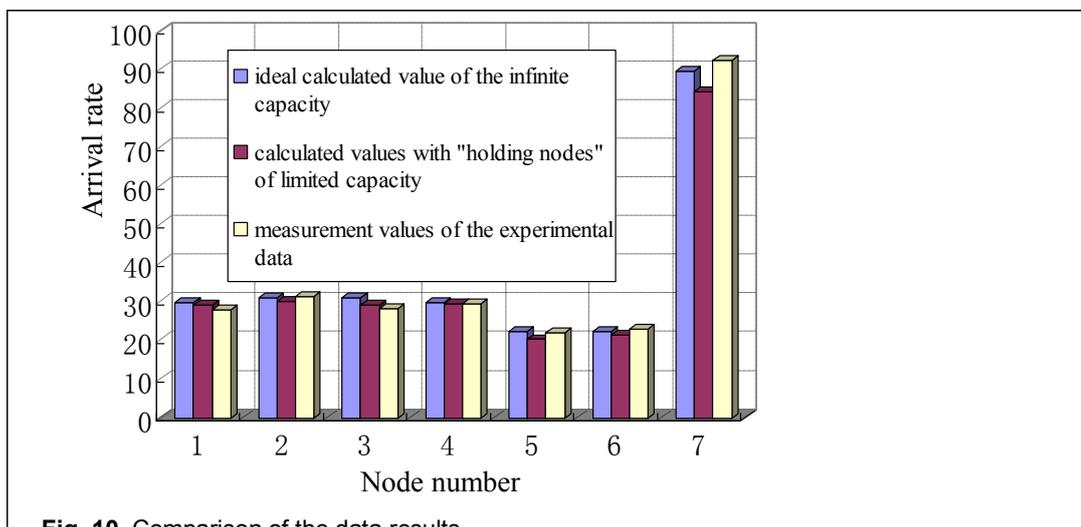


Fig. 10. Comparison of the data results.

In Figure 10, the arrival rates of the three situations are described: (i) Assuming queueing network model for wireless sensor works in an ideal state, the blocking probabilities $\{pb_{ij}, pb_j^a, pb_j^e\}$ always equal 0, which is called the ideal calculated value of the infinite capacity nodes. (ii) Queueing network model for wireless sensor maintained by adding a finite number of nodes (based on the values obtained in Section 6.3), considering the given blocking probabilities $\{pb_{ij}, pb_j^a, pb_j^e\}$, the values of the effective arrival rate are calculated, these are called the calculated values with holding nodes of limited capacity. (iii) Experimental environment is created and random function for arrival rate is designed. Therefore, the actual arrival rates are obtained by statistical calculation, these are called the measurement values of the experimental data. Figure 10 clearly describes the comparison of data among the three situations. We can see that errors between the ideal calculated value of the infinite capacity nodes and the calculated values with holding nodes of limited capacity are very small, the maximum error being 6.12%. This proves that by adding finite holding nodes to a queueing network model, and obtained equivalent queueing network model can replace the infinite capacity model for performance analysis. The maximum error between the measurement values of the experimental data and the calculated values with holding nodes of limited capacity is 9.56%. As in the model calculation error is less than 10%, almost consistent indicators for equivalent queueing network model can be obtained with the actual operation of the WSN.

7. Conclusions

In allusion to differ transmission nodes, boundary nodes and the sink node in WSNs, their processing data and the task capacity are different. And thus, before designing the hardware for nodes, an evaluation of packet buffer sizes is critical for the best performance. Based on this problem, we proposed an open queueing network mode with M/M/1/N queues for modeling WSNs. In order to evaluate congestion situation in the queueing network, and get real effective arrival rates and transfer rates in the model, holding nodes were added in the queueing network model and an equivalent queueing network model is expanded. The arrival rates when system reaches a steady state in WSNs are obtained using approximate iterative algorithm. The obtained steady-state parameters will be entered into with the blocking queueing network model. Blocking probability and system performance indicators of each node are calculated using approximate iterative algorithm for blocking probability. The ideal calculated value of the infinite capacity nodes, the calculated values with holding nodes of limited capacity and the measurement values of the experimental data are compared. The consistency is verified for calculated results of model and experimental data in WSNs. The results show that the method, which is used to analyze node performance and ensure that packet buffer sizes are of reasonable configuration using queueing network

Tie Qiu, Lin Feng, Feng Xia, Guowei Wu, and Yu Zhou

model, provides a theoretical basis for design of high cost-effective hardware nodes for modeling large-scale WSNs. This work has important guiding significance for hardware design and performance evaluation of WSNs system.

This paper presents a modeling for only a single-server model in WSN and a method for calculating the packet buffer capacity size of nodes. However, the sink node requires a higher performance. Recently, there has been convergence of multiple processor nodes that can be used for M/M/m/N queues, which are also multi-server queues. In addition, for large-scale WSNs, if the clusters are set to be the priority, it will effectively improve the performance of WSNs. This will be our follow-up research.

Acknowledgment. This work was supported in part by Natural Science Foundation of China under Grant No. 61173163, 60903153 and 61173179, Program for New Century Excellent Talents in University (NCET-09-0251), the Fundamental Research Funds for the Central Universities, and the SRF for ROCS, SEM.

8. References

1. Onur, E., Ersoy, C., Deliç, H., Akarun, L.: Surveillance with wireless sensor networks in obstruction: Breach paths as watershed contours. *Computer Networks*, Vol. 54, No. 3, 428-441. (2010)
2. Lasasmeh, S., M., Conrad, J., M.: Time synchronization in wireless sensor networks: A survey. *Conference Proceedings-IEEE SoutheastCon 2010: Energizing Our Future*, 242-245, Concord. (2010)
3. Son, J., H., Lee, J., S., Seo, S., W.: Topological key hierarchy for energy-efficient group key management in wireless sensor networks. *Wireless Personal Communications*, Vol. 52, No. 2, 359-382. (2010)
4. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. *Computer Networks*, Vol. 52, No. 12, 2292-2330. (2008)
5. Akyildiz, I., F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks", *IEEE Communications Magazine*, Vol. 40, No. 8, 102-114. (2002)
6. Alnabelsi, S., H., Almasaeid, H., M., Kamal, A., E.: Optimized sink mobility for energy and delay efficient data collection in FWSNs. *15th IEEE Symposium on Computers and Communications*, 550-555, Riccione. (2010)
7. Sheu, J., P., Sahoo, P., K., Su, C., H., Hu, W., K.: Efficient path planning and data gathering protocols for the wireless sensor network. *Computer Communications*, Vol. 33, No. 3, 398-408. (2010)
8. Gao, S., Zhang, H., Song, T., Wang, Y.: Network lifetime and throughput maximization in wireless sensor networks with a path-constrained mobile sink. *2010 International Conference on Communications and Mobile Computing*, 298-302, Shenzhen. (2010)
9. Eun, D., Y., Wang, X.: Achieving 100% throughput in TCP/AQM under aggressive packets marking with small buffer. *IEEE/ACM Transactions on Networking*, Vol. 16, No. 4, 945-956. (2008)
10. Sridharan, A., Krishnamachari, B.: Maximizing network utilization with max-min fairness in wireless sensor networks. *Wireless Networks*, Vol. 15, No. 5, 585-600. (2009)

A Packet Buffer Evaluation Method Exploiting Queueing Theory for Wireless Sensor Networks

11. Vupputuri, S, Rahuri, K., K., Murthy, C., S., R.: Using mobile data collectors to improve network lifetime of wireless sensor networks with reliability constraints. *Journal of Parallel and Distributed Computing*, Vol. 70, No. 7, 767-778. (2010)
12. Han, J., Choi, S., Park, T.: Maximizing lifetime of cluster-tree ZigBee networks under end-to-end deadline constraints. *IEEE Communications Letters*, Vol. 14, No. 3, 214-216, (2010)
13. Subramanian, R. Fekri, F.: Unicast throughput analysis of finite-buffer sparse mobile networks using Markov chains. 46th Annual Allerton Conference on Communication, Control, and Computing, 1161-1168, Monticello. (2008)
14. Li, G., H., Zhu, C., M., Li, X.: Application of Chaos theory and Wavelet to Modeling the Traffic of Wireless Sensor Networks. 2010 International Conference on Biomedical Engineering and Computer Science, 1-4, Wuhan, (2010)
15. Escheikh, M., Barkaoui, K.: Opportunistic MAC layer design with Stochastic Petri Nets for multimedia ad hoc networks. *Concurrency Computation Practice and Experience*, Vol. 22, No. 10, 1308-1324. (2010)
16. Moon, C. Chung, W.: Design of navigation behaviors and the selection framework with generalized stochastic petri nets toward dependable navigation of a mobile robot. 2010 IEEE International Conference on Robotics and Automation, 2989-2994. Anchorage. (2010)
17. Shareef, A., Zhu, Y.: Energy modeling of processors in wireless sensor networks based on petri nets. 37th International Conference on Parallel Processing Workshops, 129-134, Portland. (2008)
18. Strelen, J., C., Bark, B., Becker, J., Jonas, V.: Analysis of queueing networks with blocking using a new aggregation technique. *Annals of Operations Research*, No. 79, 121-142. (1998)
19. Liehr, A., W., Buchenrieder, K., J.: Simulating inter-process communication with Extended Queueing Networks. *Simulation Modelling Practice and Theory*, Vol. 18, No. 8, 1162-1171. (2010)
20. Bisnik, N., Abouzeid, A., A.: Queueing Network Models for Delay Analysis of Multihop Wireless Ad Hoc Networks. *Ad Hoc Networks*, Vol. 7, No. 1, 79-97. (2009)
21. Kouvatso, D., Awan, I.: Entropy maximisation and open queueing networks with priorities and blocking. *Performance Evaluation*, Vol. 51, No. 2-4, 191-227. (2003)
22. Awan, I.: Analysis of multiple-threshold queues for congestion control of heterogeneous traffic streams. *Simulation Modelling Practice and Theory*, Vol. 14, No. 6, 712-724. (2006)
23. Özdemira, M., McDonald, A., B.: On the performance of ad hoc wireless LANs: A practical queueing theoretic model. *Performance Evaluation*, Vol. 63, No. 11, 1127-1156. (2006)
24. Mann, C., R., Baldwin, R., O., Kharoufeh, J., P., Mullins, B., E.: A queueing approach to optimal resource replication in wireless sensor networks. *Performance Evaluation*, Vol. 65, No. 10, 689-700. (2008)
25. Qiu, T., Wang, L., Feng, L., Shu, L.: A new modeling method for vector processor pipeline using queueing network. 5th International ICST Conference on Communications and Networking, 1-6, Beijing. (2010)
26. Qiu, T., Xia, F., Lin, F., Wu, G., Jin, B.: Queueing theory-based path delay analysis of wireless sensor networks. *Advances in Electrical and Computer Engineering*, Vol. 11, No. 2, 3-8. (2011)
27. Boucherie, R., J., Dijk, N., M.: *Queueing Networks: A Fundamental Approach*. Springer, 1st Edition. (2010)

Tie Qiu, Lin Feng, Feng Xia, Guowei Wu, and Yu Zhou

28. Chiasserini, C., F., Garetto, M.: An Analytical Model for Wireless Sensor Networks with Sleeping Nodes. *IEEE Transactions on Mobile Computing*, Vol. 5, No. 12, 1706-1718. (2006)
29. Paul, S., Nandi, S., Singh, I.: A dynamic balanced-energy sleep scheduling scheme in heterogeneous wireless sensor network. *Proceedings of the 2008 16th International Conference on Networks*, 1-6, New Delhi. (2008)
30. Raymond, D. R., Marchany, R., C., Brownfield, M., I., Midkiff, S., F.: Effects of Denial-of-Sleep Attacks on Wireless Sensor Network MAC Protocols. *IEEE Transactions on Vehicular Technology*, Vol. 58, No. 1, 367-380. (2009)
31. Almeida, D., D., Kellert, P.: Analytical queueing network model for flexible manufacturing systems with a discrete handling device and transfer blockings. *International Journal of Flexible Manufacturing Systems*, Vol. 12, No. 1, 25-57. (2000)
32. Osorio, C., Bierlaire, M.: An analytic finite capacity queueing network model capturing the propagation of congestion and blocking. *European Journal of Operational Research*, Vol. 196, No. 3, 996-1007. (2009)
33. Kerbache, L., Smith, J., M.: Asymptotic behavior of the expansion method for open finite queueing networks. *Computers and Operations Research*, Vol. 15, No. 2, 157-169. (1988)
34. Tahilramani, H., Manjunath, D., Bose, S., K.: Approximate analysis of open network of GE/GE/m/N queues with transfer blocking. *Proceedings of the 1999 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 164-172, Maryland. (1999)
35. Brandwajn, A., Begin, T.: Higher-order distributional properties in closed queueing networks. *Performance Evaluation*, Vol. 66, No. 11, 607-620. (2009)
36. Andriansyah, R., Woensel, T., V., Cruz, F., R., B., Duczmal, L.: Performance optimization of open zero-buffer multi-server queueing networks. *Computers and Operations Research*, Vol. 37, No. 8, 1472-1487. (2010)
37. Kouvasos, D.: Maximum entropy analysis of queueing network models. *Lecture Notes in Computer Science, Performance Evaluation of Computer and Communication Systems*, Vol. 729, 245-290. (1993)
38. Lefebvre, M.: *Queueing Theory, Applied Stochastic Processes*. Universitext, Springer, 315-356. (2007)
39. Ross, S., M.: *Introduction to Probability Models*, 10th Edition, Academic Press. (2009)

Tie Qiu is a Lecturer and Ph. D. candidate in computer science at Dalian University of Technology, China. His research interests cover embedded high performance computing, wireless sensor networks and systems modeling.

Lin Feng is a Professor in School of Innovation Experiment, Dalian University of Technology, China. His research interests cover data mining, wireless sensor networks and Internet of Things.

Feng Xia is an Associate Professor and Ph.D Supervisor in School of Software, Dalian University of Technology, China. He is the (Guest) Editor of several international journals. He serves as General Chair, PC Chair, Workshop Chair, Publicity Chair, or PC Member of a number of conferences. Dr. Xia has authored/co-authored one book and over 100 papers. His

A Packet Buffer Evaluation Method Exploiting Queueing Theory for Wireless Sensor Networks

research interests include cyber-physical systems, mobile and social computing, and intelligent systems. He is a member of IEEE and ACM.

Guowei Wu received B.E. and Ph.D. degrees from Harbin Engineering University, China, in 1996 and 2003, respectively. He was a Research Fellow at INSA of Lyon, France, from September 2008 to March 2010. He has been an Associate Professor in School of Software, Dalian University of Technology, China, since 2003. Dr. Wu has authored three books and over 20 scientific papers. His research interests include embedded real-time system, cyber-physical systems, and wireless sensor networks.

Yu Zhou received B.E. degree from Dalian University of Technology, China. Currently he is a Master student in School of Software, Dalian University of Technology. His research interest covers wireless sensor networks and Internet of Things.

Received: February 27, 2011; Accepted: April 25, 2011.

On the Efficiency of Cluster-based Approaches for Motion Detection using Body Sensor Networks

Kun-chan Lan, Chien-Ming Chou, Tzu-nung Wang, and Mei-Wen Li

Computer Science and Information Engineering
National Cheng Kung University
No.1, University Road, Tainan City 701, Taiwan
klan@csie.ncku.edu.tw

Abstract. Body Sensor Networks (BSN) are an emerging application that places sensors on the human body. Given that a BSN is typically powered by a battery, one of the most critical challenges is how to prolong the lifetime of all sensor nodes. Recently, using clusters to reduce the energy consumption of BSN has shown promising results. One of the important parameters in these cluster-based algorithms is the selection of cluster heads (CHs). Most prior works selected CHs either probabilistically or based on nodes' residual energy. In this work, we first discuss the efficiency of cluster-based approaches for saving energy. We then propose a novel cluster head selection algorithm to maximize the lifetime of a BSN for motion detection. Our results show that we can achieve above 90% accuracy for the motion detection, while keeping energy consumption as low as possible.

Keywords: body sensor network, motion detection, energy conservation, KNN.

1. Introduction

In recent years Body Sensor Networks (BSNs) have attracted increasing attention for their use in remote healthcare such as remotely monitoring the elderly and children. A BSN is composed of a number of tiny wireless sensors placed on the user's body. Sensor nodes can transmit data to a remote server for online or offline analysis via various wireless technologies (e.g., IEEE 802.15.4 [1] or Bluetooth) so that doctors can monitor the patient's physiological states in real-time. BSNs have been widely used for various healthcare applications [2, 3, 4, 5], motion detection/analysis [6, 7, 8, 9], and gaming [10].

A critical issue when designing a BSN is the energy consumption, since wireless sensors typically employ small batteries for ease of deployment. A number of works have studied the issue of reducing the energy consumption of BSN. For example, Xiao *et al.* [3] proposed a method to reduce the transmission power at the physical layer without sacrificing the reliability for a

single-hop body area network. Lamprinos *et al.* [11] and Omeni *et al.* [12] employed a TDMA-based approach to avoid collisions. In addition, with TDMA a node only needs to turn on its radio in its assigned time slot to save energy. Furthermore, Xia *et al.* [2] proposed a prediction-based data transmission scheme for static data (e.g., blood pressure and heartbeat). In their method, both the sensor node and the base station (BS) periodically and independently execute an identical predictor (i.e., dual prediction) and thus they will obtain the same estimate. If the error between real and predicted data is acceptable, the sensor node just updates the state of the system using the predicted data without actually transmitting data to the base station. Similarly, the BS also simply updates the system using the predicted data unless it receives the real data from the sensor node. However, this approach only considered an individual sensor node instead of the entire BSN.

Cluster-based algorithms have recently been proposed for sensor networks, and showed promising results. In these approaches, nodes are organized into clusters and one node is then chosen as the cluster head (CH). The task of the CH is to aggregate sensor data and then transmits this to BS while the other cluster nodes (CNs) communicate only with the CH. Given that it typically consumes more power to communicate with BS than with other cluster nodes, a CH will consume more energy than a CN does. Therefore, each sensor node will take turns to serve as the CH in order to evenly distribute the energy load. Abbasi and Younis [13] have surveyed clustering algorithms and classified them based on clustering attributes such as the cluster property, cluster head capability, and clustering process. Most prior approaches randomly selected their CHs. For instance, LEACH [14] and DMCLUSTER [15] choose CHs probabilistically or based on the nodes' remaining energy. However, these prior works cannot guarantee to uniformly distribute the energy load and minimize the total intra-cluster energy consumption at the same time.

The objective of this work is to prolong the lifetime of a body sensor network for motion detection by distributing the energy load evenly among sensors while, at the same time, minimizing the total intra-cluster energy consumption. In this paper we focus on an application of BSN: posture detection. We develop a novel CH selection algorithm to maximize the lifetime of the BSN. We utilize the k-nearest neighbors (KNN) method to detect the motion of the user. In this work, we classify nodes in a BSN into two types: fixed and moving. Fixed nodes are nodes whose positions remain relatively static (i.e. unchanged) when the user changes to a new posture. We analyze different parameters such as the ratio between the number of moving and fixed nodes and the effect of CH selection on the energy-saving. We propose an algorithm for selecting a CH that can minimize the intra-cluster energy consumption by considering the relationship between inter-node distance and allowable power levels. In addition, a threshold-based method is developed to avoid a certain node from depleting its energy quickly by being frequently selected as the CH. Our contributions are threefold: 1) We discuss and analyze the amount of energy that can be saved using a cluster-based approach. 2) We propose a novel cluster head selection algorithm by

considering the required power level for sending a packet from the cluster nodes to the cluster head for a given posture. 3) We implement our cluster-head selection algorithm on a testbed for motion detection using KNN and show that we can achieve high accuracy (above 90%) while saving energy for a body sensor network.

The remainder of this paper is organized as follows. In Section II we review the related work. We briefly describe different motion detection methodologies in Section III. In section IV, we propose a methodology to minimize the transmission cost in a BSN while balancing each node's energy consumption. The results of the experiment are shown in section V and we conclude this paper in Section VI.

2. Related Work

In this section, we describe the related work. Our work builds on prior research on body sensor networks, motion detection, and energy conservation.

2.1. Body Sensor Network

Numerous studies have proposed the use of body sensor networks (BSN) for healthcare applications [3, 4, 6]. Earlier works also pointed out that QoS [5] and energy conservation [2, 16] are key research issues for the BSN, since the former could affect life-or-death matters while the latter decides the lifetime of the network especially for those sensors embedded in a patient's body. In addition, BSN can be also applied to sports applications using inertial sensors to monitor the trainee's posture during actions such as walking/running [7], golf swings [8], and hand swings [9]. However, most of these prior studies either can only identify basic postures (sitting, standing, walking, and running) or can only detect single motions. While using some techniques such as machine learning [9] to detect more complex motion is possible, this might increase the latency or computation cost of the BSN.

2.2. Motion/Posture Detection

Various kinds of sensors can be used to capture the motion of a user, and these are briefly described as follows: Acoustic trackers can use high-frequency sound to triangulate a source [17, 18, 19]. These systems rely on line-of-sight (LOS) between the source and the sensors, and may therefore suffer from interference when surrounded by hard walls or other acoustic signal/noise. Inertial systems employ devices such as accelerometers or gyroscopes to measure positions and angles. They are often used in conjunction with other systems to provide updates and improvements of

measurements, since they only measure relative changes instead of absolute positions. Most prior works focused on analyzing the characteristics of inertial sensors to capture some basic postures or daily behaviors [20, 21, 22]. Image-based systems use cameras to capture the movements of a subject who is attached with retro-reflective markers. The number of cameras used depends on what type of motions are being captured [23, 24, 25]. However, these image-based approaches are limited by the location of cameras because of the LOS requirement [26, 27, 28]. Magnetic systems measure changes in the magnetic field to estimate the position and orientation of an object [29, 30, 31], but these can be affected by any metallic material nearby, and thus are easily influenced by electromagnetic interference. Some hybrid systems have been proposed [32, 33, 34] that combine two or more of the above techniques to improve accuracy. In our work, we use G-sensors to capture the motions of the users.

2.3. Energy Conservation

While most of the prior studies focus on energy conservation at the physical layer ([35], [36]), data link layer (PAMAS [37], EAR [38], DBTMA [39], S-MAC [40]), and network layer (LEACH [14], DMCLUSTER [15]), our work employs an energy-efficient cluster-based routing system to reduce energy consumption in data transmission. The basic idea of using cluster-based routing is to choose a CH that will aggregate data from other CNs and then communicate with the remote BS. Given that the CH is generally much closer to CNs than the BS, the transmission power required to send a packet from the CN to the CH will be lower than that needed to send one from the CN to the BS. Therefore, if CNs only communicate with the CH, CNs will consume less energy (as compared to then every CN sends its packets directly to the BS), and the overall energy consumption will be reduced to prolong the lifetime of the sensor nodes and the entire network. Note that, given that CHs will consume more energy than other sensor nodes, in such a cluster-based architecture these need to be periodically re-selected to ensure the energy load is evenly distributed. Otherwise, some nodes which are frequently chosen as the CHs might quickly run out of power quickly, thus rendering the BSN useless.

In LEACH [14], clusters of the sensor nodes are formed based on the received signal strength. Each node in a cluster randomly elects itself as the CH depending on a pre-defined probability, which is based on the desired percentage (P) of CHs required. In $1/P$ rounds, all nodes will become CHs once so that the energy load can be evenly distributed. However, selecting CHs based on probability cannot guarantee that the selection is always optimal since these CHs might not be chosen uniformly over time. PEGASIS [41] is a near-optimal chain-based protocol that presents an improvement of LEACH. In this protocol, each node only communicates with its adjacent node and takes turns becoming the cluster leader that receives at most two packets before transmitting the aggregated data to a BS. This approach

distributes the energy load more evenly among sensor nodes, but also causes a long delay. A hierarchical PEGASIS [42] was thus proposed to overcome this problem. Both PEGASIS and hierarchical PEGASIS operate with the following assumptions, which might be difficult to realized in some cases. First, all sensor nodes are static (i.e. there is no mobility). Second, every node has global knowledge of the network. Bandyopadhyay and Coyle [43] analyzed the optimal parameters p (the optimal probability of becoming a CH) and k (the maximum number of hops allowed from a sensor to its CH) to minimize the energy consumption, and extended the cluster from one level to multiple hierarchies. They focused on the minimization of energy without considering evenly distributing the energy load among nodes. In addition, the computational cost of obtaining p and k might increase the end-to-end latency. HEED [44] is a distributed clustering protocol in which tentative CHs are periodically selected based on the residual energy of sensor nodes. Intra-cluster communication cost, cluster properties (e.g., cluster size) and cluster power levels are considered for selecting the CH. However, HEED cannot guarantee an optimal CH selection, because it relies on secondary parameters to resolve conflicts. Furthermore, selecting temporary CHs could increase the energy consumption.

Generally speaking, all the above techniques aim to save energy by either minimizing the intra-cluster communication or balancing the energy load among nodes, which is also the goal of our work.

3. Motion Detection

In this work, we employ a KNN algorithm [45] to detect the motions of the subject. In this section, we briefly describe the KNN process, which can be divided into the training and detection phases, as follows.

3.1. Training Phase

Each accelerometer sensor has triaxial sensing data, and each axis represents a dimension. We adopt the concept of a Bit-code and Distance based index (BD) [46] to implement the KNN algorithm. In the training phase, the training data of each motion is separately measured, collected and combined together (i.e., $d(d_1, d_2, \dots, d_h)$, h = number of the axes (i.e. 3) \times number of sensor nodes), in the multi-dimensional data space, as shown in Fig. 1. We then decide a reference point $O(o_1, o_2, \dots, o_h)$ to split the data space into $2^{h'}$ partitions represented in bit code (h' is a user-defined positive integer and $h' < h$). Thus, the partition/bit code $S(s_1, s_2, \dots, s_{h'})$ of each set of training data $d(d_1, d_2, \dots, d_h)$ is defined as

$$s_i = \begin{cases} 1, & d_i > o_i, \\ 0, & \text{otherwise,} \end{cases} \quad 1 \leq i \leq h'. \quad (1)$$

For instance, if $h'=2$, the space will be divided into 2^2 partitions, encoded by '00', '01', '10', and '11' respectively, as depicted in Fig. 2(a) [46]. The purpose of using the reference point is to set a central point in the multi-dimensional space to evenly distribute the data density of each partition. For non-uniform data distributions, reference points can be acquired by using some existing techniques, such as K-means, BIRCH or CURE [47]. An efficient KNN search can thus be used by exploiting these partitions. In other words, the KNN algorithm only compares and computes the data in the intersected partitions rather than the entire space.

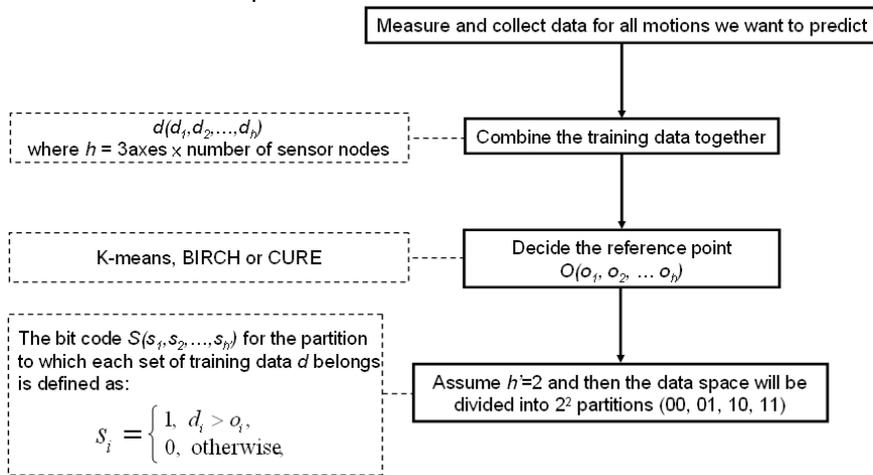


Fig. 1. Flow chart of the training phase

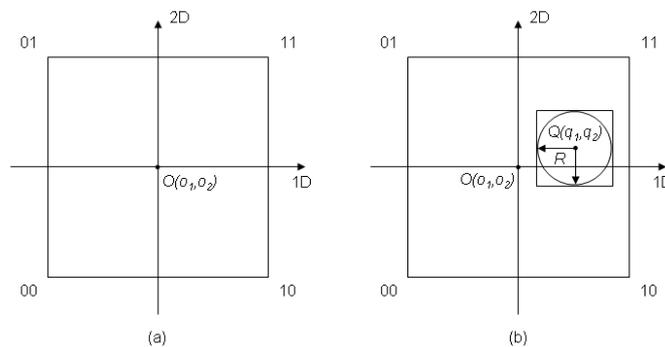


Fig. 2. (a) four-partition data space; (b) in this case, when $|q_1 - o_1| \geq R$, $s_1 = 1$; when $|q_2 - o_2| < R$, $s_2 = 0, 1$ so intersected partitions are (10) and (11)

3.2. KNN Detecting Phase

When the training phase is completed, motions can be detected in the KNN detecting phase, the flow chart of which is shown in Fig. 3. First of all, when the BS receives all sensing data as a query $Q(q_1, q_2, \dots, q_h)$ via a BS, the bit code S of the query Q can be defined as in equation (1), but using q_i instead of d_i . Then, based on the information obtained above, the intersected partitions $T(t_1, t_2, \dots, t_h)$, the number of intersected partitions (≥ 1) are computed by the equation (2) [46], so that only data in the intersected partitions needs to be compared/computed.

$$t_i = \begin{cases} s_i, & |q_i - o_i| \geq R, \\ 0 \text{ and } 1, & \text{otherwise,} \end{cases} \quad 1 \leq i \leq h', \quad (2)$$

where R is the radius for building the desired search field, as the example in Fig. 2(b).

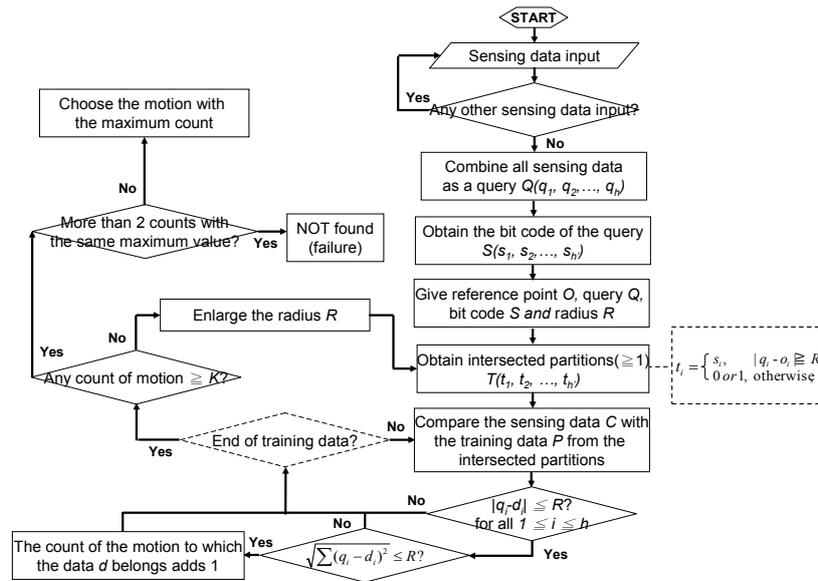


Fig. 3. Flow chart of the KNN detection phase

Next, in order to save computation time by first comparing rather than calculating the square and radical expressions for all data in the intersected partitions, the radius R becomes a pre-filter to filter out the data with any i , $|q_i - d_i| > R$ for $1 \leq i \leq h$, e.g., filtering out the data which is out of the smaller square in Fig. 2(b). Alternatively, one can calculate the Euclidean distance,

$$\sqrt{\sum_{i=1}^n (q_i - d_i)^2} \quad (3)$$

between the training data and the query to check if the sum is equal to or smaller than R , to ensure that the desired data is inside the circle. When that happens, the counter for the classified motion will be increased by one. This process will be repeated until all data in the intersected partitions are computed. KNN will then choose the motion that has the maximum count which is equal or larger than k . Here k is a user-defined threshold. However, if none of the motions has a count that is equal to or larger than k , then the radius R will be increased to enlarge the desired field for searching and the aforementioned process will be repeated. If more than two motions have the same maximum count or the count of every motion is smaller than k when R is already over the upper bound, then the detection is considered as a failure.

4. Energy Conservation

In this section, we first discuss the amount of energy that can be saved using a cluster-based approach. We then propose a novel cluster-head selection algorithm for energy-efficient data transmission. In the context of motion detection, nodes are placed at different places on the body (e.g. arms and legs). During any motion, some nodes might remain at the same positions between two consecutive postures, while others might change their locations. Here we consider two cases. Full-transmission (FT): all nodes periodically send their sensing data directly to the BS, for any motion. Partial-transmission (PT): Only nodes changing their locations will send their data to the CH. We use a threshold-based method to decide whether the nodes have moved from the last sampling time.

4.1. The Benefit of Using a Cluster-based Approach for Motion Capture

In this section, we provide an analysis of the amount of energy that can be saved using a cluster-based approach. We first define some parameters that will be used throughout our analysis. $E_{Tx}(BS)$ is the energy consumption for sending data from a CN to a BS. $E_{Tx}(CH)$ is the energy consumption for sending data from a CN to a CH. E_{Rx} is the energy consumption for receiving a packet. n is total number of sensor nodes and m is the number of static nodes. To simplify our analysis, here we assume that every CN has the same $E_{Tx}(CH)$ and all nodes have the same $E_{Tx}(BS)$ and E_{Rx} . One common practice to save energy is to compress the aggregated data before sending it to the BS [48, 49]. Here we assume that r is the compression ratio.

If all sensor nodes transmit their data directly to the BS without using clusters, the energy consumption E to detect one posture would be:

$$E_1 = E_{Tx}(BS) \times n \times s \quad (4)$$

where s is the size of sensor data.

In a cluster-based approach, a CH receives and integrates CNs' sensing data and then transmits the aggregated data to a BS. Here we consider two situations when using clusters for the case of PT.

Case 1: CH is Moving at The Time of Sampling

$$E_2 = (E_{Tx}(CH) + E_{Rx}) \times (n - 1 - m) \times s + E_{Tx}(BS) \times k \quad (5)$$

Case 2: CH is Static at The Time of Sampling

$$E_2 = (E_{Tx}(CH) + E_{Rx}) \times (n - m) \times s + E_{Tx}(BS) \times k \quad (6)$$

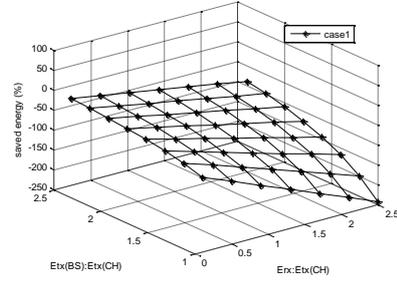
where k is the aggregated packet size. For example, in case 2, $k = (n - m) \times s \times r$.

Here $(n - m)$ nodes have sensing data to be sent out. In case 1 the CH is one of the moving nodes, so it will receive packets from the other $(n - m - 1)$ moving nodes.

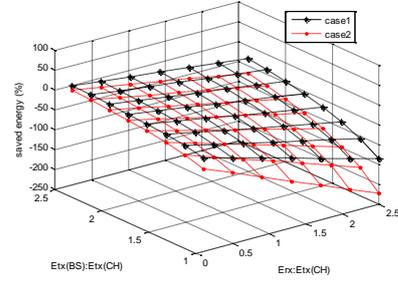
Different kinds of transceivers could have different energy consumption for the transmission and reception of packets [50]. Here we consider the case for different ratios of $E_{Tx}(BS):E_{Tx}(CH)$ from 1.0 to 2.4 (the distance between BS and CH is always longer than that between CH and CNs) and different ratios of $E_{Rx}:E_{Tx}(CH)$ from 0.1 to 2.5. To understand the effect of compression on the aggregated data, we also look at two different compression ratios, $r=0.2$ and 0.4 .

In general, the combination of highest $E_{Tx}(BS):E_{Tx}(CH)$ and lowest $E_{Rx}:E_{Tx}(CH)$ can achieve the best performance in energy saving (i.e. $E_1 - E_2$), since it represents the situation when a cluster-based approach is the most efficient. This can be quickly observed from equations (4), (5) and (6).

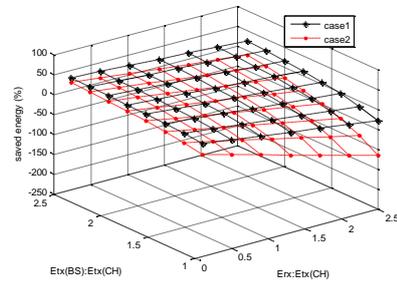
Fig. 4 and Fig. 5 show that the ratio of n and m (i.e. n/m) could strongly affect the amount of energy saved. In general, more energy can be saved when there are more static nodes (i.e. m) in the network. Using clusters does not guarantee that energy can always be saved, especially for the case of low $E_{Tx}(BS):E_{Tx}(CH)$ and high $E_{Rx}:E_{Tx}(CH)$. In addition, the compression ratio r could also affect the results. Based on the above insight, a network protocol designer might need to be careful when employing a cluster-based approach by taking the transceiver characteristics, the motion patterns and the network topology into consideration.



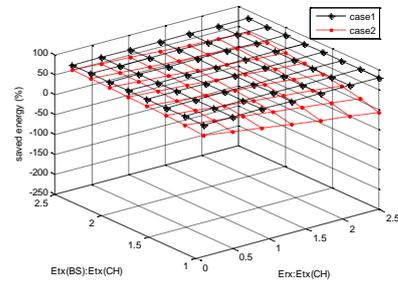
(a) $r=0.2, n=4, m=0$



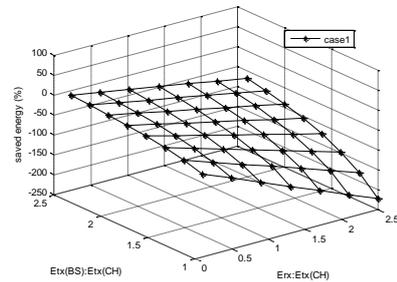
(b) $r=0.2, n=4, m=1$



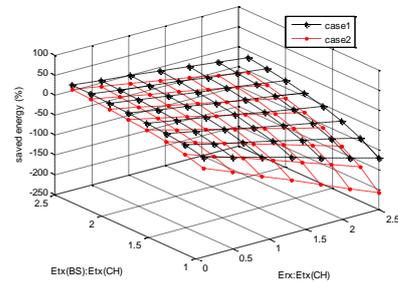
(c) $r=0.2, n=4, m=2$



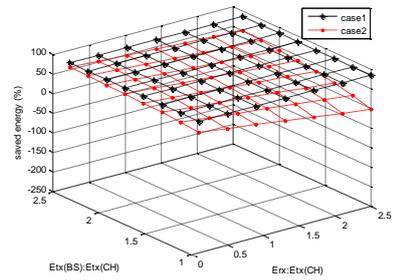
(d) $r=0.2, n=4, m=3$



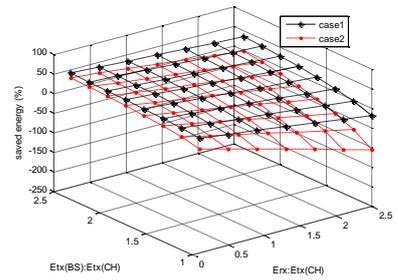
(e) $r=0.4, n=4, m=0$



(f) $r=0.4, n=4, m=1$



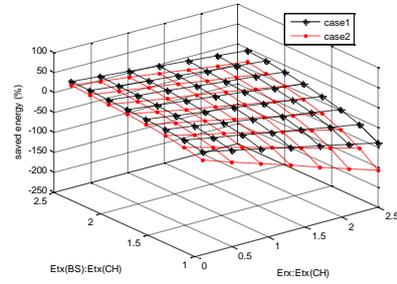
(g) $r=0.4, n=4, m=2$



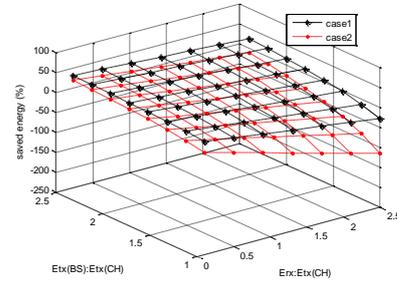
(h) $r=0.4, n=4, m=3$

Fig. 4. The percentage of saved energy when n and r (4 graphs as a set) are fixed

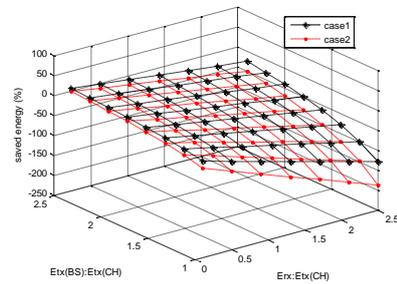
On the Efficiency of Cluster-based Approaches for Motion Detection using Body Sensor Networks



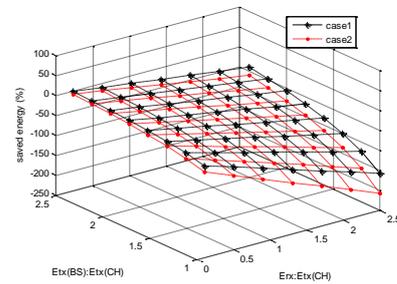
(a) $r=0.2, n=4, m=2$



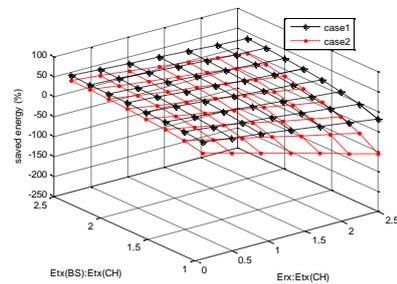
(b) $r=0.2, n=5, m=2$



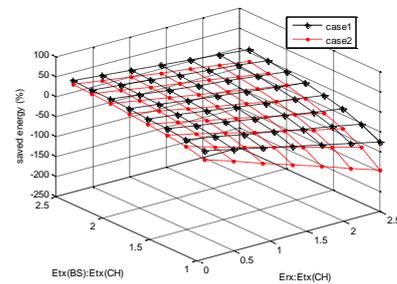
(c) $r=0.2, n=6, m=2$



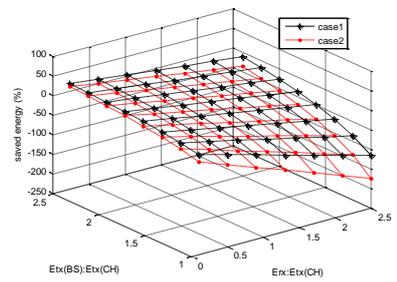
(d) $r=0.2, n=7, m=2$



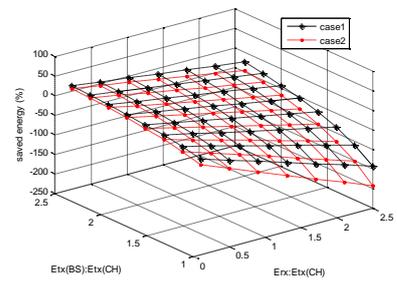
(e) $r=0.4, n=4, m=2$



(f) $r=0.4, n=5, m=2$



(g) $r=0.4, n=6, m=2$



(h) $r=0.4, n=7, m=2$

Fig. 5. The percentage of saved energy when m and r (4 graphs as a set) are fixed

4.2. Cluster Head (CH) Selection

In most previous studies, the CH is typically selected randomly or probabilistically. Some prior works chose the CH based on the nodes' residual energy. In this work, we develop a novel algorithm by minimizing the required energy for intra-cluster communication and preventing certain nodes from serving as the CH too often in order to prolong the lifetime of the entire network.

1) Minimization of Intra-cluster Communication Energy. Here we assume that all postures can be predefined. Therefore, the possible network topologies can be estimated in advance so that the maximum distance between any node in the BSN can be measured. Given that the required transmission power is a function of the distance, we can infer the minimum power level required if the maximum distance is known. Fig. 6 shows the relation between the power levels and node distances based on the measurement collected using Chipcon CC2420 RF transceivers [51]. As shown in Fig. 6, the transmission range using power level 3 is probably sufficient to cover the entire BSN including the BS. Therefore, we first measure all postures with different CHs to collect the total distances between all nodes and the CHs by

$$\sum_{i=1}^n Dis(N_i, CH) , CH \in (N_1, N_2, \dots, N_n) \quad (7)$$

where n is the total number of sensor nodes and $Dis(N_i, CH)$ is the distance between node i and CH. Based on the above insight, we select the CH by choosing the one which has the smallest distance from all CNs as the CH. Therefore, CNs can use the smallest allowable power level when transmitting their data to the CH. In other words, for each posture we find a CH that can minimize intra-cluster energy consumption by

$$\sum_{i=1}^n E_{Tx}(N_i, CH) , CH \in (N_1, N_2, \dots, N_n) \quad (8)$$

where n is the total number of sensor nodes and $E_{Tx}(N_i, CH)$ is the energy consumption of transmitting data from node i to a CH. We assign a unique priority number, ranging from 1 to n (where n is the number of nodes in the BSN), for each node. A smaller number means a higher priority. The node that consumes the least energy in equation (8) will have the highest priority (i.e. priority number=1). Therefore, by assigning the node with the highest priority to be the CH for every posture, we can minimize the intra-cluster energy consumption of the network.

On the Efficiency of Cluster-based Approaches for Motion Detection using Body Sensor Networks

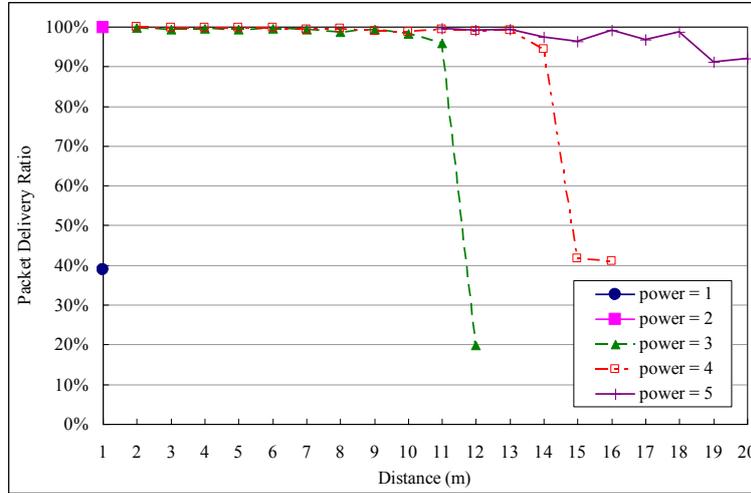


Fig. 6. Power level versus Distance. The range of power level is 0 to 31 for CC2420 RF transceiver

2) Energy Load Distribution. Assume x_i is the number of times node i serves as the CH and y_i is the number of times node i is a CN during the lifetime of a BSN. For a given set of motions we can estimate the maximum amount of energy a node might consume when it is a CH, denoted as $E(CH)$. Similarly, we can also estimate $E(CN)$, which is the maximum amount of energy a node might consume when a node is a CN. To simplify our analysis, here we assume that every node has the same $E(CH)$ and the same $E(CN)$. If the initial energy of each node is I , then the residual energy, $rE(N_i)$, for node i after $(x_i + y_i)$ runs can be computed by

$$rE(N_i) = I - E(CH) \times x_i - E(CN) \times y_i \quad (9)$$

The objective here is to maximize the sum of all x_i for $1 \leq i \leq n$ while keeping $rE(N_i)$ greater than zero.

$$\text{Maximize: } \sum_{i=1}^n x_i \quad (10)$$

$$\begin{cases} rE(N_i) > 0 & , 1 \leq i \leq n \\ s.t. \ y_i \geq \sum_{j=1, j \neq i}^n x_j & , 1 \leq i, j \leq n \\ 0 \leq x_i, y_i & , 1 \leq i \leq n \end{cases} \quad (11)$$

Based on equation (11), the server can compute the threshold x_i that maximizes the objective function for each node i .

When selecting a CH, a node with the highest priority will be considered first. This node will then check if the number of times it has acted as the CH exceeds the assigned threshold. If not, it will be elected as the next CH. Otherwise, the node with the second highest priority will be considered. In addition, the server will recompute new thresholds based on each node's residual energy and broadcast them to each node. This process will be repeated until a node is finally selected as the CH.

In some situations, one node might accidentally be selected as the CH one more time than its assigned threshold, which leads to

$$y_i < \sum_{j=1, j \neq i}^n x_j \quad (12)$$

When this happens, all the other nodes need to readjust their thresholds. Specifically, all x_j (except $j = i$) are proportionally decreased by

$$new \ x_j = y_i \times \frac{old \ x_j}{\sum_{k=1, k \neq i}^n x_k} \quad , 1 \leq j \leq n \text{ and } j \neq i. \quad (13)$$

An example of cluster head selection is shown in Fig. 7. There are four sensor nodes in this scenario. We assume that each round represents a different posture and CH is selected from the node with the highest priority. In this example, N1 is first selected to be the CH in round i and then N3 is selected as the CH in round $i+1$. In every round, when the server receives the aggregated data from the CH, it also records the number of times each node has acted as the CH. Once the number of times of that a node has acted as the CH exceeds its assigned threshold (in Figure 7, N3 exceeds its threshold in round $i+k$ so that N2, the node with the second highest priority, is chosen as the CH instead), server will compute, according to equation (11), new thresholds for all sensor nodes based on the residual energy of each node, and then transmits the re-computed thresholds to all nodes.

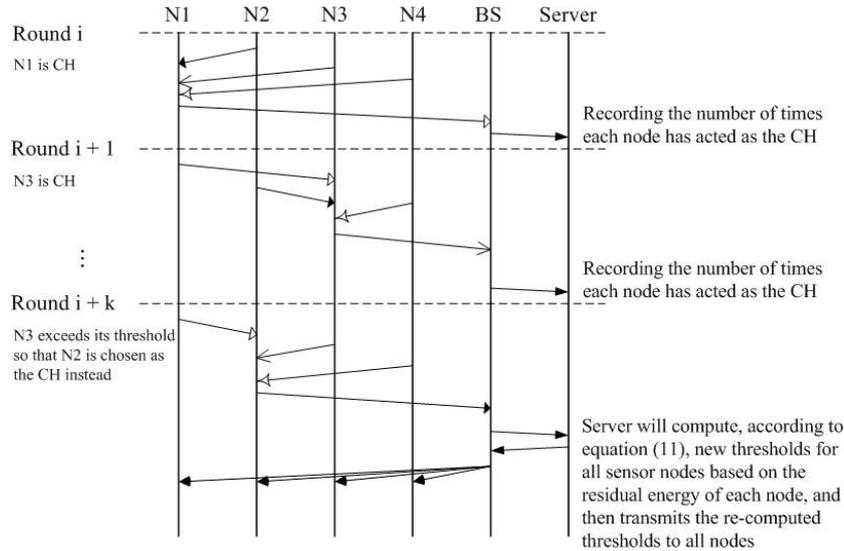


Fig. 7. Example of energy load distribution

5. Implementation

To evaluate the performance of our protocol, we compare it with the case of PT, as in Section 4, (in which all nodes use the same power level, one that is sufficient to reach the entire BSN). We look at the real packet format in TinyOS in which sensing data is a small part (6 Bytes) of the whole packet (26 Bytes) and thus conservatively choose a compression ratio $r=0.5$ in our analysis. In addition, in our analysis we model the energy consumption for transmission and reception based on the specifications of a CC2420 transceiver. As shown in Fig. 8, our protocol can achieve more energy saving when the ratio of m/n increases. Here n is the number of nodes in the network and m is the number of static (non-moving) nodes. For a 10-node network, our protocol can achieve up to 90% less energy consumption as compared to PT. Here

the saved energy is defined as $(E_{PT} - E_{our\ protocol}) / E_{PT}$. In other words, in the best case, our protocol can prolong the lifetime of the network to be ten times longer as compared to PT.

We used KNN algorithm for the motion detection and implemented a testbed using the TelosB mote [52] which has a TI MSP430 [53] processor and a Chipcon CC2420 RF transceiver [51] that supports the IEEE 802.15.4 for communication. A pair of AA batteries are used for power supply. The data rate is 250 kbps when operating at the frequency of 2.4 GHz. We employed four triaxial MEMS accelerometer sensors placed on the wrists and ankles of

a person to capture the subject's motions. The BS receives sensing data from the BSN and forwards it to a PC server where KNN is executed. We developed our program under TinyOS [54], which is a component-based and embedded operating system written in the nesC programming language.

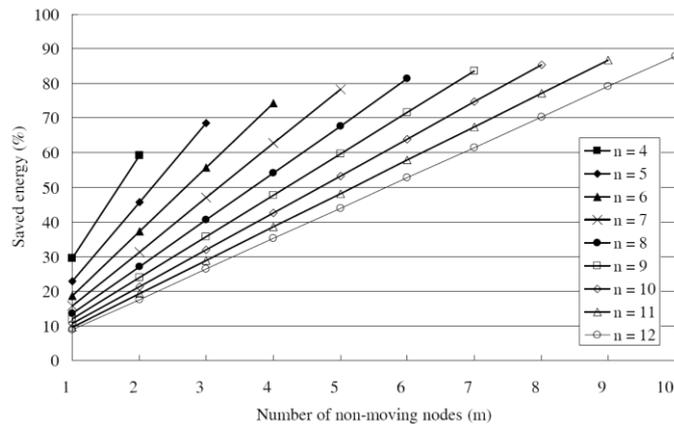


Fig. 8. Comparison between PT and our protocol for saving energy

To evaluate the accuracy of our cluster-based approach and compare it with the FT case, we used five pre-defined motions from boxing, namely Right Straight Punch (RSP), Right Hook Punch (RHP), Right Uppercut Punch (RUP), Right Front Kick (RFK), and Right Side Kick (RSK) in our experiments.

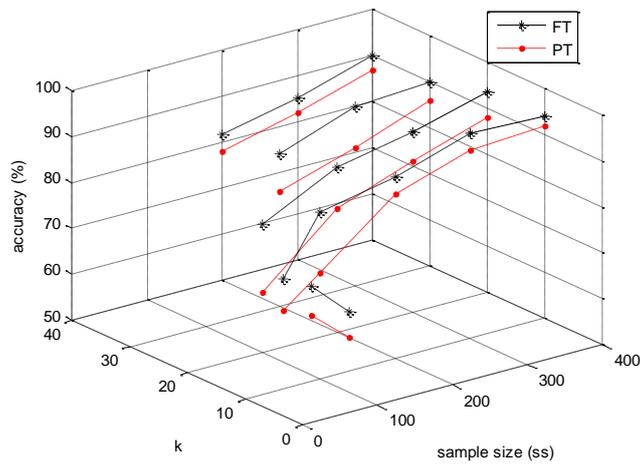


Fig. 9. Detecting accuracy based on varying combinations of ss and k

Our experiments were run in real time. In Fig. 9, each point represents the results from 50 repetitions of each motion (250 times in total). Sample size (ss) and k are important parameters for the KNN algorithm, so both of them are dynamically tuned to find the highest accuracy. Generally, when k is fixed, the accuracy will go up with the increase in sample size, but the reverse is not true. There is a tradeoff between (ss , k) and the accuracy, since we need to consider the computation cost/time depending on the characteristics of the applications. The results indicate our cluster-based algorithm (known as PT) can achieve similar performance as that of FT. When $ss=300$ and $k=10$, the accuracy is 94.8% for FT and 91% for PT respectively. The performance is slightly better when we increase the sample size to 400. The accuracies for FT and PT are 94% and 92%, respectively. Generally, the accuracy of FT is higher than that of PT, since it provides more information to the KNN.

6. Conclusion

In this paper, we provide a detailed analysis of how different parameters in a cluster-based algorithm can affect the amount of energy saving. Additionally, we propose a novel CH selection algorithm by considering the required power level for sending a packet from the CN to the CH. Furthermore, we use a threshold-based approach to evenly distribute the energy load among nodes by considering the number of times a node has acted as a CN and a CH. We implement our cluster-based algorithm on a testbed using the KNN algorithm and show that it can achieve high accuracy (above 90%). We are currently working on a prediction mechanism to reduce the number of transmissions required when motions of CNs can be predicted.

References

1. IEEE 802.15 WPAN™ Task Group 4 (TG4). [Online]. Available: <http://www.ieee802.org/15/pub/TG4.html> (current August 2011)
2. Xia, F., Xu, Z., Yao, L., Sun, W., Li, M.: Prediction-based Data Transmission for Energy Conservation. In Proceedings of the 5th Annual International Conference on Wireless Internet, Singapore. (2010)
3. Xiao, S., Dhamdhere, A., Sivaraman, V., Burdett, A.: Transmission Power Control in Body Area Sensor Networks for Healthcare Monitoring. IEEE Journal on Selected Areas in Communications, Vol. 27, No. 1, 37-48. (2009)
4. Park, S., Jayaraman, S.: On Innovation: Quality of Life and Technology of BodyNets. In Proceedings of the 3rd International Conference on Body Area Networks, Tempe, Arizona, USA. (2008)
5. Otal, B., Alonso, L., Verikoukis, C.: Novel QoS Scheduling and Energy-saving MAC Protocol for Body Sensor Networks Optimization. In Proceedings of the 3rd International Conference on Body Area Networks, Tempe, Arizona, USA. (2008)

6. Lorincz, K., Chen, B.-R., Challen, W. G., Chowdhury, A. R., Patel, S., Bonato, P., Welsh, M.: Mercury: A Wearable Sensor Network Platform for High-fidelity Motion Analysis. In Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, Berkeley, California. (2009)
7. Biswas, S., Quwaider, M.: Body Posture Identification using Hidden Markov Model with Wearable Sensor Networks. In Proceedings of the 3rd International Conference on Body Area Networks, Tempe, Arizona, USA. (2008)
8. Ghasemzadeh, H., Jafari, R.: Sport Training using Body Sensor Networks: A Statistical Approach to Measure Wrist Rotation for Golf Swing. In Proceedings of the 4th International Conference on Body Area Networks, Los Angeles, California, USA. (2009)
9. Kwon, D. Y., Gross, M.: Combining Body Sensors and Visual Sensors for Motion Training. In Proceedings of the ACM SIGCHI International Conference on Advances in computer Entertainment Technology, Valencia, Spain, 94-101. (2005)
10. Vlasic, D., Adelsberger, R., Vannucci, G., Barnwell, J., Gross, M., Matusik, W., Popovic, J.: Practical Motion Capture in Everyday Surroundings. In Proceedings of the 34th International Conference on Computer Graphics and Interactive Techniques, San Diego, California, USA. (2007)
11. Lamprinos, I. E., Prentza, A., Sakka, E., Koutsouris, D.: Energy-efficient MAC Protocol for Patient Personal Area Networks. IEEE Engineering in Medicine and Biology Society, 3799-3802. (2005)
12. Omeni, O. C., Eljamaly, O., Burdett, A. J.: Energy Efficient Medium Access Protocol for Wireless Medical Body Area Sensor Networks. In Proceedings of the 4th IEEE/EMBS International Summer School and Symposium on Medical Devices and Biosensors, 29-32. (2007)
13. Abbasi, A.A., Younis, M.: A Survey on Clustering Algorithms for Wireless Sensor Networks. Elsevier Computer Communications, Vol. 30, 2826-2841. (2007)
14. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient Communication Protocol for Wireless Microsensor Networks. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Hawaii, USA, 1-10. (2000)
15. Ci, S., Guizani, M., Sharif, H.: Adaptive Clustering in Wireless Sensor Networks by Mining Sensor Energy Data. Computer Communications, Vol. 30, 2968-2975. (2007)
16. Yan, L., Zhong, L., Jha, N.: Energy Comparison and Optimization of Wireless Body-area Network Technologies. In Proceedings of the Second International Conference on Body Area Networks, Florence, Italy. (2007)
17. Foxlin, E., Harrington, M., Pfeifer, G.: Constellation: A Wide-Range Wireless Motion-Tracking System for Augmented Reality and Virtual Set Applications. In Proceeding of ACM SIGGRAPH 1998, New York, 371-378. (1998)
18. Foxlin, E., Harrington, M.: Wear Track: A Self-referenced Head and Hand Tracker for Wearable Computers and Portable VR. In Proceedings of the 4th International Symposium on Wearable Computers, Atlanta, Georgia, USA. (2000)
19. Smith, A., Balakrishnan, H., Goraczko, M., Priyantha, N.: Tracking Moving Devices with the Cricket Location System. In Proceedings of the Second International Conference on Mobile Systems, Applications, and Services, Boston, MA, 190-202. (2004)
20. Bussman, J., Martens, W., Tulen, J., Schasfoort, F., Van Den Bergemons, H., Stam, H.: Measuring Daily Behavior using Ambulatory Accelerometry: The

On the Efficiency of Cluster-based Approaches for Motion Detection using Body Sensor Networks

- Activity Monitor. *Behavior Research Methods, Instruments & Computers*, 349-356. (2001)
21. Mayagoitia, R., Nene, A., Veltink, P.: Accelerometer and Rate Gyroscope Measurement of Kinematics: An Inexpensive Alternative to Optical Motion Analysis Systems. *Journal of Biomechanics*, Vol. 35, 537-542. (2002)
 22. Sakaguchi, T., Tsutomu, K., Katayose, H., Sato, K., Inokuchi, S.: Human Motion Capture by Integrating Gyroscopes and Accelerometers. In *Proceedings of the 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Washington, DC, USA, 470-475. (1996)
 23. KIRK, A. G., OBRIEN, J. F., FORSYTH, D. A.: Skeletal Parameter Estimation from Optical Motion Capture Data. In *Proceeding of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 782-788. (2005)
 24. Kurihara, K., Hoshino, S., Yamane, K., Nakamura, Y.: Optical Motion Capture System with Pan-tilt Camera Tracking and Tealtime Data Processing. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Vol. 2, Washington, DC, USA, 1241-1248. (2002)
 25. Dorfmueller-Ulhaas, Klaus: Robust Optical User Motion Tracking using a Kalman Filter. In *Proceedings of the 10th ACM Symposium on Virtual Reality Software and Technology*, Osaka, Japan. (2003).
 26. Chen, Y., Lee, J., Parent, R., Machiraju, R.: Markerless Monocular Motion Capture using Image Features and Physical Constraints. In *Proceedings of the Computer Graphics International*, New York, USA, 36-43. (2005)
 27. Sidenbladh, H., Black, M. J., Fleet, D. J.: Stochastic Tracking of 3D Human Figures using 2D Image Motion. In *Proceedings of the 6th European Conference on Computer Vision*, Dublin, Ireland, 702-718. (2000)
 28. Uchinoumi, M.; Joo Kooi Tan; Ishikawa, S.: A Simple-Structured Real-Time Motion Capture System Employing Silhouette Images. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, The Hague, Netherlands, 10-13. (2004)
 29. O'brien, J., Bodenheimer, R., Brostow, G., Hodgins, J.: Automatic Joint Parameter Estimation from Magnetic Motion Capture Data. In *Proceedings of the Graphics Interface*, Québec, Canada, 53-60. (2000)
 30. Yu Su, Allen, C.R., Geng, D., Burn, D., Brechany, U., Bell, G.D., Rowland, R.: 3-D Motion System (Data-Gloves): Application for Parkinson's Disease. *IEEE Transactions on Instrumentation and Measurement*, Vol. 52, 662-674. (2003)
 31. Hoshino, K.: CGA Synthesizer Interpolating and Extrapolating Motion Data. In *Proceedings of the IFIP First International Workshop on Entertainment Computing*, Makuhari, Japan, 339-346. (2002)
 32. Roetenberg, D., Luinge, H.J., Veltink, P.H.: Inertial and Magnetic Sensing of Human Movement near Ferromagnetic Materials. In *Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality*, Tokyo, Japan, 268-269. (2003)
 33. Nguyen, K. D., Chen, I-M., Yeo, S. H., Duh, B.-L.: Motion Control of a Robotic Puppet through a Hybrid Motion Capture Device. In *Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering* Scottsdale, AZ, USA. (2007)
 34. Klein, G., Drummond, T.: Tightly Integrated Sensor Fusion for Robust Visual Tracking. *Image and Vision Computing*, Vol. 22, 769-776. (2004)

35. Kawadia, V., Kumar, P.R.: Power Control and Clustering in Ad Hoc Networks. In Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, San Francisco, USA. (2003)
36. Narayanaswamy, S., Kawadia, V., Sreenivas, R.S., Kumar, P.R.: Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW protocol. In Proceedings of the European Wireless 2002. Next Generation Wireless Networks: Technologies, Protocols, Services and Applications, Florence, Italy, 156-162. (2002)
37. Singh, S., Raghavendra, C.: PAMAS: Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks. ACM Computer Communication Review, Vol. 28, No. 3, 5-26. (1998)
38. Sohrabi, K., Gao, J., Ailawadhi, V., Pottie, G. J.: Protocols for Self-Organization of a Wireless Sensor Network, IEEE Personal Communications Magazine, Vol. 7, No. 5, 16-27. (2000)
39. Haas, Z., Tabrizi, S.: On Some Challenges and Design Choices in Ad-Hoc Communications. In Proceedings of the IEEE Military Communications Conference, Boston. (1998)
40. Ye, W., Heidenmann, J., Estrin, D.: An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies, New York, NY. (2002)
41. Lindsey, S., Raghavendra, C. S.: PEGASIS: Power Efficient GATHERing in Sensor Information Systems. In Proceedings of the IEEE Aerospace Conference, Big Sky, Montana. (2002)
42. Lindsey, S., Raghavendra, C. S., Sivalingam, K.: Data Gathering in Sensor Networks using the Energy*Delay Metric. In Proceedings of the 2001 International Parallel and Distributed Processing Symposium, Workshop on Issues in Wireless Networks and Mobile Computing, San Francisco, CA. (2001)
43. Bandyopadhyay, S., Coyle, E.: An Energy-Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks. In Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, San Francisco, USA. (2003)
44. Younis, O., Fahmy, S.: HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks. In Proceedings of the 23rd Conference of the IEEE Communications Society, Hong Kong. (2004)
45. Manning C. D., Schütze H.: Foundations of Statistical Natural Language Processing. Cambridge, MIT Press. (1999)
46. Liang, J. J., Feng, Y. C.: Indexing the Bit-code and Distance for Fast KNN Search In High-dimensional Spaces. Zhejiang University: Science A, Vol. 8, 857-863. (2007)
47. Guha, S., Rastogi, R., Shim, K.: Cure: An Efficient Clustering Algorithm for Large Databases. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Seattle, Washington, USA, 73-84. (1998)
48. Krishnamachari, B., Estrin, D, Wicker, S.: The Impact of Data Aggregation in Wireless Sensor Networks. In Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops, Vienna, Austria. (2002)
49. Maddes, R., Szewczyk, M., Franklin, J., Culler, D.: Supporting Aggregate Queries over Ad-hoc Wireless Sensor Network. In Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications, Callicoon, NY, USA. (2002)
50. Shnayder, V., Hempstead, M., Chen, B.-R., Werner-Allen, G., Welsh, M.: Simulating the Power Consumption of Large-scale Sensor Network Applications.

On the Efficiency of Cluster-based Approaches for Motion Detection using Body Sensor Networks

In Proceedings of the Second ACM Conference on Embedded Networked Systems, Baltimore, Maryland, USA. (2004)

51. CC2420 Datasheet, <http://focus.ti.com/docs/prod/folders/print/cc2420.html>
52. TPR2400 Datasheet. [Online]. Available: http://www.willow.co.uk/html/telosb_mote_platform.html (current August 2011)
53. Microcontrollers (MCU). [Online]. Available: <http://focus.ti.com/mcu/docs/mcuhome.tsp?sectionId=101> (current August 2011)
54. Tinyos. [Online]. Available: <http://www.tinyos.net/> (current August 2011)

Kun-chan Lan received his Ph.D degree from Computer Science at University of Southern California in 2004, advised by Professor John Heidemann. From 2004 to 2007, Kun-chan joined the Network and Pervasive Computing Program at NICTA (National ICT Australia) in Sydney as a researcher. He is currently an assistant professor at the Department of Computer Science and Information Engineering of National Cheng Kung University.

Chine-Ming Chou received his B.S. degree from Computer Science at Vanung University of technology in 2005 and M.S. degree from Information and Communication Engineering at Chaoyang University of technology in 2007. He is currently working for his Ph.D degree in Department of Computer Science and Information Engineering, National Cheng Kung University. His research interests include Network Simulation, Network measurement, Opportunistic routing, WiMAX and DSRC protocol, and mobility model.

Tzu Nung Wang received his BS (Industrial Engineering and Management) from National Taipei University of Technology and MS (Medical Informatics) from National Cheng Keng University in Taiwan in 2003 and 2010 respectively. His research interest is related to wireless sensor networks especially for energy saving issue. Now he works as an engineer of software development in a LCD company.

Mei-Wen Li received her M.S degrees from Chaoyang University of Technology in Taiwan in 2006. Her research interests encompass issues relating to Cellular Network, Vehicular network and Cognitive network. Currently, she works an camera development company as an engineer.

Received: March 10, 2011; Accepted: May 4, 2011.

Optimization of Multiple Gateway Deployment for Underwater Acoustic Sensor Networks

Jugen Nie^{1,2}, Deshi Li¹, and Yanyan Han¹

¹ School of Electronic Information, Wuhan University
430079 Wuhan, China
dshi@whu.edu.cn

² School of Electronic and Information Engineering, Jinggangshan University
343009 Ji'an, China
{niejugen, hanyan4981}@163.com

Abstract. This research aims to develop novel technologies to efficiently integrate wireless communication networks and Underwater Acoustic Sensor Networks (UASNs). Surface gateway deployment is one of the key techniques for connecting two networks with different channels. In this work, we propose an optimization method based on the genetic algorithm for surface gateway deployment, design a novel transmission mechanism—simultaneous transmission, and realize two efficient routing algorithms that achieve minimal delay and payload balance among sensor nodes. We further develop an analytic model to study the delay, energy consumption and packet loss ratio of the network. Our simulation results verify the effectiveness of the model, and demonstrate that the technique of multiple gateway deployment and the mechanism of simultaneous transmission can effectively reduce network delay, energy consumption and packet loss rate.

Keywords: Underwater acoustic sensor networks, gateway deployment, transport mechanism.

1. Introduction

In recent years, UASNs (Underwater Acoustic Sensor Networks) have drawn broad attentions in scientific research, social services, and military applications. UASNs can be widely applied in underwater target tracking and environment monitoring, etc [1][2][3]. Compared with remote sensing methods, UASNs can provide real time in-situ information for shallow-sea and deep-sea monitoring. However, unlike terrestrial wireless sensor networks, UASNs could not use electromagnetic wave as communication media due to its quick absorption in water. Therefore, acoustic signal is often used as a transport carrier in UASNs [4]. Underwater acoustic channel is a complex random channel with variable time-space-frequency parameters, low carrier frequency, long transmission delay, narrowband, strong noise and

multipath interference, and many other transmission attenuation factors. Therefore it becomes a highly challenging wireless communication channel. The propagation speed of underwater acoustic signal (approximately 1500m/s, with variation due to minor changes of pressure, temperature, and salinity of water) is lower than electromagnetic wave propagation speed by five orders of magnitude; such high propagation delay will not only limit the interactive application, but also prolong the response time of communication.

Compared to its single gateway counterpart, data in a underwater sensor network with multiple gateways (Fig. 1) do not have to be transmitted via a long path to a fixed surface gateway, but via a path selected in light of optimized network performance (such as minimal delay, minimal energy consumption and least packet loss rate) to one of the available gateways [5]. Depends on the service requirement, the surface gateways can adopt various wireless communication channels such as Cellular network, Zigbee and so on. Note that the data propagation delay from surface gateways to the base station is much shorter than that in underwater acoustic network. The bandwidth of wireless channel can vary from tens of million to hundreds of million bits per second; in a sharp contrast, the data rate of underwater acoustic channel is only between several hundred bits and ten kilobits per second. Moreover, the packet loss rate of surface gateways is much smaller than that of underwater acoustic network. Also compared to radio communication, acoustic communication needs much more energy [6]. The surface gateways can obtain power supply from solar panels or by charging battery periodically. Due to the aforementioned factors, an underwater sensor node can select a path to one of these gateways, aiming to minimize the delay, energy consumption or packet loss rate from it to the gateway. Therefore, how to design an efficient routing algorithm according to different optimization goals, how to select the number and the position of surface gateways are the key research issues of the UASNs with multiple gateways.

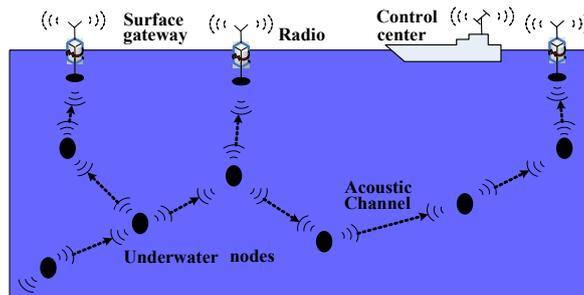


Fig. 1. Structure of Multiple Gateway UASNs

In our previous work [5], we have studied the surface gateway positioning problem based on GA in UASNs and formulated it as an optimization problem. We have shown that using multiple surface gateways can effectively reduce network energy consumptions and delay. However, the

routing of data packets from underwater sensor nodes to surface gateways under various objects was not studied.

In this work, we assume that the coordinates of underwater sensor nodes are known. We show how routing is discovered under various objectives such as minimal delay and payload balance, and design a novel transmission mechanism—simultaneous transmission. We further develop an analytic model to study the delay, energy consumption and packet loss ratio of the network. As a result, we verify the effectiveness of the model.

The rest of the paper is organized as follows. Sec. 2 illustrates the related works of multiple gateway deployment. Sec. 3 presents the model for optimization of multiple gateway deployment. Sec. 4 proposes a new mechanism of simultaneous transmissions. Sec.5 describes the process of route discovery. Sec.6 describes gateway deployment based on GA. Sec. 7 discusses simulation results. Finally, Sec. 8 concludes the paper.

2. Related Works

Deployment of multiple gateway in wireless networks can be divided into two categories: one is terrestrial wireless networks, and the other is multiple surface gateway deployment in UASNs.

Most multiple gateway deployment research in terrestrial wireless networks is mainly on the gateway site selection in wireless mesh networks (WMNs). Literature [7] proposed a new routing algorithm called two concurrent path routing (2CPR) for wireless mesh networks with multiple gateways. Literature [8] proposed a gateway selection scheme that considers multiple end-to-end QoS parameters. Literature [9] presented a multiple gateway selection algorithm for WMNs called Weighted Clustering Algorithm (WCA), which is an optimization mechanism based on the Quality of Service (QoS), and used a heuristic algorithm to improve the performance of WCA. The core of WCA is determining the number and location of gateways. However, WCA assumes that nodes locations are known, so that some nodes can be selected as gateways for cluster planning. Aforementioned research focuses on the selection of gateway. For the number of gateways has been pre-determined, the algorithm aimed at selecting the gateway with balanced payload in the network, but did not refer to the gateway position and the effect of gateway number. Literature [10] combined graph partition and spanning tree algorithm, but how to get good computational efficiency of this algorithm is one problem.

In the aspect of network structure, terrain wireless network is similar to a 2-dimensional structure, while underwater acoustic network is a typical 3-dimension structure. In the research of multiple gateways deployment, more emphases have been put on the selection of gateways in mesh network and cluster heads in Wireless Sensor Networks (WSNs). Their general goal is to get balanced payload and optimized QoS. For to the large delay, low bandwidth and limited energy supply in UASNs, the deployment of surface

gateways should be focused on the propagation delay from underwater sensor nodes to gateways, energy consumption of underwater nodes and packet loss rate.

For multiple gateway deployment in UASNs, Literature [11] gave a triangular deployment method of 2-dimensional network, and it was aimed at target area coverage with minimal sensors. Literature [12] studied the coverage and routing in 3-dimensional networks. Neither of the two studies involved network structure with multiple sinks nodes (or gateways). Literature [13] used multiple sink nodes to improve the performance of UASNs, but they did not analyze the influence of network delay and energy consumption with multiple gateways. Literature [14] proposed the architecture of surface gateway that has several wireless communication channels and communication protocols. However, they didn't study the effect of the gateway number and positioning. Literature [6][15] focused on how many gateways should be used, and then derived the positions of multiple gateway from the fixed grid on the surface. The queuing delay due to MAC protocol is not considered, the deployment of surface gateways is not resolved with heavy network payload, and the preliminary simulation results were obtained only for one layer (which is 100 meters below the surface). For most of the applications, underwater sensor network would be a 3-dimensional network, because of limited available bandwidth and low signal propagation speed of underwater acoustic channel, data packet collisions, avoidance and reservation in MAC layer protocol will inevitably bring about queuing delay. At the same time, different transport mechanisms will also affect data transmission delay and energy consumption in network, these problems should be considered for 3-dimension UASNs.

To solve the above issues, this paper proposes a new method of surface gateway deployment. Its main idea is to take the fixed underwater nodes as source nodes and the movable surface gateways as destination nodes. We design a novel transmission mechanism—simultaneous transmissions by nodes which are separated by three hops and realize two efficient routing algorithms that achieve minimal delay and payload balance among sensor nodes.

With the delay, energy consumption and packet loss ratio of network as the optimization objectives, we can obtain the optimal positions of surface gateways and a routing that can optimize network performances.

3. Optimization Model

Surface gateway deployment can be treated as an optimization problem. In this section, we discuss the definitions, constraints and objective functions for gateway deployment optimization model.

3.1. Definitions

3.1.1 Nodes

Here let V denote all the underwater sensor nodes, G be all the surface gateways, V' denote all nodes, that is to say $V'=V\cup G$. $I(v)$ denotes all nodes in the communication range of node v , i.e., $I(v)=\{w:w\in V', v\neq w, d(v,w)\leq R\}$, where $d(v,w)$ is the distance between v and w , and R is the communication range of sensor nodes.

3.1.2 Edges

Let E be all the edges $e=(v,w)$, $v\in V, w\in I(v)$, and $e(v,u)=\{e(v,u):(v,u) v\in E, u\in I(v)\}$ denote the direction from node v to node u . For any surface gateway node, because the data relaying rate of surface gateways is much higher than underwater nodes, so it doesn't need to consider the relay delay in UASNs. Therefore we can mainly focus on the receiving data for gateways.

3.1.3 Gateway Position

$G(X, Y)$ denotes all surface gateway positions; $G_i(x_i, y_i)$ is the position of the i th gateway.

3.1.4 Queuing Delay of MAC

T_{mac} is the delay caused by MAC layer, which is associated with the connectivity degree of a node. For simplicity, let t_m be the queuing delay of one unit. The queuing delay with only one neighbor is set to $T_{mac}=0$, then the queuing delay caused by two one-hop neighbors is $T_{mac}=t_m$, three neighbor links lead to the queuing delay $T_{mac}=2t_m$; and so on. For a node with n neighbors, its queuing delay is approximated as $(n-1)t_m$.

3.2. Constraints

In current UASNs, half-duplex communication mode is often adopted. Therefore we set a simple conflict model: when a node is receiving data, it could not send data simultaneously.

3.3. Optimization Variables

We set two optimization variables: the number of surface gateways N and the position of surface gateways $G(X, Y)$. Because gateways are located on the surface, the Z-axis coordinate is similar to a fixed value, so the position variables are simplified as X-axis coordinate and Y-axis coordinate. Our goal

is to optimize N and $G_i(x_i, y_i)$ according to the objective functions introduced below.

3.4. Optimization Function

3.4.1 Minimum Delay

The goal is to minimize the total delay of all packets that reach the surface gateway. Packet delay is the sum of all delays generated from every hop along the path from the transmitter to the receiving gateway. Combining queuing delays of the relay-nodes (resulted from the MAC layer or routing layer), delay of each hop composites of transmission delay, queuing delay and propagation delay, the delay t of data on edge e can be written as

$$t(e) = t_{mac}(e) + t_s(e) + t_p(e) = n \times t_m + \frac{L_e}{C} + \frac{d(e)}{v_p} \quad (1)$$

Here, $t_{mac}(e)$ represents queuing delay, $t_s(e)$ represents transmission delay and $t_p(e)$ denotes propagation delay, n is the connectivity degree of a receiving node (i.e., the number of neighbor nodes in the spanning tree), t_m is the connectivity degree associated with unit queuing delays, L_e is a total length of transmitted data packet in unit of *bit*, including all data generated by the sending node and data to be relayed by the next hop, C is the channel capacity in unit of *bit/sec*, $d(e)$ is the edge length, v_p is the underwater acoustic propagation speed.

Therefore, the objective function of minimum total delay is

$$\text{Minimize} \left(\sum_{e \in E_r} t(e) \right) \quad (2)$$

Where E_r stands for all edges of the route spanning tree from the underwater nodes to gateway. Obviously, $e \in E_r$.

3.4.2 Minimum Energy Consumption

The goal is to achieve the minimum energy consumption when all the packets have been transmitted to the surface gateway. If a node sends data L_v , the energy it consumes can be written as

$$\varepsilon(v) = P_s(v)t_s + P_r(v)t_r = (P_s(v) + P_r(v)) \frac{L_v}{C} \quad (3)$$

Here, $P_s(v)$ is the transmitting power of node v , $P_r(v)$ is the receiving power of node v . t_s and t_r denote transmission time and reception time respectively. C is the channel capacity in unit of *bit/sec*.

Corresponding objective function as:

$$\text{Minimize} \left(\sum_{v \in V} \varepsilon(v) \right) \quad (4)$$

3.4.3 Minimum Packets Loss Rate

Packet loss may be resulted from the physical layer, MAC layer, network congestion and other factors. For example, when the data generation period of a node is less than the minimum period required for network transmission, the data must be stored in the memory of a relay-node, and when the memory is full, the data will overflow possibly leading to packet loss. Here considering packet loss generated when traffic constraint is not satisfied, the optimization objective is to minimize packet loss rate of all packets reaching the surface gateway. The packet loss rate of each node can be expressed as:

$$\beta(v) = \frac{f_v - C}{C} \quad (5)$$

Here f_v represents the flow of the node. When $f_v > C$, it indicates that there is packet loss; when $f_v \leq C$, the above formula is zero or negative, and thus there is no packet loss.

The packet loss rate of the whole network is

$$\beta = \frac{\sum_{v \in V} (L_v - C)}{m \times C}, \quad (6)$$

where m is the total number of underwater nodes.

The corresponding objective function is:

$$\text{Minimize} (\beta) \quad (7)$$

4. Data Transmission Mechanism

The state-of-the-art UASNs protocols fall in to two categories: one is based on the mechanism of competition, and the other is based on the mechanism of time sequence [16]. In order to reduce the delay and energy consumption caused by packet collision and retransmission, we adopt the TDMA scheme of MAC protocol in this research. As the positions of underwater sensor nodes are known, the positions of surface gateways and the route from underwater nodes to gateways can be obtained according to the optimization objective functions (i.e., minimal delay, minimal energy consumption or least packet loss rate). We make use of the initialization method in literature [17], allocate time slots to nodes, and at the same time, send the information of synchronization and routing to underwater nodes by surface gateways. Then the data of underwater nodes will be transmitted according to the routes and time slots periodically.

Usually the data transmission from underwater sensor nodes to surface gateways is a directional data flow (from the last hop in the bottom to

previous hop until the surface gateway is reached), and the transmission can be based on the Shortest path transport mechanism, Accumulated transmission hop by hop, or Mechanism of simultaneous transmission every three hops.

4.1. Shortest Path Transport Mechanism

With the shortest path transport mechanism, data generated by each node will be transferred to the surface gateway nodes via the shortest path.

As shown in Figure 2(a), each node chooses the shortest gateway from it as the destination, uses the shortest path to reach the gateway. The total delay is the sum of delays of all data packets reaching gateway. The dotted lines indicate the connections, and the solid lines represent the actual transmission paths (numbers beside lines denote delay). Figure 2(b) represents the time sequence. By this method, the time slot of every node could be selected arbitrarily.

When UASNs is small, low delay could be achieved with the shortest path transport mechanism, because the primary delay here is the transmission delay.

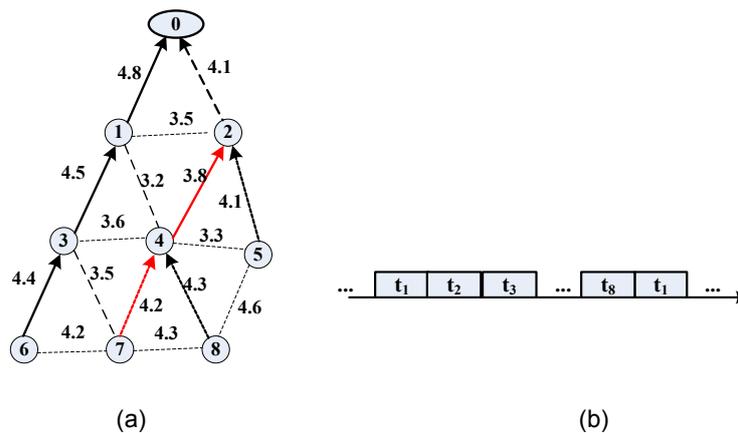


Fig. 2. Shortest Path Transport Mechanism

4.2. Accumulated Transmission Hop-by-Hop

This transport mechanism is based on the following idea. From the node of the last hop along all other nodes in the path, data is accumulated and relayed hop by hop, and finally arrives at the gateway. This transport mechanism can reduce repeated path propagation delay by nodes of

previous hops. The last hop node initiates data transmitting, and accumulates the data ahead hop by hop. The total delay constitutes of transmission delay and propagation delay on the edge with minimum distance as routing edge. In comparison with Figure 2(a), it is obvious that the resulting routing paths are different, as node 4 and node 7 in the Figure 3(a) (numbers beside lines denotes delay). Nodes 7 and 4 select nodes 1 and 3 as the next hop node, respectively, which have smaller delay among the previous hop nodes. Figure 3(b) is the corresponding time sequence. Compared with Figure 2(b), its time sequence must firstly be τ_3 , τ_2 and τ_1 . Nodes with the same hop number can arbitrarily select their time slots.

Compared with the shortest path transmission mechanism, this approach yields a smaller total delay when there are more hops, because it reduces repeated transmission delay.

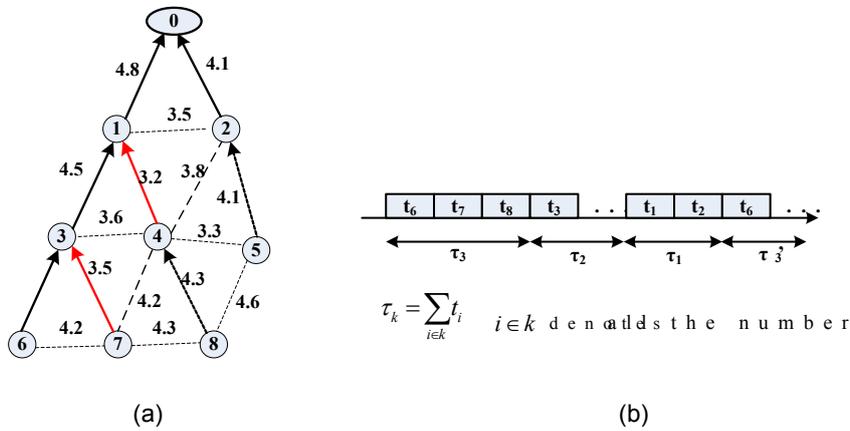


Fig. 3. Accumulated Transmission Hop-by-hop

4.3. Simultaneous Transmission every Three Hops

When two nodes are separated by three hops (such as nodes 15 and 4 shown in Figure 4(b)), there will be no conflict in the channel utilization. So one improved method of Accumulated Transmission is simultaneous transmitting by every three hops. First of all, the last hop nodes (e.g., nodes 15 and 16) transmit data ahead, and at the same time, nodes three hops away from them (i.e., nodes 4, 5, and 6) also send data ahead. When these nodes have completed their transmissions, nodes of the second last hop and corresponding nodes three hops away from them begin to send data. Similarly, the process repeats until all data have reached the gateways. As a result of the simultaneous transmission, the transmission delay could be decreased remarkably for the data that reach the gateway.

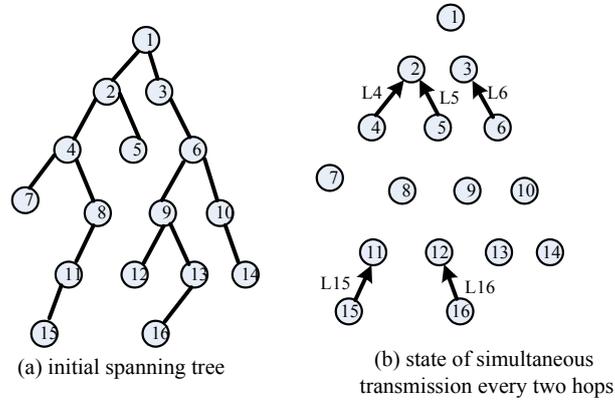


Fig. 4. Simultaneous Transmissions every Three Hops

Figure 5 is the corresponding time sequence of data transmission. Similar to figure 3(b), the last hop and its previous three hops firstly send data simultaneously, take the largest delay in the group of nodes as the transmission delay of this group, then the second last hop and its previous three hops send data simultaneously to gateway, and a period of data collection is completed. Figure 5 also shows, when nodes of the third last hop (τ'_5 in the figure) send data in the first period, node of the last hop can send data in the second period (τ_5 in the figure). Obviously, this transmission scheme can efficiently reduce the delay of the whole network.

With simultaneous transmission every three hops, the minimum delay in the corresponding optimization model must use formulas 8 and 9 instead of formulas 1 and 2; the delay consumed in the transmission of every hop is:

$$t(i) = \max \left(n_i \times t_m + \frac{L_i}{C} + \frac{d_i(e)}{v_p}, n_{i-3} \times t_m + \frac{L_{i-3}}{C} + \frac{d_{i-3}(e)}{v_p}, \dots, n_{i-3k} \times t_m + \frac{L_{i-3k}}{C} + \frac{d_{i-3k}(e)}{v_p} \right) \tag{8}$$

$i = N, N-2, \dots, 1; i-3k > 0; k = 1, 2, \dots$

where $n_i \times t_m$ is the waiting delay of all nodes of the i th hops, L_i/B is all the transmission delay of the i th hop, d_i/v_p is all propagation delay of the i th hop, N is the maximum number of hops.

The total delay of all the packets can be expressed as:

$$\text{Minimize}[t(e)] = \left(\sum_{i=1}^N t(i) \right) \tag{9}$$

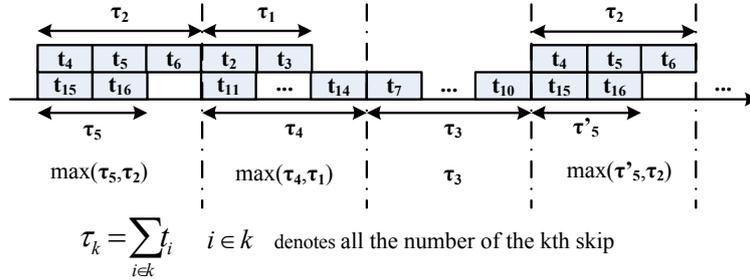


Fig. 5. Time sequence of Data transmission

5. Route Discovery

The underwater nodes, surface gateways and the links between them can be abstracted as a connected graph. To obtain the available routes, the underwater nodes can be viewed as source nodes and surface nodes can be viewed as destination nodes. So the network routes Forms the structure of the minimum spanning trees derived from different objective functions.

5.1. Minimum Delay Spanning Tree

The main contributor of delay are total transmission delay and total propagation delay. The former is related to network size (in terms of hops). The smaller the network, the less transmission delay is in relaying data. Therefore, in order to obtain the minimum total transmission delay, we adopt minimum hops as the optimization goal when constructing the minimum spanning tree. Because total propagation delay is related to total path length, in order to obtain the minimum total propagation delay, we consider the shortest path as the best goal on the spanning tree. Combining the two objectives, we firstly use the minimum hop number and shortest path spanning tree, as shown in left part of Figure 6, treat the gateways as the vertex set and all the underwater sensor nodes form the endpoint set. Then find the edges between the vertex set and endpoint set as the spanning tree. If one endpoint in the endpoint set simultaneously connects many vertexes, then the shortest edge is retained and other edges are removed. Adding all connected endpoints to vertex set, while remove all connected endpoints from the endpoint set, check whether the endpoint set is empty, if not empty, it denotes there are still underwater nodes not included in the spanning tree, so repeat the above steps until the endpoint set is empty.

There is an exceptional case we must consider as shown in Figure 7. Assuming nodes 2 and 3 generate data of length L , acoustic speed is v_p , the distance between nodes is d , channel capacity is C , so the time when data arrives at relay-node 1 before adjusting the spanning tree is $T = 2t_m + (d_{2,1} + d_{3,1})/v_p + 2L/C$, and after adjusting the spanning tree, the time $T = 2t_m + (d_{2,1} + d_{3,2})/v_p + 3L/C$, so the delay is changed by $\Delta T = L/C - (d_{3,1} - d_{3,2})/v_p$. When ΔT is negative, it denotes the delay is decreased, so the adjustment is effective.

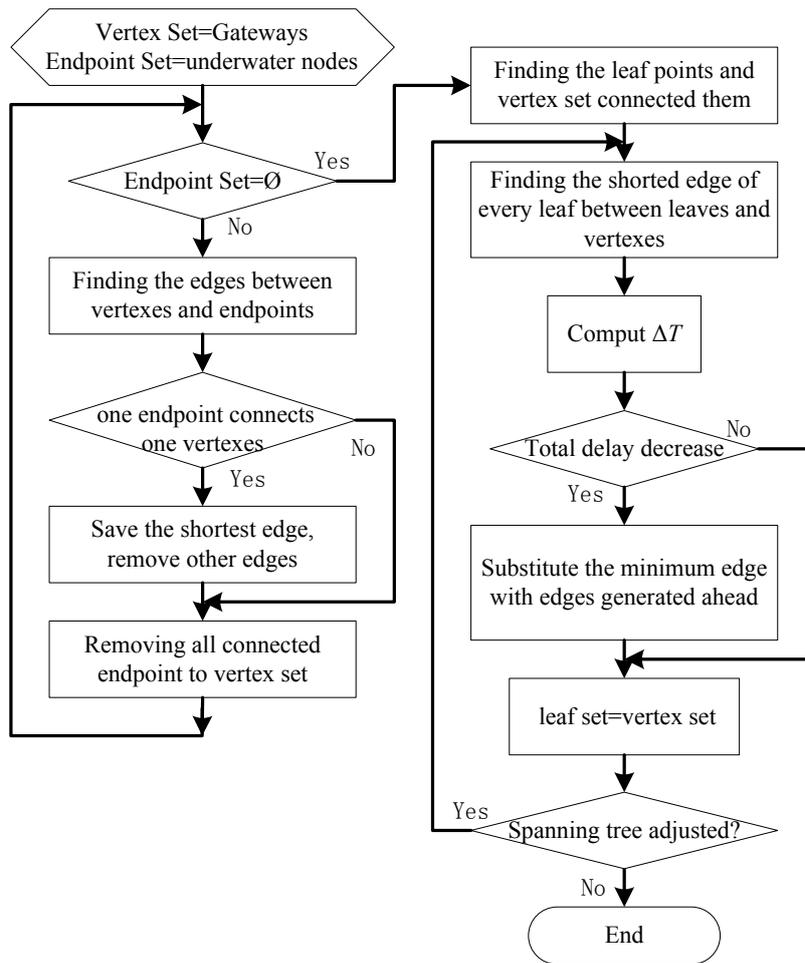


Fig. 6. Flow Chart of Constructing Minimal Delay Spanning Tree

The flow diagram is shown in the right part of Figure 6. First, the algorithm finds all the leaf nodes and the smallest edges connected with them in the spanning tree which is obtained by minimum number of hops and smallest

path, and substitute the minimum edge for connected edges with leaf node in the spanning tree. Judge whether delay is decreased, if so, replace the old edges with new ones in the spanning tree, continue this process until all leaf nodes have been determined, and then remove the leaf nodes and edges connected with them in the spanning tree, find the new leaf in the residual tree. As the judgment mentioned above, the decreased propagation delay and increased transmission delay are (transmission delay is $(n + 1) * L / C$, in which n is the number of nodes whose data should be relayed via this leaf node in the initial spanning tree connected) determined until there is no need for adjustment of spanning tree, at this time, the spanning tree with minimum total delay is obtained. Figure 8 and figure 9 are the minimum delay trees derived by 2-dimensional simulation, and figure 8 is Spanning tree before adjustment while figure 9 is that after adjustment.

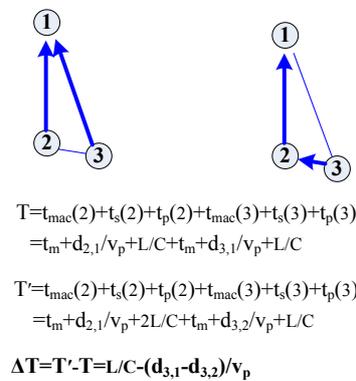


Fig. 7. The Adjustfment of Total Delay.

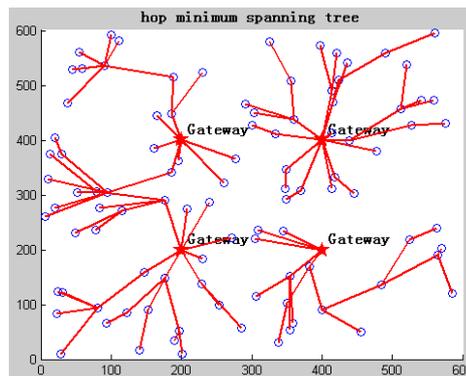


Fig. 8. Minimum Hop-number Spanning Tree of 2-dimensional Diagram

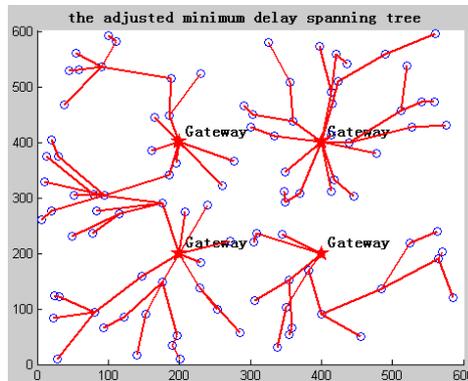


Fig. 9. The Adjusted Minimum Delay Spanning Tree of 2-dimensional Diagram

5.2. Minimum Spanning Tree with Balanced Energy

In UASNs, the energy consumption is one of the key issues we concern about, but how to balance all network nodes, and make the network connectivity as long as possible is one of the research emphasis in UASNs area. The energy consumption of a node is related to the data transmission and reception of the nodes. Therefore in order to achieve balanced energy in the network, we can solve the problem in a perspective of the balance between transmitting data and receiving data. The amount of data being sent and received by a node is inversely proportional to the node connectivity degree; energy balance can be realized when network connectivity degree of node is the smallest, and total energy consumption can be minimized when the hop number of every node is the smallest. Therefore, to achieve energy balanced minimum spanning tree, we can first create spanning tree by the minimum hop number.

Figure 10 is the flow chart. Similar to the left part of Figure 6, we create the connectivity graph via the minimum hop number. When the endpoints of the same hop number is connected with multiple vertices, select the edges connected to the least degree vertices, and remove the edges connected to other vertices. The right part of the figure 10 shows the adjustment when the payload is not balanced as nodes 1 and 8 in Figure 11. As it implies, firstly find the relay-node set of spanning tree, then find the relay-node with the heaviest payload and the corresponding endpoints set, and finally calculate the edges between all nodes in the endpoint set and all other nodes other than the endpoints, find the shortest edge. Then put the node into the endpoints set connecting neighbor relay-nodes of this node with the shortest edge. Compute the adjusted maximum payload. If the maximum payload is decreased, then remove the initial edges in the spanning tree; make the new edge join the spanning tree, at the same time, update maximum payload. Next, judge whether the extreme point set is empty. If it is, it indicates that

the adjustment of this node is completed, and then removes it from the relay-node set. Judge whether the relay-node set is empty, if it is, it denotes that the adjustment of all nodes has completed. Figure 12 is a 2-dimensional minimum spanning tree with balanced energy.

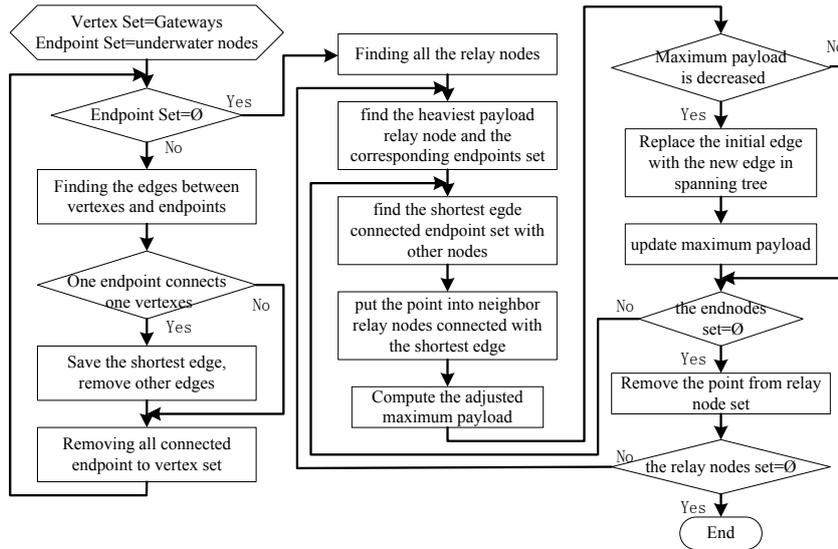


Fig. 10. Flow Chart of Minimum Spanning Tree Obtained with the Objective of Balanced Energy

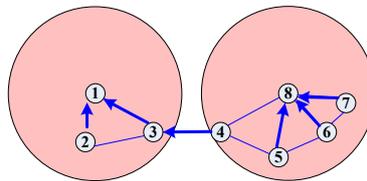


Fig. 11. Adjustment of Payload Imbalance

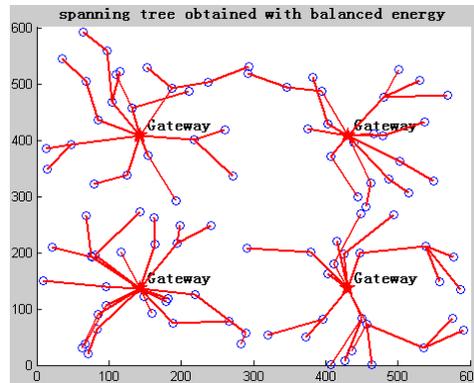


Fig. 12. 2-dimensional Spanning Tree Obtained with the Objective of Balanced Energy

6. Gateway Deployment Based on GA

Genetic algorithm is a global optimized probability searching algorithm, which simulates the evolution process of creature in nature. The population consists of multiple individuals, and it is the computation object, by a repeating iterative process, the individuals will be continuously put on genetic and evaluative operation, and individuals with higher fitness will be transferred to the next generation via the principle of survival of the fittest. So the ultimate result is that individuals with the highest fitness. Because it is a heuristic search based on population, it will find multiple solutions satisfying different preferences in one optimization if it is used in the problem of multiple objectives optimization, and it can handle the objective functions and constraints of all types. Here, we adopt the genetic algorithm, where genetic individual is set to be x coordinate and y coordinate of surface gateways, and the search scale is set to the surface area that underwater nodes corresponds to.

The realization process is as follows: first, when the numbers and location of gateways are fixed, a best route is obtained from underwater nodes to gateway depending on different object functions. When the number of gateways is fixed, different locations of gateways lead to different best routes, so the problem of deployment can be turned into finding the best location of the gateways that achieves the optimal objective function. Mathematical analysis could be used to solve partial differential equations. As the number of gateways increases, it will be very complex to solve this question, so a heuristic search algorithm can be used to obtain the optimal or near-optimal solution. By changing the number of gateways, we can get the corresponding gateway number and a best network performance chart, and get the critical

parameters (the minimal gateway number making network performance optimal).

7. Simulation and Analysis

7.1. Simulation Setup

In order to evaluate the performance of multiple gateway optimal deployment, we simulate a UASN as shown in Figure 13. The packet length is set as $L = 400 \text{ bits}$. The underwater sound propagation velocity is $v_p = 1500 \text{ m/s}$, transmission power $P_s(v) = 1 \text{ watt}$, and reception power $P_r(v) = 0.2 \text{ watt}$. Note that although data packet collision, avoidance, reservation, and waiting cause power consumption in the MAC layer, such power is generally far less than the transmission power. Also the network is randomly deployed in an area of $600 \text{ m} \times 600 \text{ m} \times 600 \text{ m}$, where the largest sound communication distance of underwater sensor nodes and gateway nodes is $R = 150 \text{ m}$. All underwater sensor nodes are arranged randomly and network connectivity can be maintained. The surface gateway location selection uses GA, running parameters are: the sample size is 40, individual length is 25, number of generations is 25, crossing probability is 0.7, and mutation probability is 0.028. The minimum value of each objective function is regarded as a target. The MAC layer adopts TDMA.

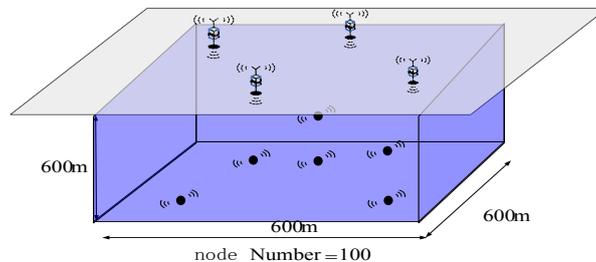


Fig. 13. The Configured Scene for Our Simulations

7.2. Result Analysis

We use a corresponding critical parameter that the minimum number of gateways can get optimum network performance, in order to demonstrate the optimization deployment and verify the advantages of multiple surface gateways in real network deployment.

7.2.1 Relation between the gateway number and network performance

Under usual circumstances, when the number of surface gateways increases, network performance will be improved. To verify this, we derive optimal location deployment of 1-25 gateways using GA according to different optimization objectives. Figures 14 and 15 compare the results of delay and energy consumption calculated by GA after three simulation runs with that calculated randomly. Accumulated transmission hop-by-hop is employed in our simulation. The channel capacity $C = 9600$. The vertical axes of the figures represents packet delay and energy consumption required to reach the gateway, respectively, while horizontal axis shows the number of gateways. From the coincidence degree of the three curves in the graph, we can see stable computational results can be obtained by GA. The simulation results show that the increase of surface gateways can dramatically improve the network performance, in comparison with a single gateway,. It also indicates that the improvement degree of network performance by increasing surface gateways is not evident as surface gateways increases in some cases. When the number of surface gateways reaches a threshold, further increasing the number of gateways cannot noticeably improve network performance. This is because when gateways are enough, underwater nodes can communicate with the nearest gateway in its communication range instead of choosing other gateways newly added to the network. Therefore, it is redundant to increase the surface gateways.

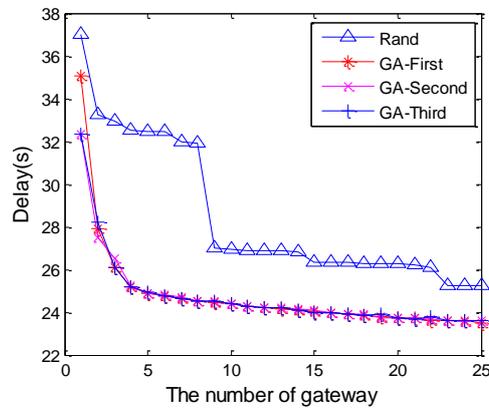


Fig. 14. Delay and the Number of Gateways

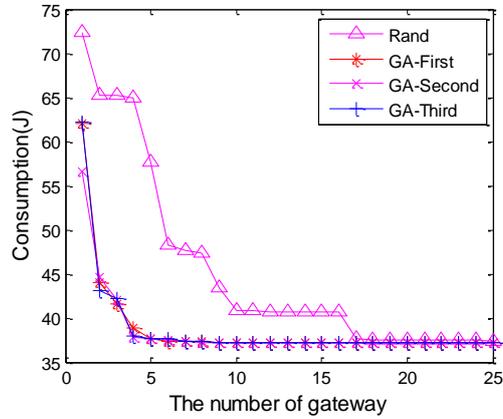


Fig. 15. Energy and the Numbers of Gateways

7.2.2 Effect of Channel Capacity

Obviously, when the ratio of total data generation rate to channel bandwidth of node increases, (including the transmission of its generated data and data to be relayed), more nodes are needed to handle the increased traffic in order to make it accommodate the increase of data transmission rate. We can increase the minimum amount of surface gateways to solve the problem. On the other hand, the increase of channel capacity will reduce the network payload and make it easier to satisfy the constraint of balanced flow.

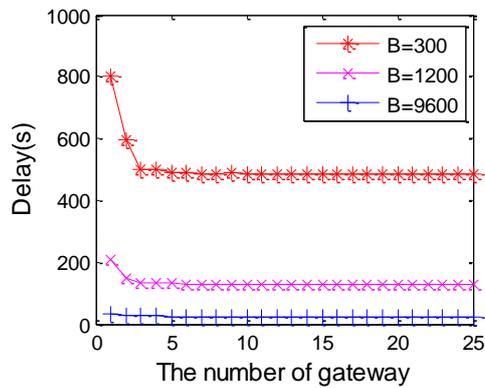


Fig. 16. Delay and the Number of Gateways under Different Rates

To test the effect of channel capacity on network performance, we compare different channel capacities (300,1200 and 9600 bps), and the simulation results show that as network payload increases, the degree of performance

improvement decreases while increasing the number of surface gateways, as the network gradually reaches a saturated state. Figures 16 and 17 show the impact of channel capacity on network delay and energy consumption, respectively. Figure 18 shows the effect of different data rates on the packet loss rate of network.

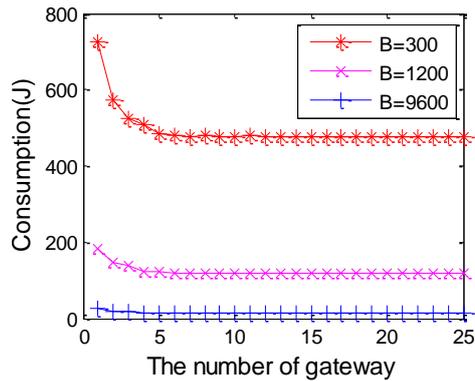


Fig. 17. Energy and the Number of Gateways under Different Rates

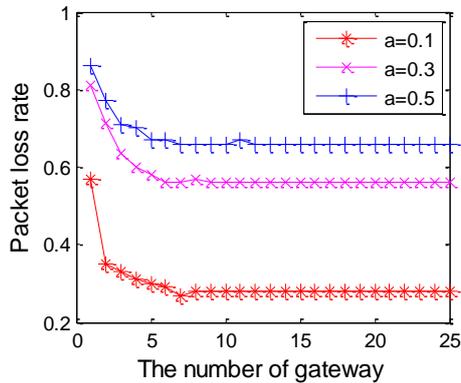


Fig. 18. Packet Loss Rate and the Number of Gateways under Different Data Generation Rates ($g_v = \alpha \times \beta$ is the number of different generation rate)

7.2.3 The Impact of Different Transmission Mechanisms

When the network structure is more complex (e.g., with more hops or heavy payload), different transport mechanisms will result in different network performances, and the mechanism of simultaneous transmission every three hops has a distinct advantage. As shown in Figure 19, when $C = 1200$, this mechanism is compared with accumulated transmission hop-by-hop, where the delay is reduced almost by half. This proves that a lot of nodes simultaneously send data and total delay is reduced. Because simultaneous transmission every three hops can reduce the propagation delay, but energy

consumption of nodes is only related to transmission delay, the mechanism of simultaneous transmission every three hops can make little improvement on energy consumption.

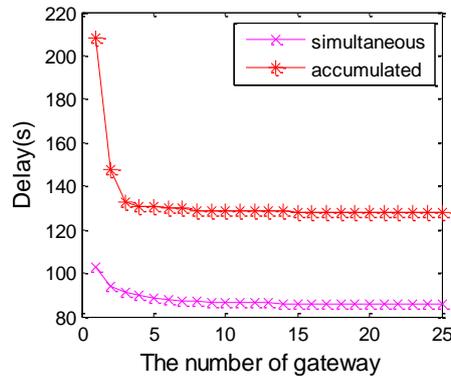


Fig. 19. Delay and the Number of Gateways under Different Transport Mechanisms

8. Conclusions and Future Work

In this paper, the deployment of surface gateways in UASNs is studied, by considering the acoustic characteristics. More specifically, we propose an optimization method of surface gateways deployment dynamically based on genetic algorithm, design a novel transmission mechanism—simultaneous transmission, and realize two efficient routing algorithms that achieve minimal delay and payload balance among sensor nodes. The simulation results show that the use of multiple surface gateways can effectively improve network performance; the number of surface gateways depends on channel capacity (network capacity) and the deployment of underwater sensor nodes; surface gateway location derived by the GA has good stability; and the network delay can be greatly reduced by this mechanism.

In the future, we will investigate possible improvements on joint deployment of surface gateways and underwater nodes. Other future work includes a more accurate conflict model, and selection of suitable MAC protocols that reduce queuing delays.

Acknowledgment. The authors would like to thank the reviewers for their detailed comments that have helped improve the quality of the paper. This research was funded in parts by National Basic Research Program of China (973 Program) 2011CB707106, NSF China 60972044 and 41074023.

References

1. Zhong, Z., Zheng, P., Cui, J.H., Shi, Z.J., Bagtzoglou, A.C.: Scalable Localization with Mobility Prediction for Underwater Sensor Networks. *IEEE Transactions on Mobile Computing*, Vol.10, No. 3, 335 – 348. (2011)
2. Alvarez, A.: Volumetric Reconstruction of Oceanographic Fields Estimated From Remote Sensing and In Situ Observations From Autonomous Underwater Vehicles of Opportunity. *Oceanic Engineering*. Vol. 36, No. 1, 12–24. (2011)
3. Cui, J.H., Kong, J., Gerla, M., Zhou, S.: Challenges: Building Scalable Mobile Underwater Wireless Sensor Networks for Aquatic Applications. *IEEE NETWORK*. Vol. 20, No.3, 12-18. (2006)
4. Lo, K.W., Ferguson, B.G.: Underwater Acoustic Sensor Localization Using a Broadband Sound Source in Uniform Linear Motion. *OCEANS 2010 IEEE – Sydney*. 1–7, (2010)
5. Nie, J.G., Li, D.S., Han, Y.Y., Fu, W.Y., Zhang, G.M.: The Method of Multiple Surface Gateways Positioning in UWSNs. In *Proceedings of the 6th International Conference on Wireless Communications Networking and Mobile Computing*. Chengdu, China.1-5. (2010)
6. Ibrahim, S., Cui, J.H., Ammar, R.: Surface-Level Gateway deployment for underwater sensor network. In *Proceedings of the International Conference on Military Communications*. Orlando, Florida, USA. 1-7. (2007)
7. Chang, C.H., Liao, W.J.: On Multipath Routing in Wireless Mesh Networks with Multiple Gateways. In *Proceedings of the International Conference on Global Telecommunications Conference*. Miami, Florida, USA. 1-5. (2010)
8. Bouk, S.H., Sasase, I.: Multiple end-to-end QoS metrics gateway selection scheme in Mobile Ad hoc Networks. In *Proceedings of the International Conference on Emerging Technologies*. Islamabad, Pakistan. 446-451. (2009)
9. Cagatay Talay,A.: A gateway access-point selection problem and traffic balancing in wireless mesh networks. *Lecture Notes in Computer Science*, Vol. 4448, 161-168. (2007)
10. Bejerano Y.: Efficient Integration of Multi-hop Wireless and Wired Networks with QoS Constraints. *IEEE/ACM Trans on Networking*. Vol. 12, No. 6, 1064-1078. (2004)
11. Pompili D., Melodia T., and Akyildiz I. F.: Deployment Analysis in Underwater Acoustic Wireless Sensor Networks. *The First ACM International Workshop on UnderWater Networks*. 48-55, (2006)
12. Badia L., Mastrogiovanni M., Petrioli C., Stefanakos S., Zorzi M.: An Optimization Framework for Joint Sensor Deployment, Link Scheduling and Routing in Underwater Sensor Networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, Vol. 11, No. 4, 44-56. (2007)
13. Seah W. K. and Tan H.-X.: Multipath Virtual Sink Architecture for Underwater Sensor Networks. In *Proceedings of the International Conference on OCEANS - Asia Pacific*.1-6. (2006)
14. Jo, Y., Bae, J., Shin, H., Nam, H., Ahn, S., An, S.: The Architecture of Surface Gateway for Underwater Acoustic Sensor Networks. In *Proceedings of the 8th International Conference on Embedded and Ubiquitous Computing*, Hongkong, China, 307-310. (2010)
15. Ibrahim, S., Ammar, R., Cui, J.H.: Geometry-Assisted Gateway Deployment for Underwater Sensor Networks. In *Proceedings of the International Conference on Computers and Communications*, Sousse, Tunisia. 932-937. (2009)

Optimization of Multiple Gateway Deployment for Underwater Acoustic Sensor Networks

16. Ghalib A. Shah : A Survey on Medium Access Control in Underwater Acoustic Sensor Networks. In Proceedings of the International Conference on Advanced Information Networking and Applications Workshops, Bradford, United Kingdom. 1178-1183. (2009)
17. Lin, W.L., Li, D.S., Chen, J., Sun, T., Wang, T.: A Wave-Like Amendment-Based Time-Division Medium Access Slot Allocation Mechanism for Underwater Acoustic Sensor Networks. In Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Zhangjiajie, China. 369 -374. (2009)

Ju-gen Nie was born in zhangshu, China, in 1975. He received the Master degree of Circuit and System in 2006 from Wuhan University, Hubei, China. He is currently a Ph.D. in Wuhan University and a lecturer in Communication and Information System at University of Jinggangshan, Jiangxi, China. His research interests are Wireless Sensor Networks, Internet of Things and Underwater Acoustic Networks. He has (co-)authored around 20 research papers published and has been serving on reviewers of many international conferences.

De-shi Li was born in Lingyi, China, in 1968. He received the PhD of Computer Applied Technique in 2001 from Wuhan University, Hubei, China. He is currently a professor in School of Electronic Information, Wuhan University as the Assistant Dean of the department. His research interests include System on Chip, Wireless Sensor Network, Internet of Things and UnderWater Acoustic Networks. He has (co-)authored around 40 research papers published. He has been serving on editorial boards of several international journals and has edited special issues in international journals. He has been member of program committees of many international conferences.

Yan-yan Han was born in Linyi, China, in 1988. She received the Bachelor degree in electronic information engineering from Shandong University, Jinan, China, in 2008 and was recommended as a postgraduate in Wuhan University, Wuhan, China. In 2009, she was selected as a Master-Doctor continuous candidate in communication and information system at Wuhan University. Her research interests include wireless sensor network, mobile network and DTN networks. She has (co-) authored around 10 research papers published.

Received: February 22, 2011; Accepted: April 25, 2011.

A Distributed Power Management Design Based on MOST Networks

Yushan Jin^{1,2}, Ruikai Liu¹, Xingran He¹, and Yongping Huang^{1,2}

¹ College of Computer Science and Technology, Jilin University,
Changchun 130012, P.R. China

² Key Laboratory of Symbolic Computation and
Knowledge Engineering of Ministry of Education,
Jilin University, Changchun 130012, P.R. China
hyp@jlu.edu.cn

Abstract. MOST (Media Oriented Systems Transport) protocol is a high-speed multimedia bus protocol. The system can make more and more media devices in the car automatically collaborate, sharing of audio, video and other data, but its own power consumption has not been a better optimization. In the paper, depending on the network management and the notification mechanism, a distributed power management solutions was designed that the slave nodes can sleep independently and the master node manages the network state, and the wake-up mechanisms in the sleep state were proposed. A mathematical modeling and analysis of MOST networks power were built in MATLAB. This program takes full advantage of MOST network protocol for the intelligent management. Simulation results shown that, with the increasing number of nodes in MOST, energy saving become more effective. More than 20% power saved can be achieved with distributed power management solution in 8-node MOST.

Keywords: power management, MOST, notification, distributed management.

1. Introduction

MOST (Media Oriented Systems Transport) is the de-facto standard for multimedia and infotainment networking in the automotive industry. The technology was designed from the ground up to provide an efficient and cost-effective fabric to transmit audio, video, data and control information between any devices attached even to the harsh environment of an automobile. The features of MOST make it suitable for any application, inside or outside the car that needs to network multimedia information along with data and control functions, for example, CD-Changer, DVD, GPS, Video camera, Phone, Radio, Laptops and communicators and so on. MOST not only defines the physical interconnection between devices but also specifies and standardizes a lean embedded communication protocol and software

framework that simplifies the development of complete systems and applications to distribute and manage multimedia content [1].

It solved some problems about the complexity of car wiring, the increase of the weight and EMI of cars. Although its roots are in the automotive industry, MOST can be used for applications in other areas such as other transportation applications, A/V networking, security and industrial applications. With the devices increasing, it has become a very serious problem that the power cost of MOST enlarged greatly. In some cases, we can reduce system power consumption actually. For example, when using a car phone, all the audio equipment should be shut down.

The current power management is ensured by a central controller in which a program is implemented. This centralized management is based on a „top-down“ approach. The master node as a central controller switched the system state between running, standby and sleep [2]. In this mode, as long as there is a node in work, the entire network is running. Only all the nodes have not the task, the system will sleep or be shut down. The centralized management requires the designer of the control system to be exhaustive in the control flow written in the program. If an event not covered by the system occurs it is unable to respond adequately. Moreover, if the configuration has to be changed (addition or removal of a node), the program must be completely redesigned. So the effect of energy saving with the centralized power management solution is low and not obvious. Therefore this paper gives a more intelligent power management solution based on a „bottom-up“ approach. Every slave node controls its state based on the event, and shares the changes with the other nodes [3]. On the basis of the overall management, we added the control of each node, only to maintain the current working node. In last the design achieved minimum power consumption across the network and reached the green energy-conserving results.

This paper makes the following specific contributions:

- Depending on the network management principle and the notification mechanism, a distributed power management solution was designed that the slave nodes can sleep independently and the master node manages the network state, and the wake-up mechanisms in the sleep state were proposed.
- A mathematical modeling and analysis of MOST networks power were built in MATLAB. After executing the same sequence of tasks generated randomly, the total cost of MOST based on a distributed power management is shown.

The paper is organized as follows: Section 2 introduces related research in power management. Section 3 describes the MOST protocols and principles which we taken as the base for power management. Section 4 addresses the design and implementations include hardware and software designs. Section 5 investigates the efficient and effective implementation of it through comprehensive simulations. Finally, section 6 draws concluding remarks and our future work.

2. Related Works

MOST is a novel network architecture which is designed for high quality of service and efficient transport of audio and video, so related research in MOST is poor and the references on the power management of MOST were very limited. So the related work we focused on was MOST principle and application of power management in other projects.

The centralized management In MOST principle was implemented by a PowerMaster(master node). It broadcasts a query message to each node to make sure whether every node is already to be shut down [2][4]. It was accomplished by a poll procedure so that there would be a heavy load on the bus. The „tp-down“ approach results in the weaknesses such as bad fault tolerance of an element, high operating costs and so on.

Distributed management idea has been introduced by various systems. The distributed management solution was proposed as an application of multi-agent (MAS) to power management in a hybrid power source [3] [5]. Various photovoltaic generator, super-condensators, batteries and grids regarded as agents were developed individually, and communicated between each other for ensure cohesion of the system. A multi-agent system was proposed for a distributed smart grid whose message exchange is designed to be compatible with an IP-based network [6].

In wireless network, [7] demonstrated Busy-time power consumption of 802.11 interfaces can be dramatically reduced by judiciously putting the interface into a power-saving mode for idle intervals as short as several microseconds. Cell2Notify was presented to minimize energy consumption by powering off the Wi-Fi interface when there is no VoIP call in progress, and powering it on only on the reception of an incoming VoIP call [8]. Proxy architecture was proposed for reducing energy waste based on network traffic patterns and user presence indicators [9]. Sleeping and rate adaptation are valuable [10] depending on the power profile of network equipment and the utilization of the network itself for reducing network energy. A power assignment was found, which induced communication graph is an energy t-spanner, and its cost is bounded by some constant times the cost of an optimal power assignment [11].Neighborhood-based Power Management was proposed for conserving energy by allowing neighbors of the signaling node to send data opportunistically [12]. A predictive control algorithm was developed which, in an online fashion, determines the transmission power levels and codebooks to be used by the sensors [13], it conserved sensor energy well.

3. MOST Transmission Principle

In order to design power management solution, a clear understanding of the relevant MOST principle and protocols should be absolutely necessary. In the section, we introduced MOST structure and network management briefly.

3.1. MOST Structure

MOST system is a ring network, which is composed of device nodes connected each other by optical fiber. The node communicates with the MOST ring network through optical-to-electrical transducer. The connection between each node is reciprocal, one-way and point to point. So there is only one path between two nodes. The data frame is transmitted in the direction fixed in the network.

In the MOST network nodes can be divided into the master node and the slave nodes. The master node is the network master controller and the interface between the network and external application. The slave node receives the control message (such as volume tuner) from the master node, able to manage one or more network function [2].

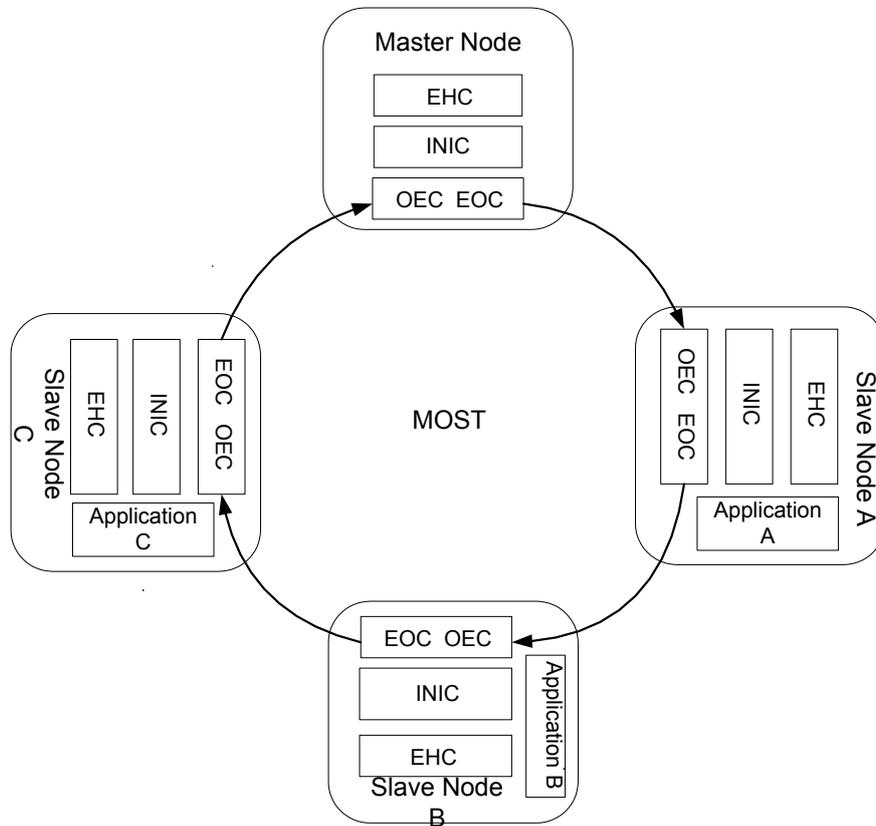


Fig. 1. Structure Diagram of MOST. EHC: External Host Controller. INIC: Intelligent Network Interface Controller. OEC: Optical-to-Electrical Converter. EOC: Electrical-to-Optical Converter

The MOST system uses either 44.1 KHZ or 48 KHZ sample rates for transmitting digital audio signals in hi-fi quality. Devices with a different sample rate can be adapted to the networks by means of a sample rate conversion. Since the MOST system transmits the audio data synchronously, additional data buffering is not needed and complexity of the device is reduced, thus saving costs.

A MOST25 frame consists of 512 bits or 64 bytes. Sixty bytes are used for the transmission of synchronous and asynchronous data. Two bytes transport part of the control message that is made of a total of 32 bytes for the administration of network and notes. The control message is transported over 16 frames that are combined into one block [14]. The first and the last bytes contain control information for the frame. Synchronous and asynchronous areas that share a total of 60 bytes are available in a frame for transmitting streaming and packet data. The bandwidths of the two areas can be adapted to their corresponding requirements by means of boundary descriptor. The boundary between the two areas can be shifted in steps of 4 bytes (a quadlet). The synchronous area can thus have a width of between 24 and 60 bytes and the asynchronous area a width of between 0 and 36 bytes.

3.2. MOST Network Management

The workflow of the MOST networks include: Wake-up start, system initialization, notification, connection management [4].

The components of the MOST system are woken up by a light signal received via the RX diode of the FOT. Basically all components can wake up the network. After a wake-up, the network master first of all builds up the communicative relations to the slave components via a system scan.

If there is a stable lock, the network master starts quarrying all nodes present about their function blocks. It addresses each physical node address and uses the physical node position address as instance. The network master stores the information by quarrying the node in the Central Registry [15] [16]. It is filled with the logical address and the corresponding function blocks of all nodes. As soon as there is a valid system status, the stored information is available for all participants in the ring, the network master can then compare with a previously stored registry and detect possible changes of the current network configuration.

If the Central Registry has not changed after the last system run, the network master sends the message `Configuration.status(OK)`. The notification of this status means for all nodes that the Central Registry can be quarried. The slave components can then establish their communicative relations with each other [17] [18].

In many cases devices must be informed of property changed in other function blocks belonging to other devices. If this had to take place by a poll procedure, there would be a heavy load on the bus. In order to prevent this, the mechanism of notification was created. In the case of a property changed, an event is automatically sent to the device concerned. The notification is

stored in a table, the Notification Matrix which is implemented in the network service layer 2 [15].

4. Design and Implementation

In this section, hardware and software design were given. Hardware design includes selection and application of components, interface design. Software design was divided into three subsections: Slave Nodes Sleep, Network Shutdown, Waking from Device Shutdown.

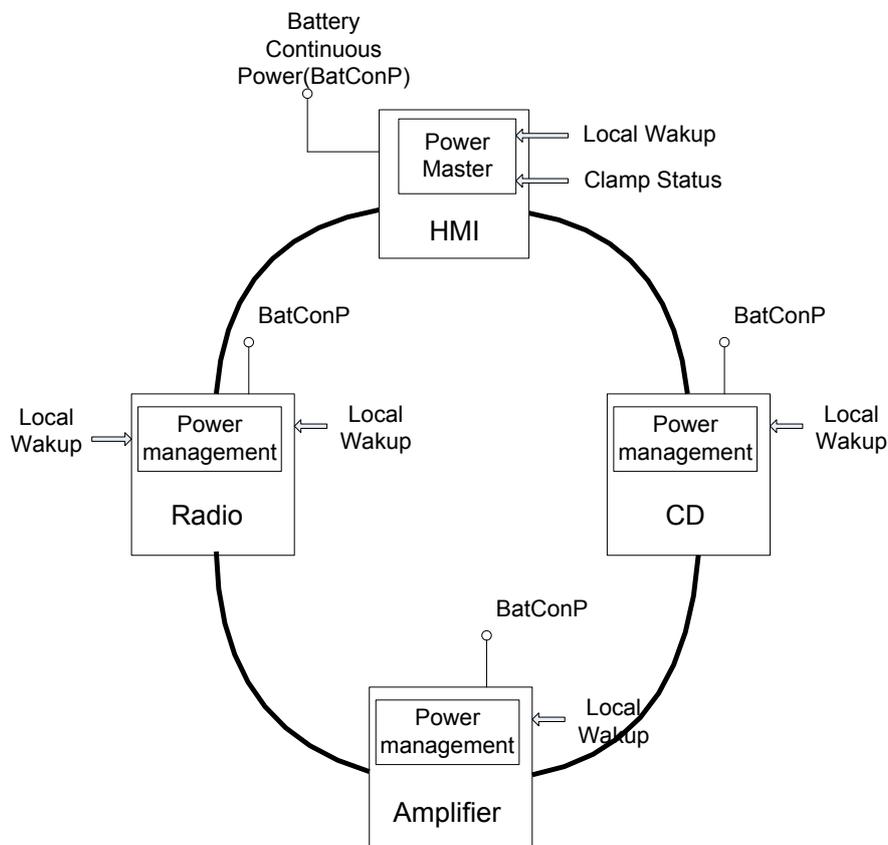


Fig. 2. Power Management Structure Diagram of MOST

Each node has a separate power management module [19]. When there is no task to be executed by a node, after a time threshold, the node's power management module will set the node in sleep state. Before going to sleep state, the slave node will notify the master node. Sleep state of each node is recorded in the master node. When all the nodes are in sleep state, after a

time threshold, the master power management module will make the entire network sleep. Any local wake-up signals from the slave node or the master node can wake up the whole network. The Wake-up signal was transmitted in the direction fixed in the network and the nodes were waked up one by one. The power management structure of the MOST networks is shown in Figure 2.

4.1. Hardware Design

In the node's power management model, the FOR (Fiber Optic Receiver) receives continually the messages from the network to detect the occurrence of the events. So as the controller the power management works without any break. Therefore, the FOR and the power manager are out of power management, and always in a working state. INIC, EHC, and applications are the managed part, which can be set to power, sleep or power-down mode.

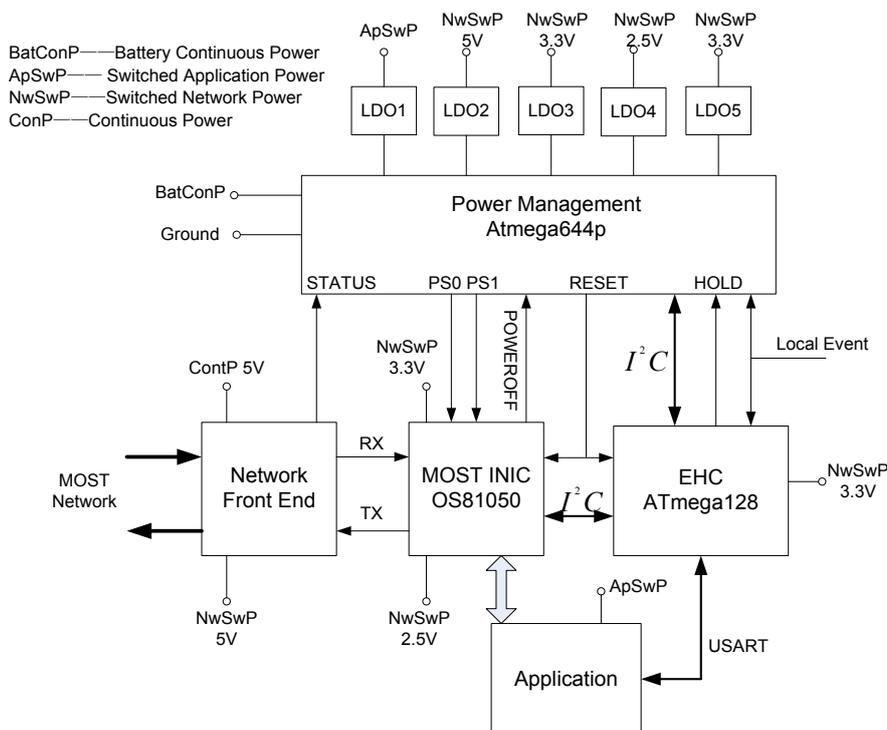


Fig. 3. Hardware Structure of Power Management Module

The devices in MOST nodes which can be managed include the Intelligent Network Interface Controller OS81050 [20], microcontroller ATmega128 and the application equipments. The Intelligent Network Interface Controller (INIC) OS81050 packages messages from the physical layer to the data link layer

and provides the corresponding API functions to application layer. The master controller ATmega128 focuses on the data controlled processing and the entire network management. In the design of power module interface, we set two external interrupt as a wake-up trigger and a closing network interrupt signal. At the same time two reset control ports were established for INIC and EHC. The data transferred between EHC and the power management Module is two-way. So we select the I²C serial bus as the communication which is simple and effective. Foregoing considerations, ATmega644 is quite fit for the control chip of the power management module, which controls the LDO on and off, to provide intermittent power supply for devices. The hardware structure was shown in Figure 3.

The wake-up process started from the network front end or the local event. Network front end received light signals from the network, and sent an external interrupt STATUS to the Power Management Module ATmega644. So the node was waked up. The ATmega644 has begun to trigger the LDOs to supply the other parts with power. The local event is also an external interrupt which can wake up the EHC and power management directly. The 2-wire Serial Interface (TWI) which is compatible with the I²C protocol is ideally suited for typical microcontroller applications. The TWI protocol allows the systems designer to interconnect up to 128 different devices using only two bi-directional bus lines, one for clock (SCL) and one for data (SDA). The EHC is master and the INIC and the ATmega644 are the slave in communication. They all receive and transmit the commands. When the EHC is deciding whether to wake up the network, HOLD is used to maintain power supply in the active state.

4.2. Software Design

Software design must be considered from the whole and part. The master node manages the entire power supply of the network, or shuts down the entire network. The slave node has its own power management features. When there is no command to be executed, the slave node can send a message to the master node and then automatically sleep [21].

4.2.1. Slave Nodes Sleep

When a node has no transactions from request to NetServices, after a delay of inquiry, still no transaction request, the node can be ready to go to sleep. Before sleeping, the node must release the information to the network through the Notification Mechanism [15]. The master node sends `FBlockID.Inst.Notification.Set(DeviceID, FktID)` to each node, so the Notification Mechanism about the sleep state property was launch. `FBlockID` mans the position of the sleep function module; `DeviceID` is the address of the node which has received a notification; `FktID` indicates the

function of the sleep property. The slave node set the notification matrix and equipment matrix through the function `NtfSetNotificationMatrix(*tx_ptr,*rx_ptr)`, then send `FblockID.Inst.Function.Status (Parameter Values)` to inform the master node that the sleep property has been changed, in which Parameter Values is the value of the property. The slave node calls the function `NtfPropertyChanged (device_index_tab_pt)` which mans some changes occurred in the property. The corresponding parameter `device_index_tab_ptr` points to the sleep property in the notification matrix. The NTFS [22] services were called for the message processing. Then contrasted to the notification matrix of the master node, the master node will know the slave node preparing to go to sleep.

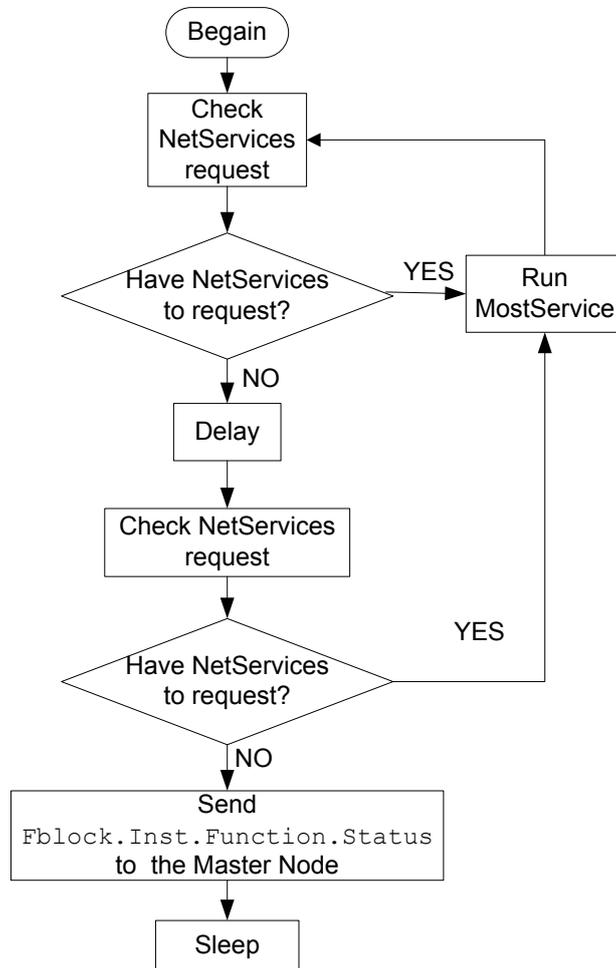


Fig. 4. Flow of Slave Node Sleep

When the slave node is Stand-by state, the external control module Interface (EHCI) is semi-protected. In this state the EHC can not access the network and communicate with the other nodes. It allows, however, full access to the Fblock INIC. EHC can configure the INIC and open ports, such as the streaming Port. Sleep is the EHCI is protected. The other nodes in MOST networks can not access this node's EHC and the communication between EHC and with INIC is also limited. In both states, for the message asked from the MOST networks, NBMIN will return an empty list as the message response.

4.2.2. Network Shutdown

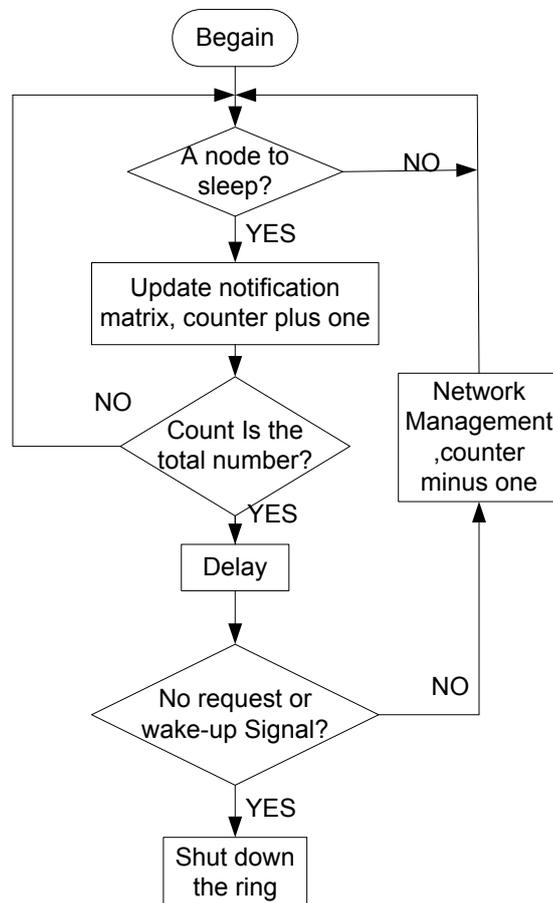


Fig. 5. Flow Chart of Network ShutDown

After every node went to sleep, a counter TIMER0 in the master node will plus one. When the count reaches the total number of nodes in the network,

TIMER0 will overflow and trigger interrupts, then the interrupt service program `ISR(TIMER0_OVF_vect)` will be called. For avoiding that devices have to save their status to persistent memory very often, the master node procedure `ShutDown` has two stages (request and execution). After interrupt, the master waits for `tShutDownWait` before it sends `NetBlock.ShutDown.Start(Query)`. If no respond on `NetBlock.ShutDown.Start(Query)`, the execution will be announced by the master node by starting `NetBlock.ShutDown.Start(Execute)`. Then the network shutdown was completed, which is shown by Figure 5.

4.2.3. Waking from Device Shutdown

The device can be woken up by the master node or the slave node itself.

If the device waked up by master node, the master node sends `NetBlock.ShutDown.Start(WakeFromDeviceShutDown)` to the slave nodes firstly. The FOR in INIC receives the message from optical fiber and wakes its application (EHC) by signaling an external interrupt STATUS. Then the NBEHC registers its own Fblocks using `FblockIDs.Status(FblockIDList)` to the master node, when the master node responds the new Fblocks, the device can be used.

The internal Wakeup is similar to the Wakeup by the master node. They are different in that the application was waked up by local events. After it, the device registers its own Fblocks using `FblockIDs.Status(FblockIDList)`, when the system state is OK or when explicitly by the master node. As soon the master node responds the new Fblocks they can be used.

5. Simulation Test

In this section, the model of most cost was created and the control algorithm was programmed by MATLAB. Based on the model, the cost of 4-node MOST, 6-node MOST and 8-node MOST were simulated, which driven by different types of events. Analysis of images and data are essential.

5.1. Modeling

The demo we simulated included the master node, CD node, radio node, and the amplifier node. In order to build the energy model, we set two random functions `A=randint(1, 10, [0 9])` and `B=randint(1, 10, [0 200])`, "A" generates 10 random numbers as the type of event. 0: No event; 1-6: CD node events (CD eject, Play, Up, Down, Pause, Shutdown); 7-9: radio node events (Play, Up, Down). As long as the CD or the radio node has events, the amplifier node

also starts to work. So there are not events set for the amplifier node. “B” also generates 10 random numbers as the time interval of the 10 events.

$P_{\text{distributed}}$ means the power cost of MOST with the distributed power management; $P_{\text{centralized}}$ means the power cost of MOST with the centralized power management; P_{D1} , P_{D2} , P_{D3} , P_{D4} identify the real-time power of the master node, the CD node, the radio node, and the amplifier node with the distributed power management, respectively. P_{L1} , P_{L2} , P_{L3} , P_{L4} identify the real-time power of the master node, CD node, radio node, and the amplifier node with the centralized power management respectively; P_{min} is the power of each node in sleep, whose device has been turned off; P_1 , P_2 , P_3 , P_4 denote respectively the power of the master node, CD node, radio node, and the amplifier node with all parts of module running. P_1^{on} , P_2^{on} , P_3^{on} , P_4^{on} denote respectively the power of the master node, CD node, radio node, and the amplifier node with the device shut off but the others running [23].

$$P_{\text{Distributed}} = P_{D1} + P_{D2} + P_{D3} + P_{D4} \tag{1}$$

$$P_{D2} = \begin{cases} P'_2 & 0 < A(i) \leq 6 \\ P_{\text{min}} & \text{Else} \end{cases} \tag{2}$$

$$P_{D3} = \begin{cases} P'_3 & A(i) > 6 \\ P_{\text{min}} & \text{Else} \end{cases} \tag{3}$$

$$P_{D4} = \begin{cases} P'_4 & A(i) \neq 0 \\ P_{\text{min}} & \text{Else} \end{cases} \tag{4}$$

$$P_{D1} = \begin{cases} P'_1 & A(i) \neq 0 \\ P_{\text{min}} & \text{Else} \end{cases} \tag{5}$$

$$P_{\text{centralized}} = P_{L1} + P_{L2} + P_{L3} + P_{L4} \tag{6}$$

$$P_{L2} = \begin{cases} P'_2 & 0 < A(i) \leq 6 \\ P_{\text{min}} & A(i) = 0 \\ P''_2 & \text{Else} \end{cases} \tag{7}$$

$$P_{L3} = \begin{cases} P'_3 & A(i) > 6 \\ P_{\text{min}} & A(i) = 0 \\ P''_3 & \text{Else} \end{cases} \tag{8}$$

$$P_{L4} = \begin{cases} P'_4 & A(i) \neq 0 \\ P_{\min} & \text{Else} \end{cases} \quad (9)$$

$$P_{L1} = \begin{cases} P'_1 & A(i) \neq 0 \\ P_{\min} & \text{Else} \end{cases} \quad (10)$$

ΔP is the difference of power consumption between the two methods:

$$\Delta P = P_{\text{distributed}} - P_{\text{centralized}} = \begin{cases} 0 & ; A(i) = 0 \\ P_{\min} - P''_3 & ; 0 < A(i) \leq 6 \\ P_{\min} - P''_2 & ; A(i) > 6 \end{cases} \quad (11)$$

It can be seen from the formula (11), when no event request, $\Delta P = 0$, which shows the power consumption of two methods is same. Now the system with two methods is also in sleep state. After examination, $P_{\min} = 0.3w$; $P_2^{\text{min}} = P_3^{\text{min}} = 0.4w$. So when the CD or the radio node has a request, $\Delta P < 0$, that means the power consumption of MOST system with the distributed power management is less than the power consumption of MOST system with the traditional power management.

5.2. Simulation Results

Based on the model, experiments were conducted and the simulation images were given by MATLAB. The random sequences were created in simulation for 4-node MOST:

$A = [0 \ 7 \ 4 \ 9 \ 4 \ 4 \ 8 \ 5 \ 2 \ 6];$

$B = [98 \ 70 \ 163 \ 73 \ 27 \ 40 \ 39 \ 121 \ 54 \ 39];$

The corresponding graph is Figure 6.

In figure 6, $P(\text{distributed})$ is the power cost of MOST system with the distributed power management. $P(\text{centralized})$ is the power cost of MOST system with the centralized power management. It can be seen from the figure the minimum power consumption is 1.2W, when the system is not the task request, the minimum power consumption of the two methods is also 1.2W. When a task requests the radio, $P(\text{distributed}) = 2.1W < P(\text{centralized}) = 2.0W$. When a task requests the CD, $P(\text{distributed}) = 2.3W < P(\text{centralized}) = 2.4W$.

Similarly, a 6-node MOST networks could be simulated, which includes the master node, CD node, radio node, the amplifier node, DVD node and display node. The simulation result is Fig. 7.

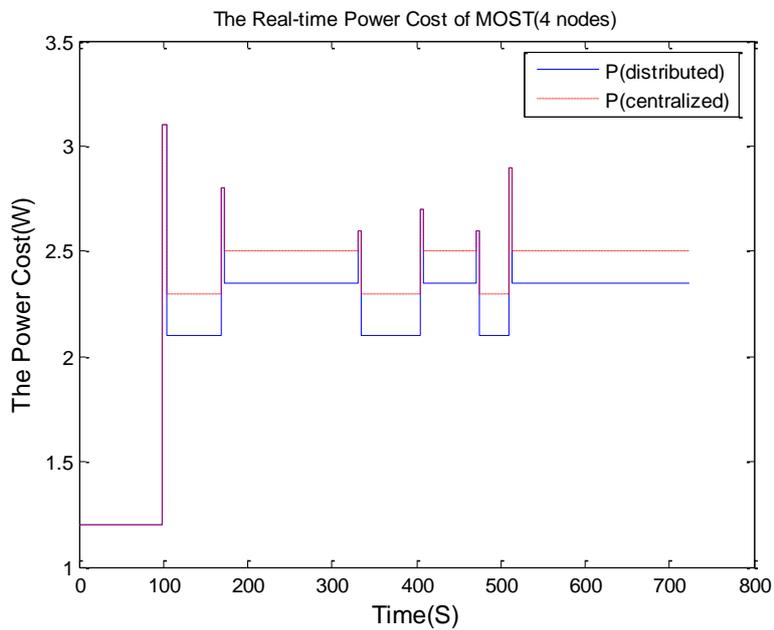


Fig. 6. Simulation Power Cost of 4-node MOST

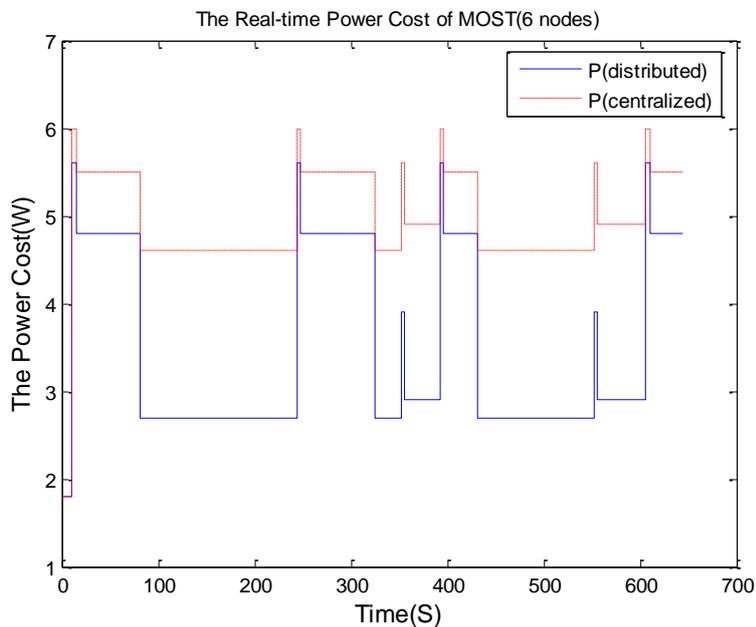


Fig. 7. Simulation Power Cost of 6-node MOST

We also obtained an 8-node MOST system by adding a MP3 node and a telephone node into 6-node MOST. The corresponding simulation result is Figure 8.

As Figure 6 shown, $P(\text{distributed})$ curve is close to $P(\text{centralized})$ curve, the energy saving result is not apparent. Figure 7 illustrates a better result than Figure 6 and Figure 8 indicated the best effect. $P(\text{distributed})$ curve was obviously lower than $P(\text{centralized})$ curve in Figure 8. After the contrast and analysis of Figure 6, Figure 7 and Figure 8, we found that the more nodes in MOST networks, the greater effect achieved by distributed power management. So a classified summary of statistical data was completed to prove it.

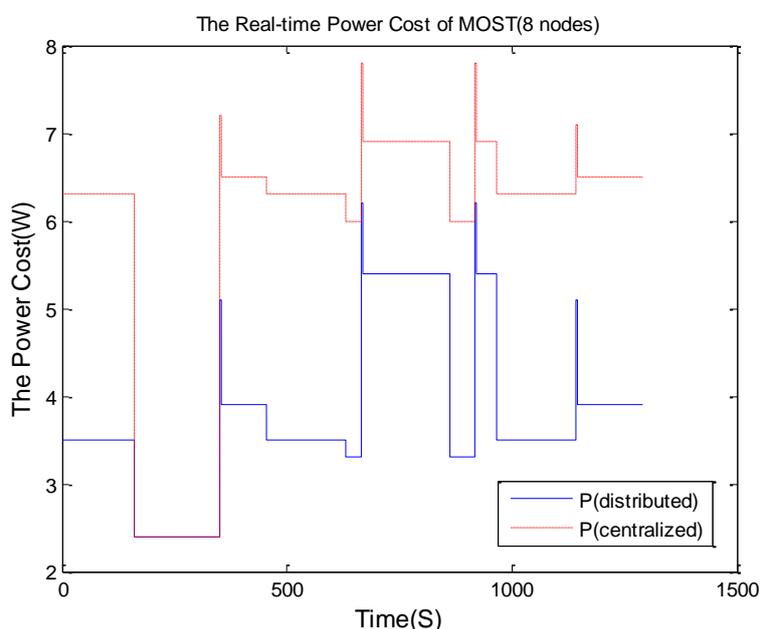


Fig. 8. Simulation Power Cost of 8-node MOST

Table 1 shows the cost of three MOST systems based on centralized management and distributed management. $A(1,i)$ as an element of random event matrix A denotes the type of event. Firstly, no matter how many nodes in MOST, $P_{\text{centralized}}$ equals $P_{\text{distributed}}$ when no events request, $\Delta P=0$. As long as there is an event to process, $P_{\text{distributed}}$ is smaller than $P_{\text{centralized}}$. This is due to free nodes shut down by the distributed management program. Secondly, we can make conclusions from Figure 9 that with the nodes increasing in ring-network, the energy saved ΔP raises rapidly. The maximum ΔP in 4-node mode is 0.2w, 2.0w in 6-node mode, and increase to 3.2w in 8-node mode. Energy-saved growth rate changed from 8.7% to 50.3% accordingly. The more nodes in MOST, the more free nodes will be shut down in processing a determinate event. For example, when a MP3 event occurred in 8-node

MOST, the MP3 and amplifier node set up the communication between each other and the master node still run for supervision, while the other five nodes can be shut down. Compared with the centralized management, the distributed management saved more energy from five nodes, especially for high power node such as DVD. This explained the reason: the more nodes in MOST networks, the greater effect achieved by distributed power management.

It can be seen from Figure 6, Figure 7, and Figure 8, when switching task between nodes or waking up from sleep, the power consumption of system increased dramatically, because the current peak appears.

Table 1. Comparisons of Various System Energy Saved

	No A(1,i)=0	CD 1≤A(1,i)≤6	FM 7≤A(1,i)≤9	DVD 10≤A(1,i)≤15	MP3 16≤A(1,i)≤21	Phone 22≤A(1,i)≤30
4-node P _{centralized}	1.2	2.5	2.3			
4-node P _{distributed}	1.2	2.35	2.1			
4-node ΔP	0	0.15	0.2			
4-node ΔP/P _{centralized}	0%	6%	8.7%			
6-node P _{centralized}	1.8	4.9	4.6	5.5		
6-node P _{distributed}	1.8	2.9	2.7	4.8		
6-node ΔP	0	2.0	1.9	0.7		
6-node ΔP/P _{centralized}	0%	40.8%	41.3%	12.7%		
8-node P _{centralized}	2.4	6.3	6.0	6.9	6.35	6.5
8-node P _{distributed}	2.4	3.5	3.3	5.4	3.15	3.9
8-node ΔP	0	2.8	2.7	1.5	3.2	2.6
8-node ΔP/P _{centralized}	0%	44.4%	45%	21.7%	50.3%	40%

Only when the total power saving of the system is greater than the power consumption of switching the state and start-up, the power management is effective. We set P_u is the system power during wake-up, P_w is the power of the working process, P_e is the power of the system into sleep, P_s is the system power in sleep, T_{sd} is the time to sleep, T_{wd} is the wake-up time, and T_{th} is the limit time of a effective power management. So we have:

$$\begin{aligned}
 (P_u - P_w) \times T_{wd} &\leq (P_w - P_e) \times T_{sd} + (P_w - P_s) \times (t - T_{wd} - T_{sd}) \\
 t &\geq \frac{(P_u T_{wd} + P_e T_{sd} - P_s (T_{sd} + T_{wd}))}{P_w - P_s} = T_{th}
 \end{aligned}
 \tag{12}$$

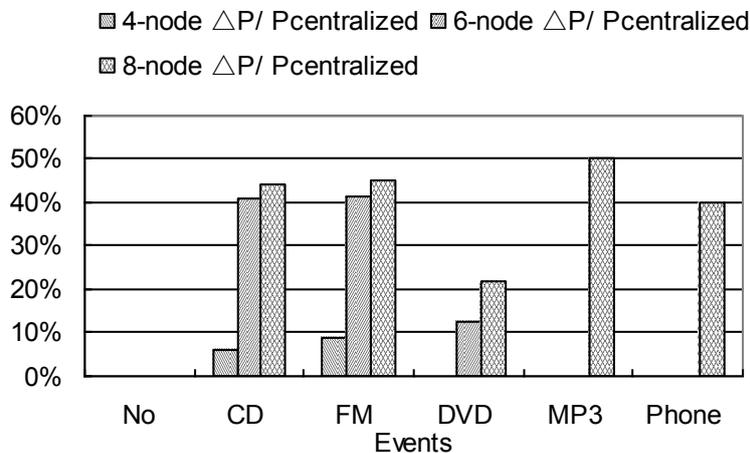


Fig. 9. Comparison of Energy Saved Efficiency Based on Various Events

We obtained from the formula (12), free time interval must be greater than T_{th} , and the system could go to sleep. Otherwise, the energy consumed when the wake-up will be more than the energy savings, so the management is useless. Switching state too often will make the system of energy consumption higher, we selected time valve for 10 seconds.

6. Conclusions

In this paper, the objective of the distributed power management program is that the node can control separately its state in the absence of task requests, regardless of the status of other nodes. The traditional management model is a master switch. Each node can not manage separately, which results that the node still work without the task, waste of the energy. As can be seen from the test, when there are tasks in the system, distributed power management is always superior to the traditional power management. As the system will perform a task, not all nodes need to participate, so we can close some idle nodes to save energy.

Although the proposed method has achieved good results, there is still much room for optimization. For example, when preparing to shut down the node, how to elect the size of the time period [24] [25] to be extended is a problem. If it is too long, there will be no effect, and if too short, it will increase the system's power consumption. Power and performance management via lookahead control [26] or prediction algorithm [27] offer a reference way to our subject, which always virtualized or studied the workload laws. How to do more intelligent management and the imitation of the event's law will be the next focus of consideration.

References

1. Grzemba, A.: Annual Achievement Report 2010. [Online]. Available:http://www.mostcooperation.com/publications/brochures_newsletters/archive/index.html?backurl=%2Fhome%2Findex.html (current March 2011)
2. Grzemba, A.: MOSTSpecification_3V0E2. (2010). [Online]. Available:http://www.mostcooperation.com/publications/Specifications_Organizational_Procedures/index.html?doc=4425&dir=291 (current March 2011)
3. Lagorse, J., Paire, D., Miraoui, A.: A multi-agent system for energy management of distributed power sources. *Renewable Energy*, vol. 35, no1, 174-182. (2010)
4. Grzemba, A.: MOST - The Automotive Multimedia Network- From MOST25 to MOST150. *Deggendorf.140-152.*(2008)
5. Jun, Z., Junfeng, L., Jie, W., Ngan, H. W.: A multi-agent solution to energy management in hybrid renewable energy generation system. *Renewable Energy*, Volume 36, Issue 5, Pages 1352-1363. (2011)
6. Pipattanasomporn, M., Feroze, H., Rahman, S.: Multi-Agent Systems in a Distributed Smart Grid: Design and Implementation. *Proc. IEEE PES 2009 Power Systems Conference and Exposition (PSCE'09)*, Seattle, Washington, USA, 1 – 8, 15-18 .(2009).
7. Jiayang, L., Zhong, L.: Micro Power Management of Active 802.11 Interfaces. *Proceeding of the 6th international conference on Mobile systems, applications, and services*, New York,146-159.(2008)
8. Agarwal, Y., Chandra, R., Wolman, A., Bahl, P., Chin, K., Gupta, R.: Wireless wakeups revisited: energy management for voip over wi-fi smartphones. *MobiSys '07 Proceedings of the 5th international conference on Mobile systems, applications and services*, New York, NY, USA ,179-191.(2007)
9. Nedeveschi, S., Chandrashekar, J., Liu, J., Nordman, B., Ratnasamy, S., Taft, N.: Skilled in the Art of Being Idle: Reducing Energy Waste in Networked Systems. *Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, USENIX Association Berkeley, CA, USA , 381-394.(2009)
10. Nedeveschi, S., Popa, L., Iannaccone, G., Ratnasamy, S., Wetherall, D.: NSDI'08 *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, USENIX Association Berkeley, CA, USA ,323-336.(2008)
11. Aschner, A., Katz, C.: Minimum Power Energy Spanners in Wireless Ad Hoc Networks, *INFOCOM, 2010 Proceedings IEEE*, San Diego, CA , On page(s): 1 - 6 ,14-19. (2010)
12. Ashraf, F., H.Kravets, R.: Poster Abstract: Neighborhood-based Power Management. *ACM SIGMOBILE Mobile Computing and Communications Review*, New York, NY, USA, Volume 13 Issue 3, 30-33. (2009).
13. Ahl n, Q., Stergaard: Energy Efficient State Estimation with Wireless Sensors Through the Use of Predictive Power Control and Coding. *Signal Processing*, *IEEE Transactions on*, Volume: 58 Issue: 9, 4811 – 4823. (2010).
14. Grzemba, A.: MOST NetServices Layer I Wrapper for INIC. Karlsruhe, Germany: MOST Cooperation. (2008).
15. Grzemba, A.: MOST NetServices Layer II. Karlsruhe, Germany: MOST Cooperation. (2008)
16. Bo, C., Yong-ping, H., Wei, C.: Design and Implementation of in-Vehicle MOST Control Node. *Journal of Jilin University (Information Science Edition)*, Vol 27, No 5,447-449. (2009)
17. Grzemba, A.: MOST FunctionBlock NetBlock. (2007-6) [Online] Available:http://www.mostcooperation.com/publications/Specifications_Organizational_Procedures/index.html?doc=2868&dir=223 (current March 2011).

18. Bo, C., Yong-ping, H., Wei, C., Feng-hua, Z.: MOST-Based in-Vehicle Audio System. Journal of Jilin University (Information Science Edition), Vol 27, No 5, 472-475. (2009)
19. Anand, M., Nightingale, E., Flinn, J.: Self-tuning wireless network power management. Proceedings of the 9th annual international conference on Mobile computing and networking. Volume 11 Issue 4. (2005).
20. SMSC Cooperation. MOST OS81050 INIC. (2007) [Online]. Available: http://www.smsc-ais.com/AIS/component?option,com_content:9-20 (current March 2011)
21. Sinha, A., Chandrakasan, A.: Dynamic power management in wireless sensor networks. Design & Test of Computers, IEEE. 18 Issue: 2, on page(s): 62 – 74. (2001).
22. Grzempa, A.: Dynamic Specification. (2007). [Online]. Available: http://www.mostcooperation.com/publications/Specifications_Organizational_Procedures/index.html?doc=2865&dir=291 (Current March 2011)
23. Dunkels, A., Osterlind, F., Tsiftes, N., Zhitao H: Software-based sensor node energy estimation. Proceedings of the 5th international conference on Embedded networked sensor systems. ACM New York, NY, USA, 409-410. (2007)
24. Srivastava, M. B., Chandrakasan, A. P., Brodersen, R. W.: Predictive system shutdown and other architectural techniques for energy efficient programmable computation. Very Large Scale Integration (VLSI) Systems, IEEE, On page(s): 4 , Issue: 1, 42. (1996)
25. Chi-Hong, H., Wu, A. C. H.: A predictive system shutdown method for energy saving of event-driven computation. ACM Transactions on Design Automation of Electronic Systems, Pages 226-241. (2000)
26. Kusic, D., Kephart, J., Hanson, J., Kandasamy, N., Guofei J: Power and Performance Management of Virtualized Computing Environments via Lookahead Control. Cluster Computing, Volume 12, Number 1, 1-15. (2008).
27. Hiltermann, J., Lodewijks, G., Schott, D. L., Rijsenbrij, J. C., Dekkers, J. A. J. M., Pang, Y.: A Methodology to Predict Power Savings of Troughed Belt Conveyors by Speed Control. Particulate Science and Technology: An International Journal, Volume 29, Issue 1, Pages 14 – 27. (2011)

Yushan Jin, born in 1963. Associate Professor. Her main research interests include intelligent control system and evolvable hardware.

Ruikai Liu, born in 1985. Master. His main research interests include embedded system and automatic control.

Xingran, born in 1984. Master. His main research interests include intelligent control and embedded system.

Yongping Huang, born in 1964. PhD. Associate Professor. His main research interests include embedded system, intelligent control system and IPv6 for Smart Object Networks.

Received: March 1, 2011; Accepted: April 23, 2011.

An Energy-Efficient Localization Strategy for Smartphones

Haifeng Liu, Feng Xia*, Zhuo Yang, and Yang Cao

School of Software, Dalian University of Technology,
Dalian 116620, China
f.xia@ieee.org

Abstract. In recent years, smartphones have become prevalent. Much attention is being paid to developing and making use of mobile applications that require position information. The Global Positioning System (GPS) is a very popular localization technique used by these applications because of its high accuracy. However, GPS incurs an unacceptable energy consumption, which severely limits the use of smartphones and reduces the battery lifetime. Then an urgent requirement for these applications is a localization strategy that not only provides enough accurate position information to meet users' need but also consumes less energy. In this paper, we present an energy-efficient localization strategy for smartphone applications. On one hand, it can dynamically estimate the next localization time point to avoid unnecessary localization operations. On the other hand, it can also automatically select the energy-optimal localization method. We evaluate the strategy through a series of simulations. Experimental results show that it can significantly reduce the localization energy consumption of smartphones while ensuring a good satisfaction degree.

Keywords: smartphone, energy efficiency, localization, mobile applications

1. Introduction

With smartphones being increasingly pervasive in the past few years, lots of Location-Based Applications (LBAs) have been developed to meet the needs of human life. Those currently popular LBAs include mobile social networking, mobile search, healthcare, traffic monitoring [1, 2, 3], etc.

The core enabler of LBA is the localization technique, which is used to obtain position information. A great number of wireless localization techniques have been outlined by Liu et al. [4], among which GPS is preferred to other techniques such as WiFi and GSM based positioning system, because of its higher accuracy. However, GPS is extremely energy-hungry, which will largely shorten battery lifetime of smartphones when GPS based LBAs are running. This particular theory has been validated by tremendous experimental results

* Corresponding author: Feng Xia (f.xia@ieee.org)

[5, 6, 7]. Meanwhile, how to get position information with minimal energy has become one of the primary issues for recent research studies.

Numerous solutions have been proposed to address the problem of high localization energy consumption of GPS. They are mainly classified into three categories. The first category is to replace GPS with other methods which are more energy-efficient but less accurate, such as WiFi [8, 9] and GSM [10] positioning system. The second category is to avoid unnecessary GPS localization operations. For instance, it requires no localization operation when the user is stationary or the user does not move beyond the limited range during a particular period. The third category is to adopt the hybrid method which continuously switches among multiple localization techniques.

In this paper, we present an energy-efficient localization strategy for smartphone applications based on the following three observations. First, LBAs do not always require the highest accuracy and their accuracy requirements will vary as the user moves in general. For example, location-based social networking application can change accuracy requirement depending upon the positions of users. Proximity can be efficiently perceived using method proposed by Kupper and Treu [11]. This method calculates the required accuracy limit for each user by utilizing the distance between them. The accuracy limits might range from 10 meters, if the users are close, to several kilometers if they are far apart. Since the application accuracy requirement is relaxed as the user moves, a less accurate but more energy-efficient localization method can be selected. Second, it is always not necessary to locate the user again if it has been away for some time while not beyond the accuracy limited range. Hence, during the particular period, it needs not to locate the user and a waste of energy is avoided. Third, most of the time, it is easy to obtain the velocity of a user. We have observed that the accelerometer is often used with the method proposed by Brezmes et al. [12, 13] to detect whether a user is stationary or not. At this time, we can leverage accelerometer to determine velocity cheaply. In addition, we can also obtain velocity by interacting with intelligent public transport system when we are on a bus.

Based on the above observations, our proposed scheme is designed to save energy in two main respects. On one hand, it can reduce localization operations by dynamically estimating the localization time point. On the other hand, it can select the energy-optimal localization method.

This paper makes the following contributions:

- 1) We present an energy-efficient localization strategy for smartphone applications which minimizes the energy consumption while satisfying the application accuracy requirement.
- 2) We evaluate the effectiveness of the proposed strategy through simulations. Experimental results show that our scheme significantly outperforms the existing GPS method.

The remainder of the paper is organized as follows: Section 2 presents the related work, which is followed by the introduction of our energy-efficient localization strategy in Section 3. In Section 4, we evaluate the performance of the proposed strategy. Section 5 concludes the paper.

2. Related Work

Recently, with the ever-increasing quantities of LBAs of smartphones, GPS has been frequently used in various applications because it is highly accurate. However, GPS is exceedingly energy-hungry. This will largely shorten battery lifetime of smartphones when LBAs are running.

To solve this problem, many previous works have already studied alternative localization methods to replace the GPS technique. Otsason et al [14] presented a GSM-based indoor localization system. WiFi positioning systems were proposed by Khan and Akidi [15]. Additionally, a great number of other techniques such as Bluetooth [16, 17, 18], UWB [19, 20], and RFID [21, 22], are also used to determine the position information. Nevertheless, most of those methods are less accurate.

Some other works focus on how to improve GPS and combine GPS with other methods to conserve energy and prolong battery life. Zhuang et al [23] presented a localization framework for smartphone applications, which also overviewed energy-efficient localization techniques. Kjargaard [24] summarized many energy-saving methods for smartphones from different levels. Abdesslem et al. [7, 25, 26] used an accelerometer to detect whether the user is stationary or not. If the user has been stationary for a long time, GPS is not necessarily used during this period. These works effectively avoided unnecessary localization operations when the user is in static state. Paek et al [6] presented a rate-adaptive GPS-based positioning system. Compared to duty-cycled GPS, this work dynamically estimated the user's velocity depending on the location-time history of the user. In fact, it leveraged user's daily living habits. Kjaergaard et al [26] used the user's velocity measured by GPS to estimate the localization time point, which is more inaccurate compared with our velocity sampling algorithm. Farrel et al [27] considered to reduce GPS energy consumption by assuming that the localization operation is only required when the user enters a specified area. Constandache et al [28] added consideration for given energy budget to the problem of localization. Lin et al [29] presented a Bayesian estimation framework to model users' locations and sensor errors. However, the energy-optimal localization method is not always selected.

3. Energy-Efficient Localization

3.1. Overview

We consider LBAs of smartphones equipped with GPS, WiFi, and GSM positioning interfaces. As the user moves, LBAs need to continually determine and update positions. When the user is stationary, we can use the method proposed by Huang et al [25, 26, 27] to detect the static state and reduce

unnecessary localization energy consumption. Hence, we only consider energy-efficient localization when the user is always in mobile state.

Application accuracy requirements vary as the user moves. We assume that we can obtain the application accuracy requirement anytime, and let a_t be the application accuracy requirement at time t . If there are multiple applications at the same time, a_t will be equal to the minimum value of accuracy at time t .

Further, for GPS, WiFi, and GSM positioning systems, we make the following assumptions. The energy consumption per localization operation for the three systems is a constant value, which is denoted with E_g , E_w , and E_m respectively. Additionally, the localization accuracy is also constant, respectively denoted with a_g , a_w , and a_m . Moreover, the process of localization is assumed to occur instantaneously.

As for the user's movement in a period of time, let E_T denote the total energy consumption for duration T , and E_{t_i} denote the energy consumption at time t_i , which is the i th localization time point. E_T can be defined as follows.

$$E_T = E_{t_1} + E_{t_2} + E_{t_3} + \dots + E_{t_i} + \dots \quad i = 1, 2, 3, \dots \quad (1)$$

In order to minimize the total energy consumption E_T , we propose an energy-efficient localization strategy, whose flow chart is depicted in Figure 1.

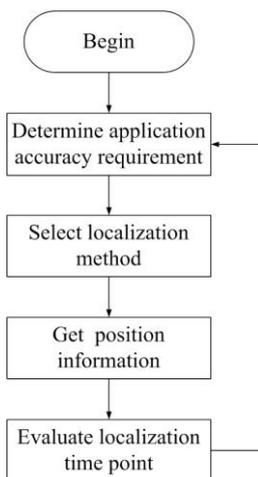


Fig. 1. Flow chart of our energy-efficient localization strategy

Our solution considers conserving energy consumption in the following two respects. First, we avoid unnecessary localization operations by dynamically estimating the next localization time point and sampling velocity. If the user has been in movement for some time within the range of application accuracy limit, it is not necessary to locate the user. Second, when it is time to locate the user again, we use the energy-optimal method by calculating the average energy consumption of each strategy and selecting the least one.

3.2. Estimating Localization Time Point

To avoid unnecessary localization operations, we sample velocity to compute the movement range of the user and dynamically estimate the localization time point. It shall be the time to locate the user again as long as the user moves beyond the application accuracy limited range.

It is assumed that we can obtain the velocity of a user anytime because of the observation that it is easy to determine velocity when an accelerometer is used to detect whether the user is stationary or not. Furthermore, it can also be based on other observations. For example, we can use the bus's velocity as a user's velocity by interacting with intelligent public transport system when the user is on a bus, or use GPS to determine velocity when it is being used to get position information. In addition, we let $v(t_{ij})$ denote the sampled velocity at time t_{ij} , where i and j are defined as the i th time in localization and the j th time to sample velocity respectively.

The detailed process of estimating localization time point is described as follows.

Algorithm 1. Estimating localization time point

1. At localization time point t_{ij} , set $r_i = 0$;
 2. Compute $v_e(t_{ij}) = \alpha \times v(t_{ij}) + (1 - \alpha) \times v_e(t_{(i-1)(j-1)})$;
 3. Obtain a_{t_i} , select M_{t_i} using Equation (4);
 4. Compute $t_s = (a_{t_i} - a_{M_{t_i}}) / v_e(t_{ij})$;
 5. Set $j = j + 1$, $t_{ij} = t_{i(j-1)} + t_s \times \beta$ ($0 < \beta \leq 1$);
 6. Compute $v_e(t_{ij}) = \alpha \times v_e(t_{ij}) + (1 - \alpha) \times v_e(t_{i(j-1)})$;
 7. Compute $r_i = r_i + v_e(t_{ij}) \times t_s \times \beta$ ($0 < \beta \leq 1$);
 8. **if** $r_i < (a_{t_i} - a_{M_{t_i}})$ **then**
 9. go to step 5;
 10. **else**
 11. **return** t_{ij} ;
 12. **end if**
-

In the algorithm, M_{t_i} is defined as the method selected at localization time point t_i . It can be obtained from Equation (4), which will be described in the following subsection. $v_e(t_{ij})$ denotes the velocity estimated by the Exponentially Weighted Moving Average (EWMA) method, during the interval between two consecutive velocity samplings. r_i , which represents the estimated movement distance since the localization time point t_{ij} , can be obtained from multiplying $v_e(t_{ij})$ and the particular interval. The coefficient α is a factor ranging from 0 to 1. A higher value of α means a closer velocity estimated to the current velocity. Additionally, β is a factor used to adapt the time point of the next velocity sampling. When the coarse localization interval t_s is computed, we

sample velocity after $t_s \times \beta$ and compute velocity estimated by EWMA method to determine movement distance. By using this method, we can dynamically estimate the localization time point.

Also, parallel to the process, we need to monitor the application accuracy requirement. Whenever it changes, we must go back to step 1 to locate the user again.

3.3. Selecting Localization Method

The objective of this subsection is to select the energy-optimal localization method, which satisfies the application accuracy requirement when it is time to locate the user again. Some researchers [29] directly select the method that consumes the lowest energy. However, it is not always the best. Generally, the lower energy a method consumes, the less accurate it will be. For example, it is known that GPS is highly accurate compared with other localization methods, while its energy consumption is extremely high. If it is used to locate a user, it will take shorter time for the user to move beyond the application limited range. In contrast, it will take longer time. Let E_{low} denote the energy consumed by a low energy-consuming method, and E_{high} for a high energy-consuming method, and their time interval between two consecutive localization operations can be denoted with T_{low} and T_{high} respectively. If two applications using the two methods run at the same time, the total energy consumptions, E_{low}^1 , and E_{high}^2 can be calculated as follows.

$$E_{low}^1 = \frac{E_{low}}{T_{low}} \times T \tag{2}$$

$$E_{high}^2 = \frac{E_{high}}{T_{high}} \times T \tag{3}$$

It is obvious that if $E_{high}/T_{high} < E_{low}/T_{low}$, then $E_{high}^2 < E_{low}^1$. Therefore, based on the above theory, we present our selection method using Equation (4), where GPS, WiFi, GSM positioning methods are respectively represented by symbols g , w , and m .

$$M_t = \arg \min_{g,w,m} \left\{ \frac{E_{M_t}}{(a_t - a_{M_t})/v_e(t)} \right\} \tag{4}$$

M_t is selected from the set of localization methods $\{g,w,m\}$ at time t , which is the method that produces the minimum ratio of E/T .

4. Performance Evaluation

In this section, we provide a simulation study of the energy-efficient strategy to assess the performance of the proposed algorithm. The following two metrics are considered in our simulations.

Total energy consumption: This is the total energy consumed by LBAs during the period that application is running.

Satisfaction degree: This is the ratio of the time that the evaluated location satisfies the application accuracy requirement to the total time that application is running.

4.1. Settings

To evaluate the performance of our strategy, the following mobile pattern is considered.

The time unit is defined as one second, and the minimum and maximum velocity are set to $1m/s$ and $10m/s$ respectively for a mobile user. Let $v(t)$ denote the velocity of an user at time t . We further assume that the acceleration changes every T_1 seconds, detailed in Equation (5).

$$\text{if } t \bmod T_1 = 0 \text{ then } a(t) = \begin{cases} 0 \text{ or } 1 & \text{if } v(t-1) = 1m/s \\ -1 \text{ or } 0 & \text{if } v(t-1) = 10m/s \\ -1, 0 \text{ or } 1 & \text{otherwise} \end{cases} \quad (5)$$

where $a(t)$ denotes the acceleration at time t . When t is equal to the integer times of T_1 , a new acceleration is generated at once. When $v(t-1)$ is equal to the minimum velocity $1m/s$, the acceleration is equal to either 0 or 1. Likewise, when $v(t-1)$ is equal to the maximum velocity $10m/s$, the acceleration is equal to either 0 or -1. In other cases, the acceleration is equal to one of the set $\{-1, 0, 1\}$ randomly. The velocity $v(t)$ is computed by Equation (6).

$$v(t) = \begin{cases} v(t-1) + a(t) & \text{if } t \bmod T_1 = 0 \\ v(t-1) & \text{if } t \bmod T_1 \neq 0 \end{cases} \quad (6)$$

The simulation runs 3600s. The application accuracy requirement is changed every 600s as 500m, 300m, 150m, 120m, 80m, and 50m. For GPS, WiFi, and GSM, using the data from previous work [26], accuracy is 10m, 50m, and 150m, and energy consumption is 1425mJ, 545mJ, and 20mJ, respectively. The acceleration generates again every 3s, using Equation (5).

4.2. Result Analysis

Using the mobile pattern above, we evaluated our strategy, as shown in Figures 2, 3, 4, and 5.

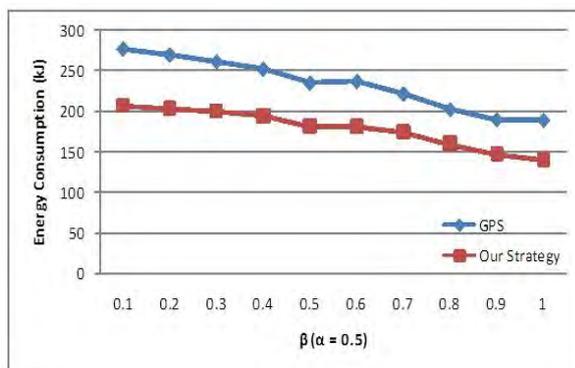


Fig. 2. Energy Consumption of GPS versus our strategy ($\alpha=0.5$)

Figures 2 and 3 depict the total energy consumption and the satisfaction degree of our strategy and GPS, respectively, when α used in EWMA method is set to 0.5 and the value of β ranges from 0.1 to 1.0. We observed that as β increases, the total energy consumption is reduced, as shown in Figure 2, but the satisfaction degree is also lowered, as shown in Figure 3, for both our strategy and GPS. The reason is that, with the increasing value of β , the interval of velocity sampling will also increase, then the localization operation after the user is beyond the application accuracy limit occurs repeatedly. In other words, the interval of two consecutive localizations is prolonged so that the frequency of localization operation is reduced, which leads to reduce localization energy consumption. At the same time, the satisfaction degree is lowered for the same reason. These results prove the effectiveness of our estimation method of the localization time point.

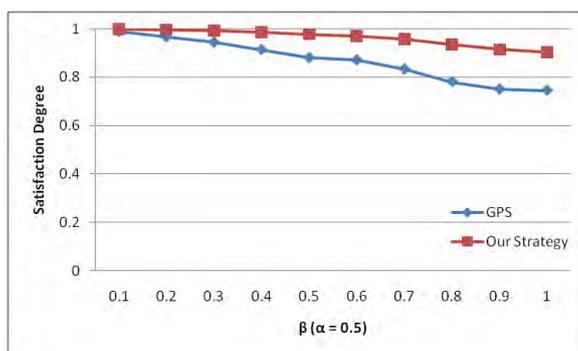


Fig. 3. Satisfaction Degree of GPS versus our strategy ($\alpha=0.5$)

Compared with GPS, our energy-efficient localization strategy conserves more energy. The reason is that our strategy is designed to select the energy-optimal method. Commonly, the selected method is less accurate than GPS, but the interval of two consecutive localizations is shorter than the one based

on GPS. As a result, the time point we estimate to locate the user again is more accurate and the corresponding satisfaction degree is better. This is proved by the simulation results.

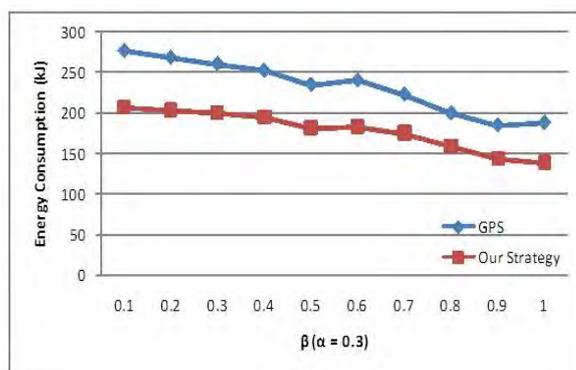


Fig. 4. Energy Consumption of GPS versus our strategy ($\alpha=0.3$)

Figures 4 and 5 show the simulation results when α is set to 0.3 and the value of β ranges from 0.1 to 1.0. They also proved the effectiveness of our strategy.

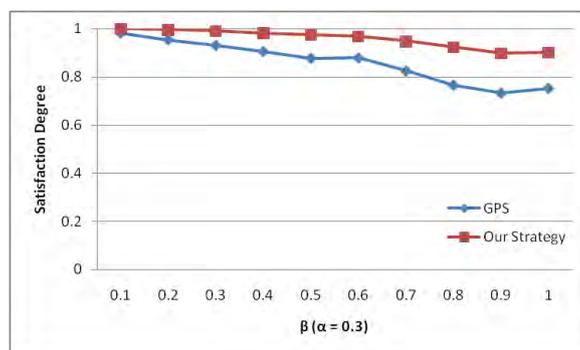


Fig. 5. Satisfaction Degree of GPS versus our strategy ($\alpha=0.3$)

As shown in the figures above, we found that the simulation results seem not to be affected by the value of α . This is because that the acceleration in our simulation is not changed significantly. In fact, the smaller α will be better when the acceleration increases as the user moves, because the previous velocity is more useful to estimate movement distance. Otherwise, the larger α will be better. For example, assuming the time interval is one second, and there are four velocities $\{2,4,8,16\}$ in four intervals. Therefore, the acceleration is $\{0,2,4,8\}$, implying that the acceleration becomes larger and larger. The moving distance is $2+4+8+16=30$. We sample two velocities at the starting time and the ending time, which are $1m/s$ and $4m/s$. The estimated moving

Haifeng Liu, Feng Xia, Zhuo Yang, and Yang Cao

distance is $(2 \times 0.5 + 16 \times 0.5) \times 4 = 36$ when α is equal to 0.5, $(2 \times 0.7 + 16 \times 0.3) \times 4 = 24.8$ when α is equal to 0.3, and $(2 \times 0.9 + 16 \times 0.1) \times 4 = 13.6$ when α is equal to 0.1. As α is set to an appropriate smaller value, the estimated velocity and moving distance are more effective and closer to the real values.

5. Conclusions

In this paper, we presented an energy-efficient localization strategy for smartphone applications. It is designed to conserve energy mainly in the two respects below. First, it dynamically determines the next localization time point and sample velocity. This is intended to reduce energy consumption by avoiding unnecessary localization operations. Second, it selects the energy-optimal method to help reduce energy consumption when the estimated localization time point comes. We have evaluated the performance of the proposed strategy through a series of simulations. The results show that it can significantly reduce energy consumption and improve satisfaction degree. One of our future works is to apply some other advanced techniques, Bluetooth for example, to achieve further improvement. Another is to consider how to collaborate on localization with users in the vicinity.

Acknowledgement. This work was partially supported by the Natural Science Foundation of China under Grant No. 60903153, Zhejiang Provincial Natural Science Foundation of China under Grant No. Y108685, the Fundamental Research Funds for Central Universities (DUT10ZD110), the SRF for ROCS, SEM, and DUT Graduate School (JP201006).

References

1. B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J. Herrera, A. Bayen, M. Annavaram, and Q. Jacobson. Virtual Trip Lines for Distributed Privacy-Preserving Traffic Monitoring. In Proceedings of ACM MobiSys'08, Breckenridge, CO, USA. (2008)
2. P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: Rich Monitoring of Road and Traffic Conditions Using Mobile Smartphones. In Proceedings of ACM SenSys'08, Raleigh, NC, USA, 323-336. (2008)
3. J. Yoon, B. Noble, and M. Liu. Surface Street Traffic Estimation. In Proceedings of ACM MobiSys'07, San Juan, Puerto Rico, 220-232. (2007)
4. H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of Wireless Indoor Positioning Techniques and Systems. IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., Vol. 37, No. 6, 1067-1080. (2007)
5. Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh. A Framework of Energy Efficient Mobile Sensing for Automatic User State Recognition. In Proceedings of 7th International Conference on Mobile Systems, Applications, and Service (MobiSys'09), 179-192. (2009)

6. J. Paek, J. Kim, and R. Govindan. Energy-Efficient Rate-Adaptive GPS-based Positioning for Smartphones. In proceedings of 8th International Conference on Mobile Systems, Application, and Services (MobiSys'10). (2010)
7. F. B. Abdesslem, A. Phillips, and T. Henderson. Less is More: Energy-Efficient Mobile Sensing with SenseLess. In MobiHeld'09: Proceedings of the 1st ACM SIGCOMM Workshop on Networking, Systems, and Applications for Mobile Handhelds. ACM, 61-62. (2009)
8. P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-based User Location and Tracking System. In INFOCOM, Vol. 2, 775-784. (2000)
9. A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place Lab: Device Positioning Using Radio Beacons in the Wild. In Proceedings of the Third International Conference on Pervasive Computing. (2005)
10. G. Sun, J. Chen, W. Guo, and K. J. R. Liu. Signal Processing Techniques in Network-Aided Positioning: A Survey of State-of-The-Art Positioning Designs. IEEE Signal Processing Magazine, Vol. 22, No. 4, 12-23. (2005)
11. A. Küpper and G. Treu. Efficient Proximity and Separation Detection among Mobile Targets for Supporting Location-based Community Services. ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 10, No. 3, 1-12. (2006)
12. T. Brezmes, J.-L. Gorricho, and J. Cotrina. Activity Recognition from Accelerometer Data on A Mobile Phone. In Proceedings of IWANN'09, Salamanca, Spain. (2009)
13. N. Györfi, Á. Fábrián and G. Hományi. An Activity Recognition System for Mobile Phones. Mob. Netw. Appl., Vol. 14, No. 1, 82-91. (2009)
14. V. Otsason, A. Varshavsky, A. LaMarca and E. de Lara. Accurate GSM Indoor Localization. UbiComp 2005, Lecture Notes Computer Science, Vol. 3660. Springer-Verlag, Berlin Heidelberg, 141-158. (2005)
15. A. U. Khan and M. Al-Akaidi. A Distributive Algorithm for WLAN Localization. In proceedings of 6th IEEE International Conference on Emerging Technologies. (2010)
16. P. Prasithsangaree, P. Krishnamurthy, and P. K. Chrysanthis. On Indoor Position with Wireless LANs. In Proc. IEEE Int. Symp. Pers. Indoor, Mobile Radio Commun., Vol. 2, 720-724. (2002)
17. A. Kotanen, M. Hännikäinen, H. Leppäkoski, and T. D. Hämäläinen. Experiments on Local Positioning with Bluetooth. In Proc. IEEE Int. Conf. Inf. Technol.: Comput. Commun., 297-303. (2003)
18. J. Hallberg, M. Nilsson, and K. Synnes. Positioning with Bluetooth. In Proc. IEEE 10th Int. Conf. Telecommun., Vol. 2, 954-958. (2003)
19. M. Kuhn, C. Zhang, B. Merkl, D. Yang, Y. Wang, M. Mahfouz and A. Fathy. High Accuracy UWB Localization in Dense Indoor Environments. IEEE Int Conf UWB, Vol. 2, 129-132. (2008)
20. R. Zetik, J. Sachs, and R. Thoma. UWB Localization-Active and Passive Approach. In Proc. 21st IEEE IMTC, Vol. 2, 1005-1009. (2004)
21. M. Bouet and A. L. dos Santos. RFID tags: Positioning Principles and Localization Techniques. Wireless Day 2008 (WD08), 1-5. (2008)
22. J. Zhou and J. Shi. RFID Localization Algorithms and Applications-A Review. Journal of Intelligent Manufacturing, Vol. 20, No. 6, 695-707. (2009)
23. Z. Zhuang, K.-H. Kim, and J. P. Singh. Improve Energy Efficiency of Location Sensing on Smartphones. In proceedings of 8th International Conference on Mobile Systems, Application, and Services (MobiSys'10). (2010)

Haifeng Liu, Feng Xia, Zhuo Yang, and Yang Cao

24. M. B. Kjærgaard. Minimizing the Power Consumption of Location-based Services on Mobile Phones. *IEEE Pervasive Computing*. (2010)
25. C.-W. You, P. Huang, H. Chu, Y.-C.Chen, J.-R. Chiang and S.-Y. Lau. Impact of Sensor-Enhanced Mobility Prediction on the Design of Energy-Efficient Localization. *Elsevier Ad Hoc Networks*, Vol. 6, No. 8, 1221-1237. (2008)
26. M. B. Kjærgaard, J. Langdal, T. Godsk, and T. Toftkjær. EnTracked: Energy-Efficient Robust Position Tracking for Mobile Devices. In *Proceedings of 7th International Conference on Mobile System, Applications, and Services (MobiSys)*. (2009)
27. T. Farrell, R. Cheng, and K. Rothermel. Energy-Efficient Monitoring of Mobile Objects with Uncertainty-Aware Tolerances. In *IDEAS'07: Proceedings of the 11th International Database Engineering and Applications Symposium*, 129-140. (2007)
28. I. Constandache, S. Gaonkar, M. Sayler, R. R. Choudhury, and L. Cox. Enloc: Energy-Efficient Localization for Mobile Phones. In *INFOCOM Miniconference'09*. (2009)
29. K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao. Energy-Accuracy Aware Localization for Mobile Devices. In *proceedings of 8th International Conference on Mobile Systems, Application, and Services (MobiSys'10)*. (2010)

Haifeng Liu received B.E. degree from Dalian University of Technology, China, in 2009. Currently he is a Master student in School of Software, Dalian University of Technology. His research interests include mobile computing and green computing.

Feng Xia is an Associate Professor and Ph.D Supervisor in School of Software, Dalian University of Technology, China. He is the (Guest) Editor of several international journals. He serves as General Chair, PC Chair, Workshop Chair, Publicity Chair, or PC Member of a number of conferences. Dr. Xia has authored/co-authored one book and over 100 papers. His research interests include cyber-physical systems, mobile and social computing, and intelligent systems. He is a member of IEEE and ACM.

Zhuo Yang received B.E. degree from China Jiliang University in 2004 and MSc degree from Nanyang Technological University, Singapore in 2007. She has been a Engineer in School of Software, Dalian University of Technology, China, since 2009. Her research interests include mobile computing, software test and quality assurance.

Yang Cao is a senior undergraduate in School of Software, Dalian University of Technology, China. Her research interests include wireless networking and mobile computing.

Received: April 30, 2011; Accepted: June 22, 2011.

Modeling Disease Spreading on Complex Networks

Xiangjie Kong¹, Yu Qi¹, Xiumiao Song¹, and Guojiang Shen^{2,3}

¹ School of Software, Dalian University of Technology,
Dalian 116620, China

xjkong@dlut.edu.cn, qiyu1988202@sohu.com, songxiumiao1987@yahoo.cn

² State Key Laboratory of Industrial Control Technology, Zhejiang University,
Hangzhou 310027, China
gjshen1975@126.com

³ Key Laboratory of System Control and Information Processing,
Ministry of Education, Shanghai 200240, China

Abstract. Based on complex network approach, a contact network model with scale-free property is built. By analyzing the fact data of H1N1 influenza provided by Beijing Health Bureau, contact tracing mechanism is used to research H1N1 virus transmission dynamics with this model. Furthermore, the contact tracing coefficient and random checking coefficient are studied to analyze their impact on the peak value of new infections and cumulative number of infections. The simulation result fits well with the statistical data. It shows that the model built in this paper is valid and complex network model to simulate the epidemics of H1N1 influenza is feasible.

Keywords: complex network, disease spreading, scale-free, contact network, contact tracing.

1. Introduction

Network widely exists in nature and human society. If a brain neuron is seen as a node in the network, the connections between neurons are the edges, which constitute a neural network. If a power station is seen as a node in the network, the power wires between power stations are edges, which constitute a power grid. If a person is seen as a node in the network, the interaction between people is an edge, which constitutes a human social network [1], [2]. There are many similar instances. With the development of research, we will find more networks in various fields.

A complex network is an abstract of a real complex system, which is described by some nodes and some edges. Nodes represent individual object of complex system, such as neuron, computer, people, etc. Edges represent the links between these individuals. Due to the topology of the network tends to affect performance of the network, for example, electricity grid structure

influencing efficiency of power transmission, we can research the structure characteristics of the network to understand some characteristics of the complex system in the real world.

Based on the topology structure of complex network dynamics in the complex network approach, there are some quantitative and qualitative researches on the dynamic spread such as the spread of computer virus in the Internet, the spread of disease in the crowd, the diffusion of information in society. In traditional epidemiology, SI, SIS, SIR and other classic models have been established [3], [4], [5], [6]. S is the at-risk group, which doesn't carry virus, but is easily infected. I is the infected group with infectious feature. R is the cured individuals with immunity. These models based on differential dynamic system, have complex computations and the equation solution is very sensitive to initial conditions, which can't be well simulated with the practical process for some unexpected and random events. However, the simulation contagion transmission based on the complex network can overcome the shortcomings.

In the past two years, the H1N1 virus has spread worldwide, influencing people's normal life. In the meantime, various strategies and effects dealing with virus spreading also cause for concern. In this paper, we establish contact network model based on the complex network theory, and adopt contact tracing mechanism to do the numerical study on the dynamic spread behaviors of H1N1 virus. We deeply analysis the effect of contact tracing coefficient, random testing coefficient in virus spreading, and reveal the importance of early detection, early isolation and early treatment in process of the disease spread.

The paper is organized as follows. In Section 2, we present the related work on complex networks. Section 3 proposes the related work on contact network model respectively. We present H1N1 propagation model in Section 4. Following that, we give and analyze the simulation result in Section 5 and Section 6 respectively. Finally, some conclusions are drawn in Section 7.

2. Related Work

There are various research problems with the complicated real-world systems. Some complex systems can be described in the form of the networks. In 1960, Erdos and Renyi proposed ER stochastic network model [7], in which the network structure is described as a completely random graph. In stochastic network model, the connection between two nodes is determined in a certain probability. However, with further research, there are many complex networks appearing, which cannot be described by the random model.

In 1998, Watts and Strogatz proposed the Small-World network model [8]. Small-World network is between regulation network and stochastic network. In a ring regulation network, an arbitrary edge between adjacent nodes constantly reconnects with other nodes according to certain probability.

Finally it can constitute a Small-World network. When $P = 0$, it is a regular network. When $P = 1$, it is a stochastic network. When $0 < P < 1$, it is a new network with highly cluster character of the regular network and less average path length of the stochastic network. In the complex network theory, a network with the two properties is called Small-World characteristics.

In 1999, Barabasi and Albert found scale-free characteristic and put forward BA scale-free network model [9]. It has the following two characteristics: (1) growth characteristics: network scale is expanding; (2) priority connection characteristics: new nodes tend to connect the nodes with large connection degree, which is called "Matthew effect". In scale-free networks, where node degree distribution obeys power-law [9], most of the nodes have only a few links and few nodes have a lot of links.

Node degree of stochastic network and Small-World network obeys Poisson distribution. The distribution is bell-shaped, and the peak value just corresponds with the average degree value of all the nodes. In both sides of peak value, distribution probability obeys exponential decline, which indicates mostly node degree distribution concentrates near average degree value. Therefore, this type of network called homogeneous network. Scale-free network's node degree distribution has the characteristics of power-law, namely $P(k) = ck^{-\gamma}$. There is no peak value in the degree distribution graph. There is a descending line showing the scale-free characteristics in the bi-logarithm coordinates. Therefore, scale-free network is inhomogeneous network.

Since the complex network research with small-world networks and scale-free network appear, many scholars have adopted complex networks to study spread of diseases. Watts and Strogatz simulated spread of disease model and found that disease spreading in small world network is faster and easier than in regulation network [8]. Newman and Watts thoroughly discussed disease spreading problem in social network and proposed an improved Small-World model namely NW model [10]. Pastor-Satorras and Vespignani studied the infinite scale-free network SIS model, and they were surprised to find the propagation threshold doesn't exist [11]. That indicated disease could exist everlastingly in scale-free network, even though disease spreading strength was very small. The conclusion fundamentally changed many disease spreading conclusions of traditional theory. When Moreno and others analyzed SIR model, they also found there are no propagation threshold and a fraction of infected individuals existing in the network perpetually [12]. It can also be observed in SIS model, and it well explained the actual situations that a fraction of the virus exists in the network for a long time. Lin and others studied the spread of the SARS virus with Small-World network model [13]. Yang and others studied the spread mechanism of bird flu based on the complex network approach [14].

3. Contact Network Model

Infectious diseases often transmit through interpersonal interaction. If people are seen as the nodes in the network and the interpersonal contact relation as the edges in the network, it constitutes a complex network called contact network [15], [16]. Because of the individual differences, the scope of individual communication is different. Therefore, the node degree in contact network distributes within the larger scope, rather than the uniform network having the same degree of network, whose node degree concentrates in peak value nearby. The contact network is more close to scale-free network.

The contact network model in this paper evolves from the BA scale-free network model proposed by Barabasi and Albert. First, we construct a standard BA scale-free network based on growth mechanism and priority connection mechanism. Then, all nodes are put in the scale-free network into 2-d lattice bitmap $N = n \times n$ randomly, in which each node links with 8 adjacent nodes [17].

It can be proved that contact network is a scale-free network. When the number of nodes $N \gg 8$, the probability of existing edge among 8 neighboring nodes tends to zero, namely $p \rightarrow 0$. That is equivalent to in the original BA network, each node is added 8 edges. The degree of each node is changing, but it does not affect the degree distribution of the whole network. So the improved contact network degree distribution obeys power-law. As shown in Fig. 1, the degree distribution of contact network in bi-logarithm coordinates is basically a straight line with power-law features.

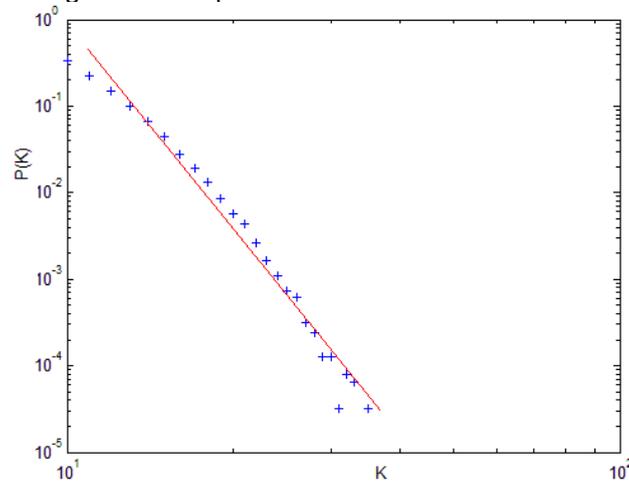


Fig. 1. Contact network node degree probability distribution ($N = 250 \times 250$).

Improved contact network model increases the node degree, thus increasing the chance of contact infection, especially in small degree nodes. In the improved model, each node is connected not only with neighboring nodes, but also a number of distant nodes, which increases the reach of the

individual to increase the speed of spread of the virus, more in line with the great personnel mobility features in the real social situation.

4. H1N1 Virus Propagation Model

Influenza A H1N1 is an acute respiratory infection, whose pathogen is a new type of influenza A H1N1 influenza virus. The virus strains contain swine flu, avian flu and human influenza virus gene fragments, spreading through contacts in the crowd. Influenza A H1N1 flu was first discovered in March 2009. The "swine flu" epidemic outbreak in Mexico and quickly spread around the globe, bringing people all over the world huge economic loss.

To make the model not too complex, the following are some simplifying assumptions for the spread model of H1N1 virus:

(1). Assuming the transmission capacity of each patient is same, not considering super-spread events. This is because researches have shown that super-spread events (SSEs) which can't be completely ignored will not change the overall trend of disease spreading.

(2). Assuming the activity of the virus is constant during the evolution, taking no account of the variability behavior of the virus, ignoring the ventilation, temperature and other factors impacting H1N1 virus activity.

(3). Assuming each individual's immunity is same, not affected by age structure, ignoring the different immune system of different individuals because the model are concerned with statistical properties of the overall performance rather than specific individuals.

(4). Assuming the incubation period for each patient obeys Poisson distribution, and the incubation period has a smaller infectious probability.

(5). Assuming the patient can't infect other individuals with isolated protection measure.

(6). Assuming the transmission capacity of a patient is not changing during the time from infection to cure.

This paper adopts Contact Tracing strategy to restrain the transmission of H1N1 virus [6], [18]. Once detect an infected individual, track and isolate contacted individuals immediately. Isolated individuals no longer infect other individuals. In the H1N1 propagation model, the nodes have 5 states:

Susceptible: The individual is not infected, and has no immunity.

Latent: The individual carries virus, and shows a slight symptoms and small infectiousness.

Infected: There is an outbreak with clinical manifestation and strong infection.

Tracking: The individual is treated in isolation. Track and isolate other individuals contacted with the patient.

Removed: The individual obtains immunity or dies, and no longer affects the communication process.

Fig. 2 describes the relationship between the states. Randomly select a node N_i , marked the state I as the infection source. The node in state I

infects the neighboring node S in probability α_I . The node in state of L infects the neighboring node S in probability α_L ($\alpha_I > \alpha_L$). The nodes in the state of I and L are respectively randomly detected in probability β_I, β_L ($\beta_I > \beta_L$) entering state T. Then contact and trace the adjacent nodes of the randomly detected node in probability ε . The node which has been traced enters the state of T. Nodes in state T will enter state R in probability δ .

Incubation period of H1N1 is generally 1 to 7 days [19], expectation $E(Latent) \approx 4$. Poisson distribution is a commonly used discrete probability distribution, suitable for describing the number of random events in the unit time, and the expectation of Poisson distribution $P(\lambda)$ is λ . Therefore, in the model make the latency of individual obeys Poisson distribution $\lambda = 4$, that is $Latent \sim P(\lambda = 4)$.

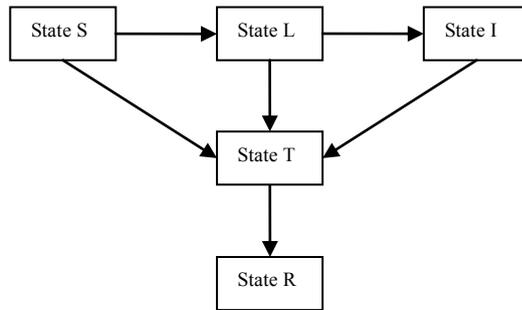


Fig. 2. H1N1 propagation model state transition graph.

5. Simulation and Analysis

Table 1. Increasing H1N1 influenza confirmed cases per week in Beijing.

Time (week)	Cases						
1st	1	13th	80	25th	1063	37th	39
2nd	3	14th	82	26th	289	38th	37
3rd	4	15th	82	27th	587	39th	12
4th	11	16th	81	28th	551	40th	7
5th	28	17th	138	29th	521	41th	9
6th	34	18th	294	30th	501	42th	4
7th	44	19th	589	31th	419	43th	5
8th	84	20th	844	32th	306	44th	3
9th	56	21th	1124	33th	240	45th	4
10th	61	22th	418	34th	158	46th	0
11th	92	23th	659	35th	82	47th	1
12th	100	24th	1287	36th	53		

As Table.1 shows, this paper refers to H1N1 epidemic data released by Beijing Health Bureau from 2009.5.7 to 2010.4.4. We adopt the contact network model and the H1N1 virus transmission model to predict and analysis the number of new infections and the cumulative number of infections per week (Fig. 3, Fig.4).

We use optimized algorithm to simulate in the programming process based on Matlab. Optimized algorithm can greatly shorten the simulation time and improve simulation efficiency. During simulation process, the simulation algorithm was improved as follows:

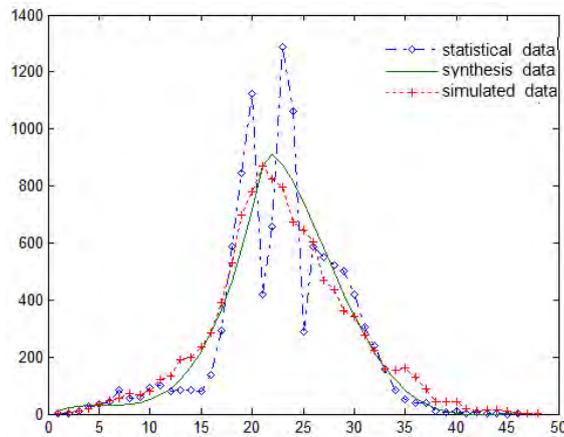


Fig. 3. The number of new infections per week.

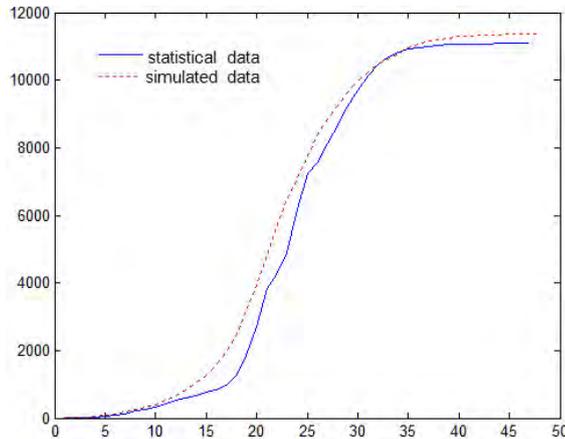


Fig. 4. The number of cumulative infections per week.

(1) When storing network topology, based on the idea of sparse matrix, store corresponding node of the each edge in the vector I, J , so that save

memory space and facilitate the establishment of large-scale networks. Using adjacent matrix directly will waste a lot of memory space. If adopting the sparse matrix $A = \text{sparse}(N, N)$, when A is large, the assignment statement $A(i,j)=...$ has slow speed of execution [20] and a large number of execution cause the entire simulation process consuming a large amount of time.

(2) When finding all the adjacent nodes of the node N_i , we firstly sort I , and then exchange the order of J in the same order. Use binary search algorithm to find I , while the corresponding element in J is adjacent node of N_i . The find() function in Matlab uses the linear search algorithm. When the network size is large, searching efficiency is low.

(3) Store I, J in a Mat file, just loading them as being used. It doesn't need to re-generate network topology map in each simulation to save time.

In the simulation process, the number of established contact network nodes $N=250 \times 250$, parameter values are: $\alpha_I = 0.39$, $\alpha_L = 0.15$, $\beta_I = 0.63$, $\beta_L = 0.42$, $\varepsilon = 0.9$, $\delta = 0.1$. The simulation results are shown in Fig. 3 and Fig.4. During 20th to 25th week, the number of new infections per week reaches the peak value. The number of infections at both sides of the peak value obeys an exponential distribution, and the growth rate of the left side is slightly larger than the decay rate of the right side. In the early and late period, the spread rate of influenza A H1N1 flattens off (Fig. 3). During the peak-time of the spread of the disease (20th to 25th week), the simulation data and statistical data fit well. Overall, the simulation results agree well with the trend of the actual propagation of disease. The simulation data of cumulative infections number is slightly larger than statistical data, and both growth trends are almost the same (Fig. 4).

Accuracy of the simulation results can be calculated by curve's goodness of fitting. And the squares sum of error can be used to measure the quality of the curve's goodness of fitting. Assuming that the actual measured value is Y , the average value is Y_1 , the calculated theoretical value according to the fitting curve is Y_2 , we can get the squares sum of error is $\sum(Y - Y_2)^2$, mean square variance is $\sum(Y - Y_1)^2$, if the ratios of the squares sum of error and mean square variance are small, it means that the observed values and estimated values are close, also the curve fitting is well. According to this we can define the determination coefficient R^2 , which calculation formula is:

$$R^2 = 1 - \frac{\sum(Y - Y_2)^2}{\sum(Y - Y_1)^2}$$

Because the statistics has strong fluctuation, statistics data and real data may be quite different. Just like the statistical data in Fig. 3, there's strong fluctuation in the peak area, thus the response can not reflect the fact. Therefore, we firstly fit the statistical data (as green curve in Fig. 3 shows), then compare the simulation data and the fitted data to get the simulation accuracy. In Fig. 3, the simulation accuracy of the number of new infections

per week is $R^2 = 0.9667$. In Fig. 4, the actual statistical value is 11087, the simulation data is 11376, the number of cumulative infections' error is $(11376 - 11087) / 11087 = 0.0261$, and the simulation accuracy of the number of cumulative infections per week is $R^2 = 0.9853$. The high accuracy of the number of new infections and number of cumulative infections shows that the model can well describe the propagation of the virus H1N1.

6. Parameters of Virus Propagation Model

There are many parameters impacting H1N1 virus spreading in the above H1N1 spreading model. We will analyze contact tracing coefficient ε and random detection coefficient β_L deeply to reveal their influence in speed and level of virus spreading.

6.1. Contact tracing coefficient

During simulation process, we remain the other parameters same, changing the contact tracing coefficient ε to get different values of the peak number of infections (showed in Fig. 5, Fig. 6, Fig. 7), from which to analyze the impact of ε on H1N1 virus spreading. Fig. 5 and Fig. 6 show that contact tracing coefficient has a significant impact on the peak number of infections. When ε is small, the peak density of infected people is large. Otherwise, peak density of infected population is small.

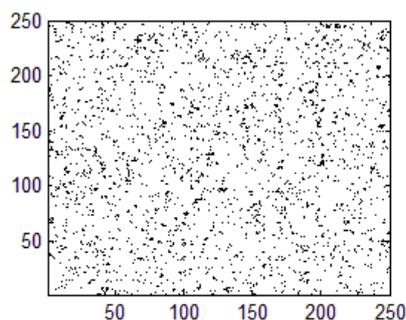


Fig. 5. Distribution the peak number of infected people when $\varepsilon = 0.4$.

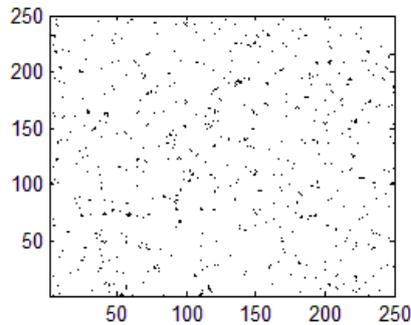


Fig. 6. Distribution the peak number of infected people when $\varepsilon = 0.8$.

Fig. 7 further reveals the relationship between contact tracing coefficient and the peak number of infections: the peak number of infections has a logarithmic decreasing with ε increasing. Thus, contact tracing coefficient strongly impacts the processing of H1N1 virus spreading. When the value of ε is large, most of the close contacts can be isolated in time to reduce the probability of individuals to be infected.

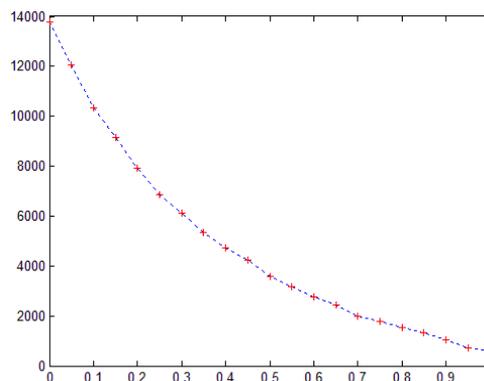


Fig. 7. Distribution the peak number of infected people according to different ε .

6.2. Random detection coefficient

In H1N1 spreading model, β_I and β_L are the same type of parameters having the similar impacts on the spread of the virus, so this article only deeply discusses β_L . Keeping other parameters unchanged to analyze different values of random detection coefficients β_L related to the number of cumulative infections. It can be seen from Fig. 8, the number of cumulative infections increases linearly with the decrease of β_L . When the value of β_L is

high, most individuals in the incubation period can be promptly detected and isolated, which reduces the probability of people around to be infected. Therefore, it will slow the rate of virus spreading and reduce the scope of the virus spreading, thus the number of cumulative infections will be reduced.

It is obvious that random detection coefficient corresponds to testing strength of test stations, and contact tracing coefficient corresponds to tracing strength of closing contact individuals in the actual situation. Therefore, when facing the widespread of virus, we should strengthen the efforts of the detection of vulnerable populations, increase the strength of tracing close contacts, and reach the target of early discovery, early isolation and early treatment. This can effectively inhibit the spread of the virus, slow down the rate of virus spreading, and reduce the virus' impacted area.

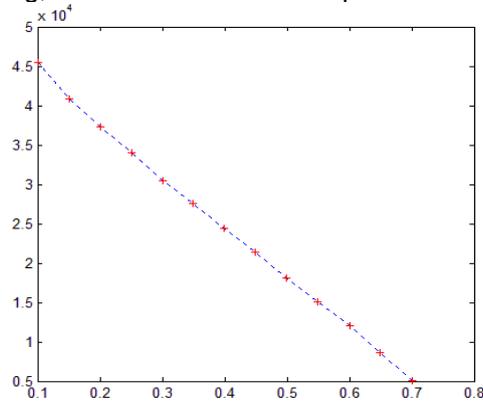


Fig. 8 The number of cumulative infections according to different β_L .

7. Conclusions

In this paper, we use the dynamics of complex networks approach to study virus spreading. We build scale-free network model (contact network) and H1N1 virus propagation model according to the actual situation of the virus propagation in order to simulate the spread of the H1N1 virus in Beijing. Then we deeply analyze how contact tracing coefficient and random detection coefficient impact virus spreading, revealing the importance of early discovery, early isolation and early treatment during treatment in the disease spreading process based on the virus spreading model. Simulation results are similar with the actual spread trend of the virus in Beijing, which verify the validity of the model, proving that complex networks as an important tool will play a major role on study of disease spreading.

Acknowledgement. This work was supported by Natural Science Foundation of China under grant 60903153 & 61174174, the Fundamental Research Funds for the Central Universities (DUT1600-852006), Zhejiang Province Natural Science

Xiangjie Kong, Yu Qi, Xiumiao Song, and Guojiang Shen

Foundation of China under grant Y1090208 and the Foundation of Key Laboratory of System Control and Information Processing, Ministry of Education, P.R. China under grant SCIP2011007.

References

1. Albert, R., Barabási, A.: Statistical Mechanics of Complex Networks. Rev. Modern Phys. (S0034-6861), Vol. 74, No. 1, 47-97. (2002)
2. Newman, M. J.: The Structure and Function of Complex Network. SIAM Review (S0036-1445), Vol. 45, No. 2, 167-256. (2003)
3. Breban, R., Vardavas, R., Blower, S.: Linking population-level models with growing networks: A class of epidemic models. Phys. Rev. E, 72, 046110. (2005)
4. Trpevski, D., Tang, W. K. S., Kocarev, L.: Model for rumor spreading over networks. Phys. Rev. E, 81, 056102. (2010)
5. Kenah, E., Robins, J. M.: Second look at the spread of epidemics on networks. Phys. Rev. E, 76, 036113. (2007)
6. Keeling, M. J., Eames, K. T. D.: Networks and epidemic models. J.R.Soc. Interface, 2:295-307. (2005)
7. Erdos, P., Renyi, A.: On the Evolution of Random Graphs. Publ. Math. Inst. Hung. Acad. Sci, 5:17-60. (1960)
8. Watts, D., Strogatz, S.: Collective Dynamics of "Small World" Networks. Nature (S0028-0836), 393(4): 440-442. (1998)
9. Barabási, A., Albert, R.: Emergence of Scaling in Random Networks. Science (S0036-8075), 286(5439): 509-512. (1999)
10. Newman, M. E. J., Watts, D. J.: Scaling and Percolation in the Small-World Network Model. Phys.Rev.E, Vol. 60, No. 6,7332-7342. (1999)
11. Pastor-Satorras, R., Vespignani, A.: Epidemic Spreading in Scale-Free Networks. Phys. Rev. Lett. (S0031-9007), Vol. 86, No. 14, 3200- 3203. (2001)
12. Moreno, Y., Pastor-Satorras, R., Vespignani, A.: Epidemic Outbreaks in Complex Heterogeneous Networks. Eur. Phys. J. B. Vol. 26, No. 4, 521-529. (2002)
13. Lin, G., Jia, X., Ou, Y.: Predict SARS infection with the small world network model. Journal of Peking University (Health Sciences), 35 Supplement:66-69. (2003)
14. Yang, H., Zhang, S.: Viruses Epidemics of Avian Influenza Based on Complex Networks. Journal of System Simulation, Vol. 20, No. 18, 5001 - 5005. (2008)
15. Yang, R., Zhou, T., Xie, Y., Lai, Y., Wang, B.: Optimal contact process on complex networks. Phys. Rev. E, 78, 066109. (2008)
16. Yang, H., Zhao, F., Li, Z., Hu, B.: Modeling SARS Spreading on Complex Networks, Phys. B, Vol. 18, No. 6, 2734-2739. (2004)
17. Yu, L., Xue, H.: Epidemic Spread Model Based On Complex Networks. Journal of Shanxi University of Science & Technology, Vol. 25, No. 3,126-129. (2007)
18. Huerta, R., Tsimring, L. S.: Contact tracing and epidemics control in social networks. Phys. Rev. E, Vol. 66, No. 4, 056115. (2002)
19. <http://www.moh.gov.cn/publicfiles/business/htmlfiles/mohwsyjbg/s9990/200910/43111.htm>
20. Shure, L.: Creating sparse finite-element matrices in matlab. March 1. (2007)

Xiangjie Kong received his Ph.D. degree from Zhejiang University, Hangzhou, China in 2009. Currently, he is an assistant professor in Dalian University of technology, Dalian, China. His current research interests include urban road traffic modeling and control technology, cyber physical systems and complex networks.

Yu Qi received the B.Sc. degree in software engineering in 2010 from Dalian University of technology, Dalian, China, where she is currently working toward the M. Sc. degree in the School of Software. Her current research interests include urban road traffic modeling and control technology, artificial intelligence, and complex networks.

Xiumiao Song received the B.Sc. degree in software engineering in 2011 from Dalian University of technology, Dalian, China, where she is currently working toward the M. Sc. degree in the School of Software. Her current research interests include artificial intelligence and complex networks.

Guojiang Shen received his Ph.D. degree from Zhejiang University, Hangzhou, China in 2004. Currently, he is an associate professor in Zhejiang University, Hangzhou, China. His current research interests include intelligent control theory and application, advanced control technology and application, urban road traffic modeling and control technology.

Received: March 12, 2011; Accepted: May 4, 2011.

An Improved Spectral Clustering Algorithm Based on Local Neighbors in Kernel Space¹

Xinyue Liu^{1,2}, Xing Yong² and Hongfei Lin¹

¹ School of Computer Science and Technology, Dalian University of Technology,
116024 Dalian, China

² School of Software, Dalian University of Technology,
116620 Dalian, China
xyliu@dlut.edu.cn

Abstract. Similarity matrix is critical to the performance of spectral clustering. Mercer kernels have become popular largely due to its successes in applying kernel methods such as kernel PCA. A novel spectral clustering method is proposed based on local neighborhood in kernel space (SC-LNK), which assumes that each data point can be linearly reconstructed from its neighbors. The SC-LNK algorithm tries to project the data to a feature space by the Mercer kernel, and then learn a sparse matrix using linear reconstruction as the similarity graph for spectral clustering. Experiments have been performed on synthetic and real world data sets and have shown that spectral clustering based on linear reconstruction in kernel space outperforms the conventional spectral clustering and the other two algorithms, especially in real world data sets.

Keywords: Spectral Clustering, Kernel Space, Local Neighbors, Linear Reconstruction.

1. Introduction

In recent years, spectral clustering has become one of the most popular clustering algorithms due to its high performance in data clustering and simplicity in implementation compared to the traditional clustering methods. It is based on the spectral analysis methods and solves the data clustering by graph partitioning problems [1], [2]. Spectral clustering algorithms consist of two steps: (1) construct a similarity graph with some kind of similarity function; (2) find an optimal partition of the graph and cluster the data points. The former which reflects the intrinsic structure of the data plays an important

¹ This paper is supported by “the Fundamental Research Funds for the Central Universities” and “the National Natural Science Foundation of China under Grant No. 61070016”.

role in spectral clustering [3], [4], thus a great deal of effort has been carried out to address it [5], [6].

Spectral clustering makes use of information achieved from an appropriately defined affinity matrix. Its primary strength is the ability to treat complex patterns where traditional methods (such as k -means) either cannot be applied, or fail. The similarity matrix should be built in such a way that reflects the topological characteristics of the data sets. The most commonly used similarity measure is Gaussian function which is defined as

$$G(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}, \text{ where } x_i \text{ and } x_j \text{ represent two point respectively [7].}$$

Although Gaussian function is simple to implement, the selection of the parameter σ is still an open issue. Generally it is non-trivial to find a good σ and spectral clustering is sensitive to the value of σ , especially in a multi-scale data sets. In practice, σ is often set by an empirical value, such as it is set as 0.05 of the maximal pair wise Euclidean distance among the data points in normalized cut algorithm [2]. What's more, sparsity is another desired property, since it can offers computational advantages [7].

In this paper, we focus on how to construct an optimal similarity graph. We propose a novel method to obtain the similarity matrix based on linear reconstruction with local neighbors in kernel space. The idea of linear reconstruction derives from the manifold learning which assumes data points are locally linear and each point can be reconstructed by a linear combination of its neighbors [8]. It has been applied to semi-supervised learning [9] and spectral clustering [10]. The kernel method has become popular largely due to many successes in applying kernel methods such as kernel PCA [11] and spectral regularization [6]. In our algorithm, we introduce the kernel methods for constructing the similarity matrix which increase the linear separability by mapping the data into high dimensional space. Experimental results on both artificial data sets and real world data sets indicate that spectral clustering based on local neighborhood in kernel space (SC-LNK) outperforms the traditional spectral clustering (SC), self-tuning spectral clustering (SSC) and locality spectral clustering (LSC).

The rest of this paper is organized as follows: some related work is discussed in Section 2. We begin with a brief overview of NJW method [12] in Section 3. Then introduce the method of linear reconstruction in Section 4 and mercer kernel in Section 5 on details of constructing the similarity matrix for our algorithm. In Section 6, we summarize our algorithm and experimental results are then presented in section 7. Finally we conclude the discussions and point out further works in the last section.

2. Related Work

An appropriate adjacency matrix is conducive to make a good partition for clustering algorithms. Much research has been conducted on the problem of

how to construct an optimal similarity matrix for spectral clustering [6], [13], [14]. Most existing works on constructing similarity matrix were built on Gaussian kernel function which is limited to its sensitive scale parameter and the real world data in some situation. Little attention has been paid to the other methods to measure the similarity in spectral clustering.

As mentioned above, the scale parameter of Gaussian function is very sensitive, especially in data sets with multiple scales. In order to deal with the problem, Manor et al. [13] proposed a new algorithm called self-tuning to use a local scale parameter for each data point, i.e., they use the distance of k -th nearest neighbor of each point as the scale parameter. The parameter k is set to 7 in their experiments. However, as have shown in our experiments, it fails on many real data sets although performs well on some synthetic data sets.

Gong et al. proposed a spectral clustering method [10] which called LSC algorithm. Instead of using the pairwise relationship, they considered the linear neighbors relationship based on the idea of local linear embedding in manifold learning [8]. With this novel relationship in hand, LSC achieved the similarity matrix which represents the local neighbors information well. In particular, the matrix obtained is sparse which offers computational advantages. However, the LSC method has some limitations in synthetic and real world data sets as shown in our experiments.

In [15], the linear Reconstruction in kernel space had been described. DeCoste et al analysis the local linear embedding technique in kernel space and applied to specific application. Through the introduction of kernel methods, it can increase the linear separability by mapping the data into high dimensional space. Therefore, the kernel trick method conduces to deal with the multiple and real data sets. But the authors only focus on the classification problem.

3. Review of Spectral Clustering

Give a set of data points $X = \{x_1, x_2, \dots, x_n\}$ with each point $x_i \in R^d$, then we can construct an undirected graph $G = (V, E)$ in which every vertex v_i represents the data point x_i . According to a similarity function, we obtain the similarity graph and let W be its weighted adjacency matrix. Note that $w_{ij} \geq 0$ and W is symmetric. Each edge $(i, j) \in E$ carries a weight w_{ij} which represents the similarity between the point x_i and x_j . We expect to find a good partition of the graph such that points with low similarities should be clustered into different groups. Many spectral clustering algorithms formalize this partitioning problem in different ways [2], [12], [14], [16-17]. In this paper, we adopt the normalized cuts (Ncut) [12] described as follows:

4. Analysis of Linear Reconstruction

The similarity graph plays an important role in a spectral clustering algorithm. Before, we need to define a similarity function on the data. In the common case, a reasonable candidate is the Gaussian similarity function, which is

defined as $G(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$. However, it indicates that there is no reliable way to choose the parameter σ when there are very few or even there are no labeled examples according to [18].

Instead of using pairwise relationship to construct the similarity graph in conventional spectral clustering, we propose to use the neighbor information of each point to construct the graph.

Algorithm 1: Normalized Spectral Clustering

Input: Data set X , number k of clusters

Output: A partitioning S_1, S_2, \dots, S_k

begin

Construct similarity matrix $W \in R^{n \times n}$

Define a diagonal matrix $D(i, j) = \sum_{j=1}^n W(i, j)$

Form the normalized Laplacian matrix $L = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$

Compute the k largest eigenvectors of L and

construct the matrix $U \in R^{n \times k}$ with the eigenvectors as its columns

Form matrix $Y \in R^{n \times k}$ from U by normalizing the rows to norm 1.

Group Y into clusters S_1, S_2, \dots, S_k with k -means

end.

For convenience of calculation, we assume the sample space is locally linear, that is, each point can be optimally reconstructed by a linear combination of its neighbors [8]. Then our objective is to minimize

$$\mathcal{E} = \left\| x_i - \sum_{j: x_j \in N(x_i)} w_{ij} x_j \right\|^2 \tag{1}$$

where $N(x_i)$ represents the neighbors of x_i , and w_{ij} is the contribution of x_j to x_i . According to [8], w_{ij} should satisfy the constraint $\sum_{j: x_j \in N(x_i)} w_{ij} = 1, w_{ij} \geq 0$. Intuitively, the more similar x_j to x_i , the larger w_{ij} would be. Thus we can use w_{ij} to measure the similarity between x_j and x_i . Then we can rewrite Eq. (1) as

$$\begin{aligned}
 \mathcal{E}_i &= \left\| x_i - \sum_{j: x_j \in N(x_i)} w_{ij} x_j \right\|^2 \\
 &= \left\| \sum_{j: x_j \in N(x_i)} w_{ij} (x_i - x_j) \right\|^2 \\
 &= \sum_{j, k: x_j, x_k \in N(x_i)} w_{ij} w_{ik} (x_i - x_j)^T (x_i - x_k) \\
 &= \sum_{j, k: x_j, x_k \in N(x_i)} w_{ij} C_{jk}^i w_{ik}
 \end{aligned} \tag{2}$$

where C_{jk}^i represents the (j, k) -th entry of the local Gram matrix at point x_i and $C_{jk}^i = (x_i - x_j)^T (x_i - x_k)$. Then we can obtain the reconstruction weights of each data point by solving the following n standard quadratic programming problems

$$\begin{aligned}
 \min_{w_{ij}} & \sum_{j, k: x_j, x_k \in N(x_i)} w_{ij} C_{jk}^i w_{ik} \\
 s.t. & \sum_{j: x_j \in N(x_i)} w_{ij} = 1, w_{ij} \geq 0
 \end{aligned} \tag{3}$$

Using the above method, we will construct a sparse matrix W . It is worth mentioning that we set the weights $W_{ij} = w_{ij}$ and $W_{ji} = w_{ij}$. If there are any overlap, the later weight will overlap the former one so that we can guarantee the symmetric of the reconstruction matrix W . Then the W can be treated as the similarity matrix.

5. Kernelized Linear Reconstruction

5.1. Mercer Kernel

Combining the kernel method can optimize the performance of clustering algorithm [19], [20]. Through the introduction of kernel methods, it can increase the linear separability by mapping the data into high dimensional space. DeCoste has analyzed the linear Reconstruction in kernel space [15]. Consider a $l \times D$ matrix X of data points. A mercer kernel $K(x_i, x_j)$ will projects two given points from D -dimensional original space into some (possibly infinite-dimensional) feature space and return their dot product in that feature space. That is

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) = \phi(x_i)' \phi(x_j) \tag{4}$$

where ϕ is some mapping function which need not explicitly to compute the coordinates of the projected vectors. Thus the kernel method can avoiding curse of dimensionality when explored large non-linear feature spaces.

One of the most popular mercer kernels is the radial basis function (RBF) kernel:

$$K(u, v) = e^{-\frac{\|u-v\|^2}{2\sigma^2}} \quad (5)$$

where the parameter σ controls a scale of the dot product of the two data points. We will use this kernel when mapping the input space to its feature space.

5.2. Linear Reconstruction in Kernel Space

In general, the selection of k nearest neighbors use the Euclidian distance in original space. In this paper, we use the distance based on Mercer kernels. The distance between data point x_i and x_j in kernel space is defined as:

$$d_{ij} = dist(\phi(x_i), \phi(x_j)) = \sqrt{(\|\phi(x_i) - \phi(x_j)\|^2)} \quad (6)$$

Through Mercer kernel, distances can be calculated directly from kernel value as follows

$$d_{ij} = \sqrt{K_{ii} - 2K_{ij} + K_{jj}} \quad (7)$$

For each $\phi(x_i)$ in kernel space, denote its k nearest neighbors in kernel space as $N(\phi(x_i))$, then the kernelized reconstruction error is:

$$\varepsilon = \|\phi(x_i) - \sum_{j: \phi(x_j) \in N(\phi(x_i))} w_{ij} \phi(x_j)\|^2 \quad (8)$$

To find the optimal weights for reconstructing, we must compute the covariance matrix C_{jk} in kernel space which is:

$$C_{jk}^i = (\phi(x_i) - \phi(x_j))^T (\phi(x_i) - \phi(x_k)) \quad (9)$$

Then replacing each resulting term of form $\phi(x_a) \cdot \phi(x_b)$ in Eq. (4) with the corresponding kernel value $K_{ab} = K(x_a, x_b)$. The kernelized covariance matrix can be obtained by

$$\forall x_j, x_k \in N(x_i) \quad C_{jk}^i = K_{ii} - K_{ij} - K_{ik} + K_{jk} \quad (10)$$

We can compute the weights matrix W of kernel space and use it as the similarity matrix for spectral clustering.

6. Proposed Algorithm

In this section, we will introduce our algorithm called spectral clustering algorithm based on local neighbors in kernel space (SC-LNK). The main procedure of SC-LNK is summarized in Algorithm 2.

```
Algorithm 2: The SC-LNK Algorithm
Input: Data set  $X$ , number  $k$  of clusters
Output: A partitioning  $S_1, S_2, \dots, S_k$ 
begin
    Construct similarity matrix  $W \in R^{n \times n}$  according to
    Eq. (3) and Eq. (9)
    Use the affinity matrix to execute the algorithm 1
end.
```

The main advantage of our algorithm is that SC-LNK can effectively and correctly discover the underlying cluster structure by taking the advantage of the smooth graph W in kernel space. When using the Mercer kernel, the selection of parameter σ adopts the concept of local scale in [13] and σ_i equals the distance of the 15th neighbor of point x_i . Furthermore the parameter k in SC-LNK algorithm is more stable and more accuracy than former methods.

7. Experiments

In this section, we will report our results on several synthetic data sets and the real data sets.

7.1. Experiments on Synthetic Data Sets

We applied the SC-LNK algorithm and LSC method to five synthetic data sets that has been mentioned in [10], [13]. The results are show in Fig. 1 and Fig. 2.

As we can see, both SC-LNK and LSC can recognize the non-convex cluster and reliably finds groups consistent with what a human's intuitive solution in Fig. 1. However, the LSC method fails to find the real structure

hidden in data sets which are multiscale in Fig. 2 while the SC-LNK algorithm can distinguish it after mapping to kernel space.

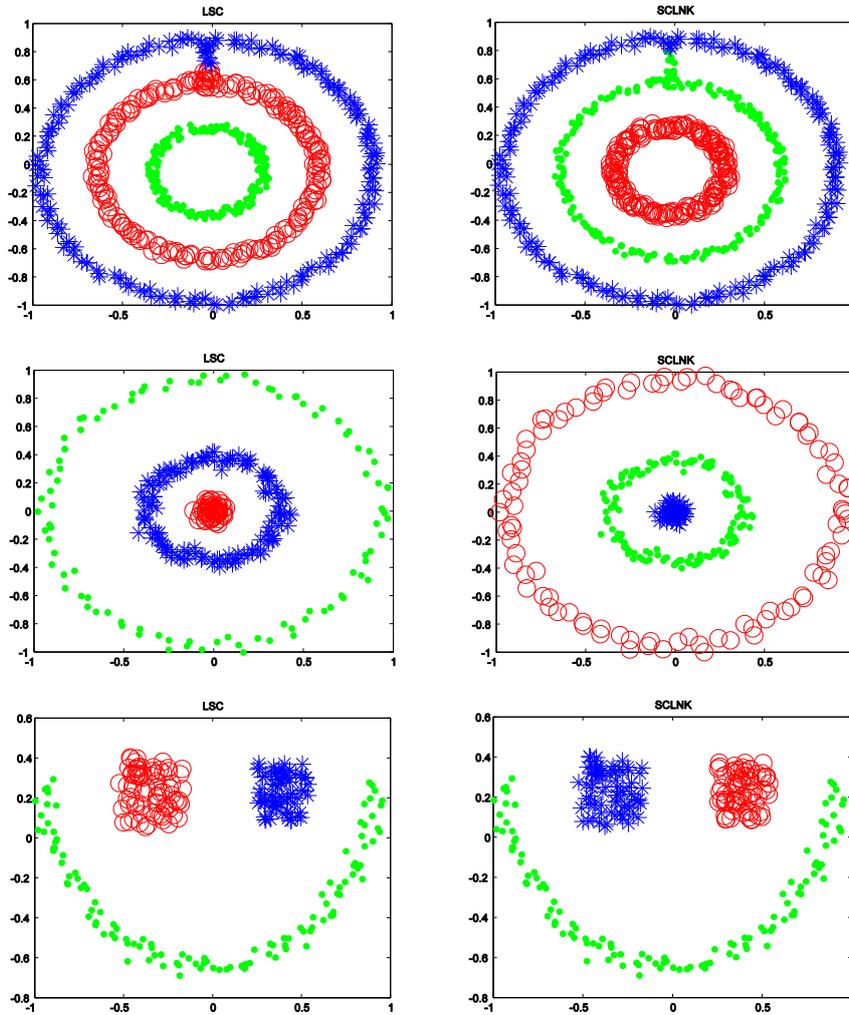


Fig. 1. Contrast results on Synthetic data. The left column shows the LSC algorithm results; The right column presents the SC-LNK clustering results. As shown above, both algorithms can recognize the non-convex clusters.

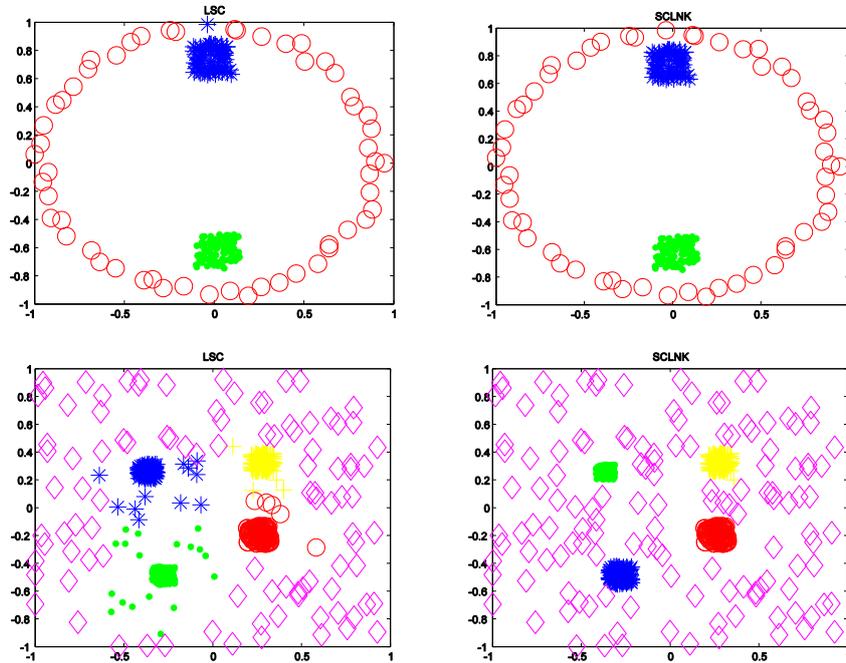


Fig. 2. Contrast results on Synthetic data. The left column shows the LSC algorithm results; The right column presents the SC-LNK clustering results. As shown above, LSC fails to find the real structure hidden in data sets while SC-LNK can distinguish.

7.2. Experiments on Real Data Sets

In order to extensively examine the effectiveness of the SC-LNK algorithm, we further compare the performances of our method with the other three clustering algorithms which include the SC algorithm, SSC method and the LSC algorithm. To evaluate the performance of different clustering algorithms, two different metrics are used: one is Rand index, and the other is the Normalized Mutual Information.

Evaluation metrics. Rand index (RI) is widely used to evaluate the clustering performance. A decision is considered correct if the clustering algorithm agrees with the real clustering. RI [21] is defined as

$$RI = \frac{\#CD}{n(n-1)/2} \quad (11)$$

where CD denotes the number of correct decisions. A larger RI value ($RI \in [0,1]$) signifies a better clustering result.

Normalized Mutual Information (NMI) is another measure for determining the quality of clusters. For two random variable X and Y , the NMI is defined as [22]:

$$NMI(X, Y) = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}} \quad (12)$$

where $I(X, Y)$ is the mutual information between X and Y , while $H(X)$ and $H(Y)$ are the entropies of X and Y respectively. Note that $NMI \in [0,1]$ and give a clustering result, the NMI is estimated as

$$NMI = \frac{\sum_{l=1}^k \sum_{h=1}^c n_{l,h} \log\left(\frac{nn_{l,h}}{\hat{n}_h}\right)}{\sqrt{\left(\sum_{l=1}^k n_l \log \frac{n_l}{n}\right) \left(\sum_{h=1}^c \hat{n}_h \log \frac{\hat{n}_h}{n}\right)}} \quad (13)$$

where n_l denotes the number of data contained in the cluster $C_l (1 \leq l \leq k)$, \hat{n}_h is the number of data belonging to the h -th class and $n_{l,h}$ denotes the number of data which are in the intersection between the cluster C_l and the h -th class. The larger the NMI, the better the performance.

Results on uci data sets. We carry out the experiments on five data sets which come from the UCI data repository [23]. The properties of the datasets are summarized in Table 1.

Table 1. Properties of UCI Datasets

Dataset	Iris	Wine	Ionosphere	Glass	Segmentation
No. of instances	150	178	351	214	210
No. of attributes	4	13	34	9	19
No. of clusters	3	3	2	6	7

Experimental results are presented in Fig. 3 and Fig. 4. Especially, following [24], we set the parameter σ in SC as 0.05 of the maximal pairwise Euclidean distance among the dataset. We adopt an empirical value to estimate the parameter k and report the best result for the LSC and SC-LNK algorithm.

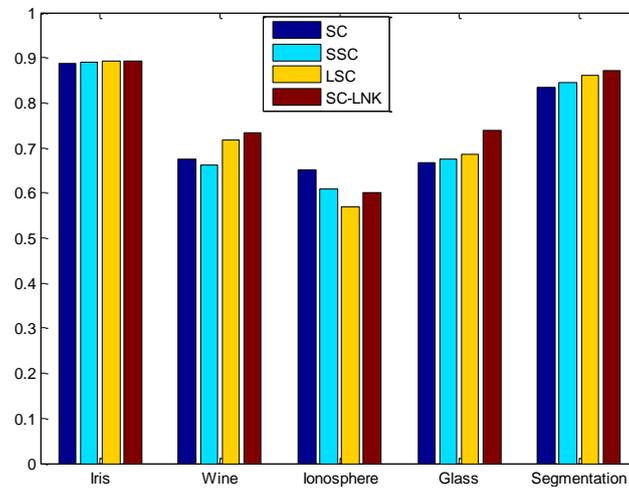


Fig. 3. Results on UCI data by RI

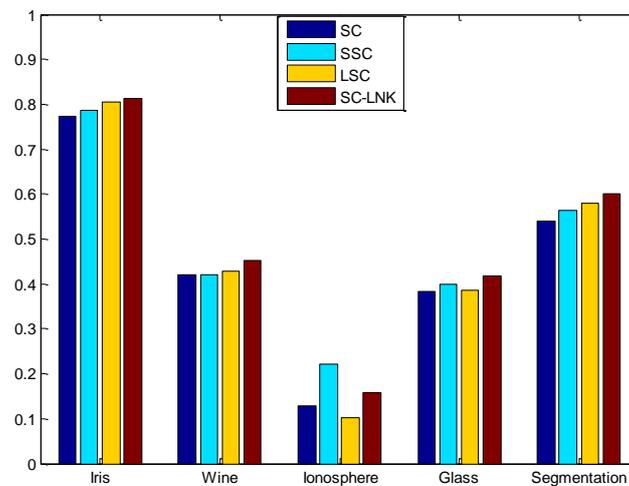


Fig. 4. Results on UCI data by NMI

From Fig. 3 and Fig. 4, we can see that the SSC algorithm has achieved limited success on real world data sets. SC-LNK outperforms the other algorithms on most data sets, especially in the wine, glass and segmentation dataset, the improvement is obvious. But on the ionosphere dataset, the performance is not as good as SC and self-tuning. Despite this, it can still be observed that SC-LNK works well on UCI datasets. With linear reconstruction

in kernel space, the local relationship is maintained and different clusters are more linearly separable, thus SC-LNK can distinguish the intrinsic structure of the data more correctly.

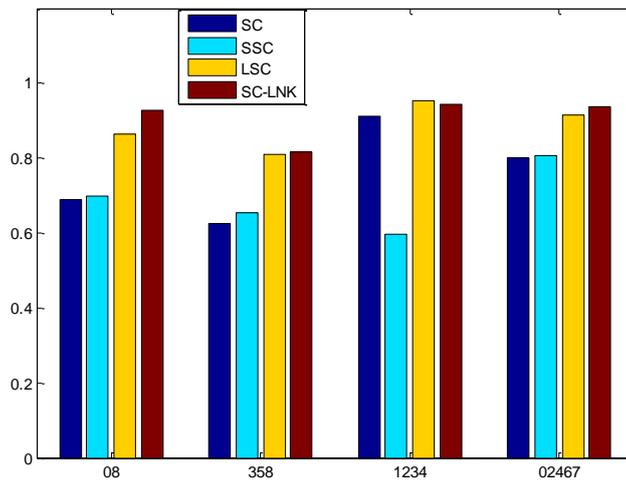


Fig. 5. Results on USPS data by RI

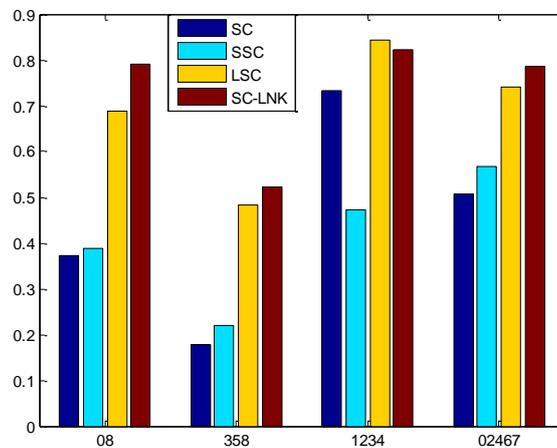


Fig. 6. Results on USPS data by NMI

Results on usps data sets. In this experiment, we consider the hand written digits from the well-known USPS database.

The digits have been normalized and centered to 16×16 gray-level images, thus the dimensionality of digit space is 256, as each sample image will be transformed to a vector as one column of the similarity graph. In the database it contains 7291 training instances and 2007 test instances.

We choose digits $\{0,8\}$, $\{3,5,8\}$, $\{1,2,3,4\}$ and $\{0,2,4,6,7\}$ as subsets and carry out the experiments separately. The clustering results by RI and NMI are presented in Fig. 5 and Fig. 6 separately.

According to the results in Fig. 5 and Fig. 6, we can see that the idea of linear reconstruction has a distinct advantage in the USPS data sets. The performance of LSC and SC-LNK algorithms are much better than the other two methods. Even on the challenging USPS subsets $\{1,2,3,4\}$, both of them still have improvement compared to SC and SSC though LSC is better than SC-LNK. But in most cases, the effectiveness of SC-LNK algorithm is better than LSC and the other two methods. It proves that the new similarity measure using linear reconstruction in kernel space is very helpful in detecting the real manifold of the digits.

8. Conclusion

In this paper, we propose a novel method of constructing similarity matrix using linear reconstruction. Based on this method we introduce the concept of kernel methods and propose an efficient spectral clustering algorithm called SN-LNK. Experimental results on five synthetic data sets and two groups of the real data sets show that the proposed algorithm achieves considerable improvements over traditional spectral clustering, locality spectral clustering and self-tuning spectral clustering algorithms.

There is still much work for us to undergo further research. The similarity matrix we obtained in this paper can be extended to other clustering algorithms based on the affinity matrix. Furthermore, the selection on the number of nearest neighborhoods k still remains to be further studied. We will pursue these research directions in our future work.

References

1. Hagen, L., Kahng, B.: New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on computer aided design*, vol. 11, no. 9. (1992)
2. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, 888–905. (2000)
3. Jordan, F., Bach, F.: Learning spectral clustering. In *Advances in neural information processing systems 16: proceedings of the 2003 conference*, vol. 16. The MIT Press, 305. (2004)
4. Ozerem, U., Erdogmus, D., Jenssen, R.: Mean shift spectral clustering. *Pattern*

- Recognition, vol. 41, no. 6, 1924–1938. (2008)
5. Wang, X., Davidson, I.: Active spectral clustering. In *2010 IEEE International Conference on Data Mining*. 561–568. (2010)
 6. Li, Z., Liu, J., Tang, X.: Constrained clustering via spectral regularization. In *Computer Vision and Pattern Recognition*. 421–428. (2009)
 7. Luxbug, U.: A tutorial on spectral clustering. *Statistics and Computing*, vol. 17, no. 4, 395–416. (2007)
 8. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. *Science*, vol. 290, no. 5500, p. 2323. (2000)
 9. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*. 55–67. (2007)
 10. Gong, Y., Chen, C.: Locality Spectral Clustering. *AI 2008: Advances in Artificial Intelligence*. 348–354. (2008)
 11. Scholkopf, B., Smola, A., Muller, K.: Kernel principal component analysis. *Artificial Neural Networks-ICANN'97*. 583–588. (1997)
 12. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14: Proceeding of the 2001 Conference*. 849–856. (2001)
 13. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. *Advances in neural information processing systems*, vol. 17, no. 1601–1608, p. 16. (2004)
 14. Zhang, X., Li, J., Yu, H.: Local density adaptive similarity measurement for spectral clustering. *Pattern Recognition Letters*. (2010)
 15. DeCoste, D.: Visualizing Mercer kernel feature spaces via kernelized locally linear embedding. In *Proceedings of the Eighth International Conference on Neural Information Processing (ICONIP-01)*. (2001)
 16. Jianbo, S., Yu, S., Shi, J.: Multiclass Spectral Clustering. In *International Conference on Computer Vision*. Citeseer, 313–319. (2003)
 17. Meila, M., Shi, J.: A random walks view of spectral segmentation. *AI and Statistics (AISTATS)*, vol. 2001. (2001)
 18. Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*. 595–602. (2004)
 19. Ben-Hur, A., Horn, D., Siegelmann, H., Vapnik, V.: Support vector clustering. *The Journal of Machine Learning Research*, vol. 2, p. 137. (2002)
 20. Girolami, M.: Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, vol. 13, no. 3. 780–784. (2002)
 21. Rand, W.: Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 846–850. (1971)
 22. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, vol. 3, 583–617. (2003)
 23. Asuncion, A., Newman, D.: UCI machine learning repository. (2007). [Online]. Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
 24. Xia, T., Cao, J., Zhang, Y., Li, J.: On defining affinity graph for spectral clustering through ranking on manifolds. *Neurocomputing*, vol. 72, no. 13–15, 3203–3211. (2009)

An Improved Spectral Clustering Algorithm used on Local Neighbors in Kernel Space

Xinyue Liu received the M.S degree in Computer Science and technology from Northeast Normal University, China, in 2006. She is currently working toward the Ph.D. degree in the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. Her research interests include multimedia information retrieval, web mining and machine learning.

Xing Yong received his B.S. degree in Software Engineering from Dalian University of Technology in 2010, and currently is a master degree candidate in the same place. His major interests lie in machine learning.

Hongfei Lin received the Ph.D degree from Northeastern University, China. He is a professor in the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. His professional interests lie in the broad area of information retrieval, web mining and machine learning, affective computing.

Received: April 15, 2011; Accepted: June 15, 2011.

A Novel Capacity and Trust Based Service Selection Mechanism for Collaborative Decision Making in CPS

Bo Zhang¹, Yang Xiang¹, Peng Wang^{1,2}, and Zhenhua Huang¹

¹College of Electronics and Information Engineering, Tongji University,
201804 Shanghai, PR China
{06zhangbo, shxiangyang}@tongji.edu.cn

²Development Center of Computer Software Technology,
201112 Shanghai, PR China
haris2003@163.com

Abstract. Cyber-physical system (CPS) provides more powerful service through combining software service and physical device. It is an effective solution to organize various CPS services to realize collaborative decision making (CDM). In CPS, finding out the most competent participators for CDM sponsor is a core problem. To solve this problem, we propose a novel capacity and trust computation based CPS service selection mechanism in intelligent and automatic manners. It comprises three phases, including capacity evaluation, trust computation and negotiation selection. In the first phase, CDM sponsor describes formal semantic of decision task and computes the capacity evaluation values according to participator instructions. In the second phase, we design a novel trust computation method to calculate the values of activity trust, subjective belief, objective reputation, physical trust and recommended trust respectively. In the third phase, service selection is achieved through a negotiation mechanism according to capacity evaluation and trust computation.

Keywords: cyber-physical system, collaborative decision making, service selection, semantic, capacity, trust.

1. Introduction

Collaborative decision making (CDM) becomes a popular and feasible solution for the increasing complexity of decision making requirement from substantive users. Traditional CDM is implemented by decision support system through web service cooperation [1-2]. In such situation, CDM consists of heterogeneous and geographically distributed cyber components with different capacity. Performance and reliability of CDM depend on all the cyber capacities in virtual world [1-3]. The advent of Cyber-physical system

(CPS) enables people to make use of the facilities in physical world to make CDM more powerful.

Cyber-physical system (CPS) is a system which is tight combination of information computation and physical environment [4-8]. Software services are embedded into physical devices in order to provide more convenient and efficient services to people. Nowadays, CPS, such as smart buildings, medical devices, intelligent traffic control system, will provide a high-quality decision making through integrating computational systems in virtual world and infrastructures in physical world to cope with the increasing complex demands from people [5]. Based on the computational and physical service capacities, CPS will offer more competent services for CDM. Since both the computational systems in virtual world and the infrastructures in physical world are service providers, it is crucial to make CDM in CPS competent in virtual and physical aspects for users. Thus, a challenge of CDM in CPS is how to find out an efficient method to select competent services from the heterogeneous and geographically distributed services.

Existing researches in CPS mainly focused on architecture [6-8], middleware designing [9], system control [10-11], system security [12-14], QoS[15-16] or real-time data management [17]. However, as a service provider for users, it is critical to identify the competence degree of CPS's services for CDM, which has received limited attention in study.

In an open, relax coupling and dynamic environment, many CPS service provisions (CPSSP), including CPSSP in virtual and physical world, are not free to be available. And CDM sponsors generally have insufficient knowledge about all CPSSPs. As a result, CDM sponsor has to accept CPSSP's payment conditions without any opportunity to experience the service in advance. On the other hand, CDM sponsor may abandon a high quality CPS service because it lacks sufficient knowledge to certify service's ability. Such asymmetric position would results in inefficient and improper CPS service provision. To overcome these problems, CPS requests an effective mechanism to identify and exhibit competence degree in order to make services ease-of-use for CDM.

Trust is an effective solution for CDM service selection in CPS. CDM sponsor can decide whether a CPS service should be selected or not depending on the security or credit degree even though they do not have ample information about the service. There are many existing trust evaluation methods, such as summation/average of trust rating or past judgment ranking, to figure out their creditable degrees for traditional information systems. But these trust evaluation methods mainly focus on cyber features which are inherent properties of software, i.e. software actions or information content security, etc. Different from traditional cyber systems, CPS trust evaluation should pay attention to formulate an appropriate trust computation for both cyber and physical features. Since existing methods lack feasible ways to evaluate trust of physical components, they are not sufficient for the CPS.

In our view, the natural characteristics of CPS' service, which should be emphasized in selection evaluation, are as follows:

- 1) Capacity of software service
- 2) Trust of software service
- 3) Capacity of physical service provider
- 4) Trust of physical service provider

From above characteristics, CDM sponsor can communicate with both cyber and physical components of CPS to realize about the competence degree of CPSSP, and launch the capacity and trust evaluation.

In this paper, we propose a novel capacity and trust based CPS service selection mechanism to identify the most competent services for CDM. We address the formal semantic to describe the characteristics of CPSSP so that the communicating between CDM sponsor and CPSSPs would share a common knowledge base in our mechanism. Our mechanism comprises three phases, i.e., semantic description for CDM sponsor requirements and capacity of CPSSP, trust evaluation of CPS and negotiation selection of CPSSP. In the first phase, CDM sponsor describes formal semantic of complex requirement and estimates the capacity values according to candidate instructions from different CPSSPs. In the second phase, a novel trust computation method is adopted to calculate the trust degrees of CPSSP's different characteristics. In third phase, service selection is achieved through a negotiation mechanism based on the results of capacity evaluation and trust computation.

The rest of this paper is organized as follows. In Section 2, a brief introduction of related work is presented. Three phases of our mechanism are described in detail from Section 3 to Section 5 respectively. The service selection framework of CDM in CPS is presented in Section 6. Finally, we conclude the paper in Section 7.

2. Related Work

2.1. Collaborative Decision Making

Collaborative decision making has been widely used in many application domains, such as airport management [18-19], GIS map [20], and stakeholder research [21]. In practice, the CDM framework is proposed in three ways, i.e., Internet based CDM [22-23], multi-agent based CDM [24], and web service based CDM [25-26]. Internet based CDM is a traditional way to organize the decision making. The main challenge of Internet based CDM is how to transfer isometric data and information across wide networks. Multi-agent is a feasible and optimized solution for CDM. Agent has abilities of negotiation, decision making and knowledge interaction, which can partially realize intelligent and automatic CDM. However, because agents lack the mechanism of self-description in a machine readable format, it is difficult for agent oriented CDM to identify qualified decision making partners. In recent

studies, web service becomes a popular solution. Web service is a software program designed to support interoperable machine-to-machine interaction over a network [27]. In service oriented architecture, collaborative work can be considered to be autonomous through a set of messages and commands. In this paper, we utilize web service environment to organize the CDM.

2.2. Selection Research of Decision Making

As the core problem of CDM, service selection is constantly treated as decision model selection in traditional DSS (decision support system). Artificial intelligent (AI) techniques are widely used for model selection, such as CBR (Case Based Reasoning) [28], RBR (Rule Based Reasoning) [29], ANN (Artificial Neural Network), and GA (Genetic Algorithm) [2]. Statistical methods, such as Bayesian information criteria, are also frequently adopted for decision model selection [30]. However, these existing methods are not designed for open and distributed network. Mou et al. proposed a QoS based service selection in CDM [26], where QoS is measured as the capacity of web service. While Mou's model mainly focuses on service capacity forecasting, our capacity and trust computation strategy provides a comprehensive solution for efficient service selection.

2.3. Trust Computation Research

In trust computation, belief and reputation are two core conceptions for creditable description. Belief is a subjective concept that demonstrates a creditable relationship between two or more individuals. On the other hand, reputation presents the whole common schema from all the qualified members. As a consequence, we think that the service selection mechanism is to identify the service with good reputation from the independent third party and the trustable ones from the sponsor's belief.

There have been a large number of research efforts on belief and reputation in the past decades [31-35]. Many methods, such as summation/average of trust rating [36] and Bayesian systems [37], have been proposed to optimize one or more aspects of trust computation performance. Based on the trust computation, there are two main types of architectures of reputation system: centralized and distributed. The former has a central authority to collect all the rating, and publish reputation score for every participant. Whereas in distributed reputation system, each member gets the belief about each experience with others, and submits the reputation on request from relying members.

In our previous research, we proposed a trust computation based model selection for decision support system, which considers the trust from subjective and objective perspective [38,39]. However, these methods cannot use both capacity and trust aspects for service selection.

For CPS, many researchers are working on security or trust issues which are important for service selection [40]. Efforts include the achievements of formalizing the definition of trust [41-42], trust management and trust negotiation methodologies [43-44]. Many researchers addressed trust issues of pervasive computing environments [45-46], trust-based communication and interoperation approaches [47-50]. However, trust computation, which should focus on the CPS substantive characteristics, has not been addressed adequately in cyber-physical systems.

In our thoughts, what is missing is a comprehensive view for cyber-physical systems that integrates both the cyber trust aspects and the physical trust aspects of CPS and allows the sponsor in the physical world to evaluate the service and interact with the cyber and physical service components based on trust value. Trust computation must be proposed for physical environments and the cyber service software should adhere to them. What is also missing is a notion of trust based interoperation for cyber-physical systems where different systems will interact in a dynamic environment to achieve CDM.

3. Semantic based Capacity Evaluation of CDM

Firstly, we provide a table (Table 1) which lists a set of nomenclatures that will be frequently used in the rest of the paper.

3.1. Semantic of Decision Requirement and CPS Service

To evaluate the quality of a candidate service, users should match the service's capacities with their requirements. In definition 1, we describe the requirements from both virtual computational aspects and the physical objects aspects so that the characteristics of CPS are shown in the definition. We define the semantic of user's requirement to as follows.

Definition 1 Requirement semantic of decision task is a 2-tuple as $\mathcal{N} = (\mathcal{N}^V, \mathcal{N}^P)$. Here \mathcal{N}^V and \mathcal{N}^P represent the user's virtual requirement semantic and physical requirement semantic respectively. Virtual requirement semantics can be defined as $\mathcal{N}^V = (\mathcal{N}^{vc}, \mathcal{N}^{vr}, \mathcal{N}^{goal}, \mathcal{N}^{cost}, \mathcal{N}^{con})$, $\mathcal{N}^{vc}, \mathcal{N}^{vr}, \mathcal{N}^{goal}, \mathcal{N}^{cost}$ and \mathcal{N}^{con} represent virtual requirement class name, structure relationships of virtual requirement, goals, affording service cost price and preconditions respectively, while \mathcal{N}^P , the physical requirement semantic, can be defined as $\mathcal{N}^P = (\mathcal{N}^{pc}, \mathcal{N}^{pr}, \mathcal{N}^{envir}, \mathcal{N}^{time})$. Parameter \mathcal{N}^{pc} denotes candidate physical service provider's class. \mathcal{N}^{pr} is the relationships of physical service provider. \mathcal{N}^{envir} is running environment

requirement of physical service. \aleph^{time} is time control criterion of physical service.

Table 1. Nomenclature

Symbol	Description
\aleph	Semantic of decision requirement
\aleph	Participator instruction semantic
$E(x (R, S, F))$	Semantic of CPS service physical environment
Ω	Semantic of CPS service activity
$capacity(\aleph^{id})$	Capacity evaluation value of CPS service
$value_{goal}(\aleph^{id})$	Goal evaluation value of CPS service
$value_{time}(\aleph^{id})$	Time forecasting value of decision making service
$match(\aleph_j^{time})$	Match function that calculates the excess time of \aleph_j^{time} relative to \aleph^{time}
$value_{price}(\aleph^{id})$	Cost evaluation value of decision making service
$over(\aleph_k^{cost})$	Function that calculates the excess cost of \aleph_k^{price} relative to \aleph_k^{cost}
$value_{environment}(\aleph^{id})$	Price environment judgment value of CPS service
$get(E_{\aleph}(x_1))$	Function that calculates number of environment requirements of \aleph^{envir} satisfied by \aleph^{envir}
$AT_{\aleph}(\Omega)$	Activity trust value of CPS service
$CT_{\aleph}(\Omega^{class})$	Class trust value of activity proposed by service \aleph
$ST_{\aleph}(\Omega^{class})$	Status trust value of service's activity
BD	Belief dependence value from sponsor to a service
BR	Belief relationship from sponsor to a service provider
RR	Reputation ranking for CPSSP
$RR_{TL}(SP)^{TL_{unit}}$	Reputation ranking value based on time limitation
$RR_{SI}(SP_1)$	Reputation ranking value based on source identity
$RR_{RD}(SP)$	Reputation ranking value based on ranking delay
$RT^{re}(SP_2, \aleph)$	Recommended trust value
$\phi(SP)$	Confidence conformation factor
$PT(\aleph^{id})$	Physical trust of CPSSP's device
$FT(\aleph^{id})$	Fault tolerance trust value of CPSSP's device
$HT(\aleph^{id})$	Healthiness trust value of CPSSP's device

CPS service by nature consists of two components: cyber software and physical environment. We consider that semantic of CPS service should be described from above two aspects. CDM sponsor needs a decision making service whose capacity can satisfy the requirement semantic. Therefore, each CPSSP would generate an instruction to introduce its service capacity of decision making. We define the participator instruction semantic of decision making service from CPSSP for capacity evaluation as follows.

Definition 2 Participator instruction semantic of CPS's service is defined as a 2-tuple as $\mathfrak{S} = (\mathfrak{S}^V, \mathfrak{S}^P)$ Here \mathfrak{S}^V and \mathfrak{S}^P represent the service's virtual capacity semantic and physical object semantic respectively. \mathfrak{S}^V can be defined as $\mathfrak{S}^V = (\mathfrak{S}^{id}, \mathfrak{S}^{vc}, \mathfrak{S}^{goal}, \mathfrak{S}^{price})$ according to its capacity. Parameter \mathfrak{S}^{id} denotes the exclusive identification of service. \mathfrak{S}^{vc} is the class of decision task which service is able to make. \mathfrak{S}^{goal} is a set of anticipated goals which can be achieved by service. \mathfrak{S}^{price} describes the price that the sponsor should pay for decision making service. \mathfrak{S}^P can be defined as $\mathfrak{S}^P = (\mathfrak{S}^{pc}, \mathfrak{S}^{source}, \mathfrak{S}^{envir}, \mathfrak{S}^{time})$ according to its capacity. \mathfrak{S}^{pc} denotes the physical class of CPSSP. \mathfrak{S}^{source} points out the source of service in CPSSP. \mathfrak{S}^{envir} represents the running environment of service. \mathfrak{S}^{time} represents the time that service would spend on decision making.

3.2. Capacity evaluation for service

The CDM sponsor may accept the service that satisfy as many as the number of goals in \aleph^{goal} under the restriction of time in \aleph^{time} , costs in \aleph^{cost} and environment requirements \aleph^{envir} . In other words, capacity evaluation for service comprises four aspects, i.e., goal evaluation, time forecasting, prices estimation, and environment judgment.

Goal evaluation. Goal evaluation aims to identify the goals of \aleph^{goal} that is achievable by a CPS service according to its \mathfrak{S}^{goal} . We measure this capacity criterion based on the number of goals which can be realized by service and the importance of the realizable goals. Firstly, we define an equalization mapping function between two semantic as follow.

Definition 3 Let x and y are the elements in \aleph and \mathfrak{S} respectively. Equalization Mapping function $N(x) \rightarrow y$ is a transfer relationship between x and y , which represents that the two elements are equal on the semantic level.

Let the set of \aleph^{goal} be $\aleph^{goal} = (\aleph_1^{goal}, \aleph_2^{goal}, \dots, \aleph_n^{goal})$. For each \aleph_i^{goal} , it has a weight w_i with the constraint $\sum_{i=1}^n w_i = 1$. Then the value of goal evaluation is calculated as follow,

$$score_{goal}(\mathfrak{S}^{id}) = \sum_{N(\mathfrak{S}^{goal}) \rightarrow \aleph^{goal}} w \quad (1)$$

Because a large number of \mathfrak{S}^{goal} can satisfy $N(\mathfrak{S}^{goal}) \rightarrow \aleph^{goal}$, the decision making task will be extremely complicated. To solve this problem, we introduce the impact factor calculation for goal evaluation. Let m is the number of \mathfrak{S}^{goal} that satisfy $N(\mathfrak{S}^{goal}) \rightarrow \aleph^{goal}$, and then the final value of goal evaluation can be calculated as follows,

$$value_{goal}(\mathfrak{S}^{id}) = \begin{cases} score_{goal}(\mathfrak{S}^{id}) \times \left(\frac{m}{n}\right)^{\left(1-\frac{1}{m}\right)} & m \geq 2 \\ score_{goal}(\mathfrak{S}^{id}) \times \frac{1}{n} & m = 1 \end{cases} \quad (2)$$

where n is the number of \aleph^{goal} set.

Formula 2 shows the gross importance of goals which can be realized by a candidate service. Here, we consider the number m and n as regulation parameters in formula 2 in order to make the result more effectively. They would influence the value of goal evaluation as adjusting parameters. For example, if there is a goal set of $\aleph^{goal} = (g_1, g_2, g_3, g_4)$ and their weights are 0.4, 0.3, 0.2, and 0.1 respectively. Service A has $\mathfrak{S}_A^{goal} = (g_2)$, while service B has $\mathfrak{S}_B^{goal} = (g_1, g_2, g_3)$. The goal evaluations of service A and B are 0.075 and 0.74 according to formula 2.

Time forecasting. Time forecasting aims to evaluate whether the CPSSP physical components response time satisfies the sponsor's requirement. Here responds time is measured as the time interval between decision and service. In general, the shorter response time in decision making, the larger value of \mathfrak{S}^{time} would be assigned from CDM sponsor.

For time forecasting, we denote maximum affording time \tilde{T} and anticipant time \aleph^{time} from CDM sponsor. Maximum affording time indicates the maximum time limit that would be acceptable by decision making sponsors. Anticipant time signifies the decision making spending time that would be the time interval sponsor looks forward to the most. Let the set of response time

given by service be \mathfrak{S}^{time} . The value of time forecasting can be calculate as following:

$$value_{time}(\mathfrak{S}^{id}) = \tilde{T} - \sum_{j=1}^l (\mathfrak{S}_j^{time})^\eta - \sum_{j=1}^l match(\mathfrak{S}_j^{time}) . \quad (3)$$

where, $\eta(\eta \in [0,1])$ is an impact factor given by sponsor, and l is the number of \mathfrak{S}^{time} set. We also propose a match function $match(\mathfrak{S}_j^{time})$ that calculates the excess time of \mathfrak{S}_j^{time} relative to \mathfrak{N}_j^{time} as follow.

$$match(\mathfrak{S}_j^{time}) = \begin{cases} 0 & \text{if } N(\mathfrak{S}_j^{time}) \rightarrow \mathfrak{N}_j^{time} \\ |\mathfrak{S}_j^{time} - \mathfrak{N}_j^{time}| & \text{else} \end{cases} . \quad (4)$$

Price estimation. Price estimation aims to test whether the service's price \mathfrak{S}^{price} is overcharge. As time forecasting, the less price service charge for decision making, the more value of \mathfrak{S}^{price} would be given from CDM sponsor. We denote maximum affording cost \tilde{C} as maximum cost limits that would be acceptable by decision making sponsors.

Let the sponsor's maximum affording cost be \tilde{C} . The price estimation can be calculated as following:

$$value_{price}(\mathfrak{S}^{id}) = \tilde{C} - \sum_{k=1}^q (\mathfrak{S}_k^{price})^\eta - \sum_{k=1}^q over(\mathfrak{S}_k^{price}) . \quad (5)$$

Here η is the same impact factor as in formula 3, and q is the number of \mathfrak{S}^{price} set. We also propose a function $over(\mathfrak{S}_k^{price})$ that calculates the excess cost of \mathfrak{S}_k^{price} relative to \mathfrak{N}_k^{cost} as follows:

$$over(\mathfrak{S}_k^{price}) = \begin{cases} 0 & \text{if } N(\mathfrak{S}_k^{price}) \rightarrow \mathfrak{N}_k^{cost} \\ |\mathfrak{S}_k^{price} - \mathfrak{N}_k^{cost}| & \text{else} \end{cases} . \quad (6)$$

We give an example here for explaining formulas of time forecasting and price estimation. Table 2 shows the related semantic values of requirement and service.

Table 2. Example of time forecasting and price estimation

	\aleph^{goal}	\aleph^{cost}		\aleph^{time}		\mathfrak{S}^{price}	\mathfrak{S}^{time}	η
		goal	cost	goal	time			
	G1	G1	≤ 20	G1	≤ 4	15	4	
value	G2	G2	≤ 10	G2	≤ 7	8	5	0.9
	G3	G3	≤ 15	G3	≤ 4	20	5	

From data in table 2, we can calculate value of time forecasting and price estimation through formula 3, 4, 5 and 6 as follows,

$$\begin{cases} value_{time}(\mathfrak{S}) = (4 + 7 + 4) - (4 + 5 + 5)^{0.9} - (0 + 0 + 1) = 3.25 \\ value_{price}(\mathfrak{S}) = (20 + 10 + 15) - (15 + 8 + 10)^{0.9} - (0 + 0 + 5) = 14.74 \end{cases}$$

Physical environment judgment. It is crucial to check whether the physical environment of CPSSP is competent for decision making. We denote the environment semantic for decision making as follows.

Definition 4 Physical environment semantic can be described as $E(x | (R, S, F))$. In this formula, the ordered pair $x | (R, S, F)$ illustrates a set of binary relations $r_1(x, y_1), \dots, r_n(x, y_n)$ of any certain physical object x , the set of statuses $s_1(x, z_1), \dots, s_m(x, z_m)$ of x at a certain time, and the set of rules f_1, \dots, f_r of the object x . We abbreviate $E(x | (R, S, F))$ as $E(x)$.

Physical environment semantic is a describable context. This kind of semantic makes it possible that each physical object has a certain context which could be understood explicitly by sponsor.

Let sponsor's environment requirement of physical object x_1 be $\aleph^{envir} = E_{\aleph}(x_1)$, and the physical environment of x_1 which can be provided by CPSSP's service instruction be $\mathfrak{S}^{envir} = E_{\mathfrak{S}}(x_1)$. The physical environment judgment can be calculated as following:

$$value_{environment}(\mathfrak{S}^{id}) = \frac{\sum get(r_{\mathfrak{S}}(x_1)) + \sum get(s_{\mathfrak{S}}(x_1)) + \sum get(f_{\mathfrak{S}}(x_1))}{|R_{\aleph}(x_1)| + |S_{\aleph}(x_1)| + |F_{\aleph}(x_1)|} \quad (7)$$

Here, the function $|R_{\aleph}(x_1)|$ indicates the total number of binary relations of physical object x_1 . We also propose a function $get(E(x_1))$ that calculates the number of environment requirements of \aleph^{envir} satisfied by \mathfrak{S}^{envir} as follows:

$$get(E_{\mathfrak{S}}(x_1)) = \begin{cases} 1 & \text{if } N(r_{\mathfrak{S}}(x_1)) \rightarrow r_{\mathfrak{S}}(x_1) \text{ or } N(s_{\mathfrak{S}}(x_1)) \rightarrow s_{\mathfrak{S}}(x_1) \\ & \text{or } N(f_{\mathfrak{S}}(x_1)) \rightarrow f_{\mathfrak{S}}(x_1) \\ 0 & \text{else} \end{cases} \quad (8)$$

According to goal evaluation, time forecasting, price estimation and environment judgment, capacity evaluation value can be calculated as follows:

$$capacity(\mathfrak{S}^{id}) = \sigma_1 \times value_{goal}(\mathfrak{S}^{id}) + \sigma_2 \times value_{time}(\mathfrak{S}^{id}) + \sigma_3 \times value_{cost}(\mathfrak{S}^{id}) + \sigma_4 \times value_{environment}(\mathfrak{S}^{id}) \quad (9)$$

Here σ is a weight with the constraint $\sum_{i=1}^4 \sigma_i = 1$.

4. Trust Computation

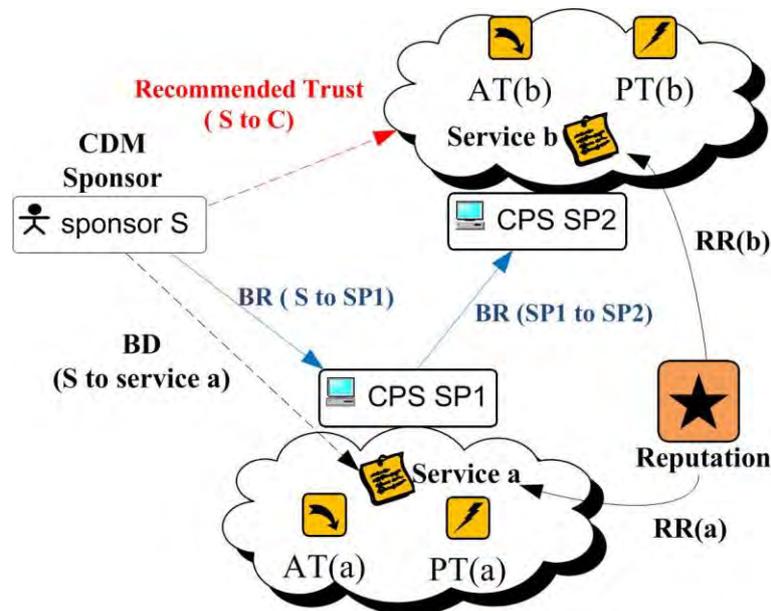


Fig. 1. Trust computation of trust

We study the trust based CPSSP selection in four aspects: activity trust, belief, reputation and physical trust. Activity trust (AT) is the trust degree of decision making process of CPS service. Belief is the subjective trust between different CPSSPs, which consists of belief dependence (BD) and

belief relationship (BR). Belief dependence is described by the trustable value from CDM sponsor to candidate services. And belief relationship means trust relationship value between CDM sponsor and CPSSP. In other hand, reputation reflects objective credit of CPSSP. Finally, physical trust (PT) is the trust that points out whether the physical device of CPSSSP is creditable or not. Furthermore, we introduce a recommended trust for CDM sponsor to study the strange CPSSPs within trust computation. Figure 1 shows an example of our trust computation framework.

4.1. Activity Trust Computation of CPS Service

Activity represents statuses transition during the process of CPS service making decision. Activity trust computation gives the opportunity for CDM sponsor to realize that whether a CPS service's work is creditable or not in advance. We define the semantic of a CPS service's activity as following:

Definition 5 A activity semantic description is a kind of representation as $\Omega = (\Omega^{class}, \Omega^{exe}, \Omega^{rec}, \Omega^{pre}, \Omega^{post})$. Here, Ω^{class} denotes the class name of activity. Parameters Ω^{exe} and Ω^{rec} represent the executor and receivers of activity respectively. Parameters Ω^{pre} denotes the previous statuses before the activity being executed, while Ω^{post} denotes the post statuses after the activity being executed.

Service activity trust can be calculated based on two aspects: the past activity records and past status transitions experiences.

Let success rate of a certain class of service \mathfrak{S} activity Ω^{class} be $p_{\mathfrak{S}}(\Omega^{class})$, and the overall success rate of certain class of activity Ω^{class} be $p(\Omega^{class})$. Then, the class trust (CT) of activity proposed by service \mathfrak{S} is calculated as:

$$CT_{\mathfrak{S}}(\Omega^{class}) = p_{\mathfrak{S}}(\Omega^{class}) \times p(\Omega^{class})^{[1-p(\Omega^{class})]} \quad (10)$$

In formula 10, we add an adjusting factor which can be calculated as $p(\Omega^{class})^{[1-p(\Omega^{class})]}$. This factor denotes creditable level of Ω^{class} . Formula 10 shows that the success rate is larger the class trust value is larger. It is similar with the real world's fact that more successful times an action is executed, the more confidence such action would gain.

For example, let service \mathfrak{S} has an activity Ω_1 . And it executes this activity Ω_1 with success rate of 0.95 in past. The overall success rate of activity in same class of Ω_1 which is executed by all existing services in CPS is 0.9. Then, the class trust of activity Ω_1 can be calculated as follows,

$$CT_{\mathfrak{S}}(\Omega_1^{class}) = 0.95 \times 0.9^{1-0.9} = 0.94$$

Let the previous statuses and post statuses proposed by a service \mathfrak{S} be $\Omega_{\mathfrak{S}}^{pre} = (\Delta_{\mathfrak{S}}^1, \Delta_{\mathfrak{S}}^2, \dots)$ and $\Omega_{\mathfrak{S}}^{post} = (\nabla_{\mathfrak{S}}^1, \nabla_{\mathfrak{S}}^2, \dots)$. In past decision making, activity Ω had the set of previous statuses and post statuses be $\Omega^{pre} = (\Delta^1 | q(\Delta^1), \Delta^2 | q(\Delta^2), \dots)$ and $\Omega^{post} = (\nabla^1 | q(\nabla^1), \nabla^2 | q(\nabla^2), \dots)$. Here, $q()$ represents the status occurrence rate. The status trust of service's activity is calculated as following:

$$ST_{\mathfrak{S}}(\Omega^{class}) = \sqrt{\frac{1}{2} \times \left(\frac{\sum_{i=1}^{|in(\Delta_{\mathfrak{S}}^i)|} q(\Delta_{\mathfrak{S}}^i)}{|in(\Delta_{\mathfrak{S}}^i)|} \right)^2 + \frac{1}{2} \times \left(\frac{\sum_{j=1}^{|in(\nabla_{\mathfrak{S}}^j)|} q(\nabla_{\mathfrak{S}}^j)}{|in(\nabla_{\mathfrak{S}}^j)|} \right)^2} . \quad (11)$$

where function $in(\Delta_{\mathfrak{S}}^i)$ denotes the status $\Delta_{\mathfrak{S}}^i \in \Omega_{\mathfrak{S}}^{pre}$ can be matched in Ω^{pre} , and $|in(\Delta_{\mathfrak{S}}^i)|$ is the number of $in(\Delta_{\mathfrak{S}}^i)$.

For example, let there be an activity $\Omega^{pre} = (\Delta^1 | 0.9, \Delta^2 | 0.95, \Delta^3 | 0.95, \Delta^4 | 0.8)$ and $\Omega^{post} = (\nabla^1 | 0.9, \nabla^2 | 1.0, \nabla^3 | 0.95)$. A service has this kind of activity with statuses $\Omega_{\mathfrak{S}}^{pre} = (\Delta_{\mathfrak{S}}^1, \Delta_{\mathfrak{S}}^2, \Delta_{\mathfrak{S}}^4)$ and $\Omega_{\mathfrak{S}}^{post} = (\nabla_{\mathfrak{S}}^2, \nabla_{\mathfrak{S}}^3)$. The ST value of this service's activity is calculated as following,

$$ST_{\mathfrak{S}}(\Omega^{class}) = \sqrt{\frac{1}{2} \times \left(\frac{0.9 + 0.95 + 0.8}{3} \right)^2 + \frac{1}{2} \times \left(\frac{1 + 0.95}{2} \right)^2} = 0.93$$

Based on class trust and status trust, activity trust can be calculation as following:

$$AT_{\mathfrak{S}}(\Omega) = \varpi_1 \times CT_{\mathfrak{S}}(\Omega) + (1 - \varpi_1) \times ST_{\mathfrak{S}}(\Omega) . \quad (12)$$

where, ϖ_1 is a weight with constraint $0 < \varpi_1 < 1$.

4.2. Belief Computation

CDM sponsor prefers to identifying a service with excellent past transaction experience. As a result, belief dependence can be calculated based on the past decision making transaction evaluations between CDM sponsor and the service.

Let a decision making service semantic be \mathfrak{S} , and it has made r times of decision for the CDM sponsor \mathfrak{R} . Let $judge^{\mathfrak{R}}(\mathfrak{S})_u (judge^{\mathfrak{R}}(\mathfrak{S})_u \in [0,1])$ denote the service's score of decision making from \mathfrak{R} . At the $r+1$ time, BD from \mathfrak{R} to party \mathfrak{S} is calculated as

$$BD_{r+1}^{\mathfrak{R}}(\mathfrak{S}) = \begin{cases} \frac{\sum_{u=1}^r judge^{\mathfrak{R}}(\mathfrak{S})_r}{r} & r \neq 0 \\ 0 & r = 0 \end{cases} \quad (13)$$

Similar as belief dependence, belief relationship reflects the whole creditable relationship between CDM sponsor and CPSSP. For d services in a CPSSP SP , if all the services have made t times of decisions, the belief relationship BR at $t+1$ time is

$$BR_{t+1}^{\mathfrak{R}}(SP) = \begin{cases} \frac{\sum_{v=1}^d BD^{\mathfrak{R}}(\mathfrak{S})_v}{t} & t \neq 0 \\ 0 & t = 0 \end{cases} \quad (14)$$

Utilizing summation or average of past evaluation to compute trust has been proven feasible and effective [36, 50]. Formula 13, 14 is proposed based on computing average value of past evaluations between two parties' interaction.

4.3. Reputation Ranking

Reputation denotes a public and authoritative trust belief from an adiaborous community. We build up an independent reputation ranking method to generate impartial reputations for CPSSPs.

Reputation of a CPSSP is the summation of evaluation scores from its all past decision making. Let CPSSP SP totally make h times of decision with evaluation score $judge(SP)$ for past decision making. Reputation ranking of SP can be calculated as follows:

$$RR(SP) = \frac{\sum_{s=1}^h judge(SP)_s}{h} \quad (15)$$

Likewise, we introduce computing average value of past evaluations for reputation ranking in formula 15. Different with belief computation, reputation

ranking is based on all the past evaluations from parties who had interactions with CPSSP in past rather than just only between two parties. It means that reputation is an objective view from whole community, while belief is a subjective relationship between two individuals.

We utilize three factors for reputation ranking, i.e., time limitation, source identity, and ranking delay. In our previous work, we calculated the reputation from above three factors, which has been testified feasible and effective in our work [38]. In this paper, we modify the corresponding formulas according to the features of CPS.

Time limitation (TL). We authorize CPSSP a unit time called time limitation. In this time period CPSSP can only receive one appointed number of decision making evaluation from the same CDM sponsor. Time limitation can reduce the risk that vicious CPSSP issues repetitious evaluation scores to cheat well-deserved reputation.

Let the appointed number of evaluation be TL_n , and the number of decision making be TL_t . Then the reputation ranking $RR_{TL}(SP)^{TL_{unit}}$ generated by CPSSP A is calculated as follows:

$$RR_{TL}(SP)^{TL_{unit}} = \frac{\sum_{j=1}^{TL_t} \sum_{i=1}^{TL_n} judge(SP)_{ij}}{i} . \quad (16)$$

where $judge(SP)_{ij}$ denotes the evaluation score to CPSSP A in a unit of time TL_{unit} , i denotes the appointed number and j denotes the different decision making.

Source identity (SI). Reputation ranking should bind with the evaluation source sponsor's reputation. An evaluation from a source sponsor with a higher reputation generally has more impacts to the receiver CPSSP.

If a CPSSP SP_2 has reputation ranking $RR(SP_2)$, and it sends an evaluation score $judge(SP_1)$ to CPSSP SP_1 , SP_1 will get the evaluation score as follows:

$$RR_{SI}(SP_1) = judge(SP_1) \times (RR(SP_2))^{[1-RR(SP_2)]} . \quad (17)$$

Similar with formula 10, $(RR(SP_2))^{(1-RR(SP_2))}$ is an adjusting factor which relies on source CPSSP's reputation $RR(SP_2)$. And the value of factor increases with the value increasing of $RR(SP_2)$.

Ranking delay (RD). To determine that the new evaluation score is not a fake or inauthentic evaluation, we adopt a delay period mechanism. In delay period, reputation is just a temporary result ($RR_{RD}(SP)$), and such reputation ranking can be withdrawn when it is identified as any illegal or cheating trick.

Let the time for reservation of the evaluation score in delay period be RD_t , and the whole length of delay period be RD_l . The temporary ranking can be expressed as:

$$RR_{RD}(SP) = \frac{judge(SP) \cdot RD_t}{RD_l} . \quad (18)$$

From above three factors, the reputation ranking $RR(SP)^{T+t}$ can be defined as:

$$RR(SP_1)^{T+t} = RR(SP_1)^T + \sum_{j=1}^m \sum_{i=1}^n \frac{judge(SP_1) \times (RR(SP))_{ij}^{1-RR(SP)_{ij}} \times I}{i \times t} . \quad (19)$$

where T is a time point, t is the delay period¹.

4.4. Physical Trust Computation

CPSSPs communicate with each other through physical devices in relax coupling network. Physical trust computation aims to identify which physical devices are legitimate and which are not to be trusted. The threat of the fault tolerance of devices, the healthiness of devices must be considered.

Let CPSSP's device totally success rate of providing service in past be $suc(SP)$, and the fault rate occurred in past be $fault(SP)$. At the same time, the rate of CPSSP's recovering from the faults be $recover(SP)$. The fault tolerance trust (FT) of CPSSP's device can be calculated as follows:

$$FT(\mathfrak{S}^{id}) = suc(SP)^{1-fault(SP)} \times recover(SP)^{fault(SP)} . \quad (20)$$

Let the ratio of whole running period of CPSSP's device be t_1 units of time, the whole sickness period caused by device faults or connection troubles be t_2 units of time. The average sickness period and the average

¹ In this paper, we utilize unit of time to measure as the length of time. Here, we define that the length of unit of time TL_{unit} is the same as the delay period t .

mending period in sickness period be t_3 units of time and t_4 units of time². The healthiness trust of CPSSP's device can be calculated as follows:

$$HT(\mathfrak{S}^{id}) = \left(1 - \frac{t_2}{t_1}\right)^{1 - \frac{1}{t_2}} \times \left(1 - \frac{t_4}{t_3}\right)^{\frac{1}{t_4}} . \quad (21)$$

Based on the fault tolerance trust and healthiness trust mention above, physical trust of CPSSP's device can be defined as:

$$PT(\mathfrak{S}^{id}) = \varpi_2 \times FT(\mathfrak{S}^{id}) + (1 - \varpi_2) \times HT(\mathfrak{S}^{id}) . \quad (22)$$

where, ϖ_2 is a weight with constraint $0 < \varpi_2 < 1$.³

4.5. Recommended Trust Relationship Computation

In an open network environment, it is impossible for the CDM sponsor to understand all of the various CPS services. To understand the strange CPS services, the sponsor can utilize the recommendations from their acquaintances. As a result, we introduce a recommended trust to initialize the relationship between CDM sponsor and strange CPSSP. Recommended trust is built up through an intermediate CPSSP that has beliefs with both CDM sponsor and the strange CPSSP.

For CDM sponsor \mathfrak{R} and two CPSSPs SP_1 , SP_2 , if $BR^{\mathfrak{R}}(SP_2) = 0 \wedge BR^{\mathfrak{R}}(SP_1) \neq 0 \wedge BR^{SP_1}(SP_2) \neq 0$ and $\exists \mathfrak{S} \in SP_2 \wedge BD^{\mathfrak{R}}(\mathfrak{S}) \neq 0$, the recommended trust ($RT^{\mathfrak{R}}(SP_2, \mathfrak{S})$) is:

$$RT^{\mathfrak{R}}(SP_2, \mathfrak{S}) = \alpha(BR^{\mathfrak{R}}(SP_1)) + \beta(BR^{SP_1}(SP_2)) + \gamma(BD^{SP_1}(SP_2, \mathfrak{S})) . \quad (23)$$

where α , β , and γ are parameters which are set by the system to demonstrate the importance degrees of different trust values for recommended trust.⁴

For CDM sponsor, recommended CPSSP is an unfamiliar service provider with full confidence. So we propose a confidence conformation factor for recommended CPSSP based on objective reputation with impartial nature. We suppose that there are d intermediary CPSSPs SP_i^{in} recommending

² Here, the length of each unit of time is same as the length of delay period in formula 19.

³ In this paper, the parameters of weight σ_i , ϖ_1 and ϖ_2 in formula 9,12 and 22 are given by system in advance.

⁴ $\alpha, \beta, \gamma \in [0, 1], \alpha + \beta + \gamma = 1$

same CPSSP SP to CDM sponsor. The confidence conformation factor $\phi(SP)$ of the recommended CPSSP SP is as follows:

$$\phi(SP) = \frac{(\sum_{i=1}^d (RT^{\mathfrak{R}}(SP) \cdot RR(SP_i^{in}))) \cdot RR(SP)}{\sqrt{\sum_{i=1}^d (RT^{\mathfrak{R}}(SP)_i \cdot RR(SP_i^{in}))^2} \times \sqrt{\sum_i (RR(SP))^2}} \quad (24)$$

In our consideration, confidence conformation factor $\phi(SP)$ aims to show the similarity between recommendation trust and recommended CPSSP's reputation. So formula 24 is proposed based on Cosin method which is widely used to calculate similarity between two vectors. We give an example here to present our formula. Let there be 3 CPSSPs SP_1 , SP_2 and SP_3 who recommend same CPSSP SP_4 to sponsor \mathfrak{R} . Related values of trust and reputation are given in table 3. According to formula 23 and 24, we can calculate the recommended trust ($RT^{\mathfrak{R}}(SP_2, \mathfrak{S})$) and confidence conformation factor $\phi(SP)$ as follows,

Table 3. Example of recommended trust

	$BR^{\mathfrak{R}}(SP)$	$RR(SP_i)$	$BR^{SP_i}(SP_4)$	$BR^{SP_i}(SP_4)$	$RR(SP_4)$	RT	$\phi(SP)$
SP_1	0.9	0.95	0.95	0.9		0.955	
SP_2	0.8	0.85	1.0	1.0	0.95	0.94	0.99
SP_3	0.9	0.95	0.9	1.0		0.94	

Here, we appoint the parameters value of α , β , and γ as 0.3, 0.3 and 0.4 respectively.

5. Service Selection Negotiation for CPSSP

In order to make the best decision, CDM sponsor always wants to find the most competent services. Capacity and trust represent two critical aspects for candidate services. Our service selection mechanism is based on the principles of capacity and trust.

CDM sponsor selects desirable services from candidate services based on the CPSSPs' applications for decision tasks. As a result, negotiation between sponsor and CPSSP is a feasible solution. Negotiation would render both CDM sponsor and CPSSPs opportunities to query, discuss, explain or revise the decision tasks.

First, we define a set of message primitives for negotiation as follows.

- $send()$: sending a message.
- $reject(a,b)$: informing to reject event a and sending event b .
- $Send_value(a,value)$: sending the value of event a .
- $Accept()$: sending a set of acceptable events to the other.
- $revise(a,b)$: revising the event a and modifying it to event b .
- $query(a)$: querying the state of event a .

Our service negotiation selection mechanism consists of 12 steps as follows:

Step1: CDM sponsor decomposes the complex decision problem according to the structure relationship in semantic $\aleph.SR$ and forming sub-problems semantic sub_S_i of \aleph .

Step2: CDM sponsor sends sub_S_i to CPSSPs who have belief relationships $BR^{st}(SP_k) \geq \vartheta$ through primitive $send(sub_S_i)$ ⁵. And CPSSPs who receive the sub_S_i also transmit sub_S_i to sponsor's strange CPSSPs with well-deserved belief relationships.

Step3: While CPSSPs receive sub_S_i , they reply CDM sponsor whether the tasks would be accepted. If sub_S_i is acceptable, CPSSP would send a message $Accept(sub_S_i)$ to CDM sponsor. Otherwise, CPSSP would send a message $reject(sub_S_i)$ to CDM sponsor to inform that CPSSP would surrender the opportunity to take part in sub_S_i .

Step4: If a CPSSP wants to recommend another CPSSP SP to CDM sponsor, it uses $send(SP.S)$ to send message and recommend the service S of CPSSP SP to CDM sponsor. While sponsor receives such recommendation, it will query the recommended CPSSP SP through primitives $query(SP.S)$ and $send(sub_S_i)$ to inform decision task and confirm if SP would take part in CDM.

Step5: All the affirmative services from different CPSSP SP_k send their service semantics through $send(S_j)$ to CDM. For each candidate service semantics S_j , sponsor computes evaluation scores of $capacity(S_j)$, $AT(S_j)$, $BD^{st}(S_j)$, $RR(SP_k)$, and $PT(S_j)$. Moreover, sponsor computes scores of $RT^{st}(S_j)$ for the recommended services.

⁵ Here, ϑ , ζ and ρ are thresholds given in advance.

Step6: Sponsor selects candidate services \mathfrak{S}_j for each sub_S_i with $capacity(\mathfrak{S}_j) \geq \zeta$. If no \mathfrak{S}_j is selected for a decision task sub_S_i , sponsor selects the \mathfrak{S}_j who has the maximum value of $capacity(\mathfrak{S}_j)$. The selected services are in a set Γ .

Step7: Sponsor sends messages $reject(\mathfrak{S}_j)$ to the CPSSPs whose services are not in Γ .

Step8: For service in set Γ , sponsor sends messages with primitive $revise(\mathfrak{S}_j^{td}, plan)$ to CPSSPs to ask for the detailed revising plans.

Step9: Upon revising claims, CPSSPs will determine whether modify their plans. If CPSSP modify the plan, it sends the new plan with $revise(\mathfrak{R}, plan)$ to sponsor. Otherwise, it sends the rejection claim $reject(\mathfrak{R}, plan)$ to sponsor.

Step10: Sponsor repeats the negotiation steps 8 and 9 until at least one service in set Γ modify its plan.

Step11: Sponsor re-computes all $capacity(\mathfrak{S}_j)$ of services in set Γ after negotiation and selects the services \mathfrak{S}_j satisfying constraint $capacity(\mathfrak{S}_j) + AT(\mathfrak{S}_j) + BD^{\mathfrak{R}}(\mathfrak{S}_j) + PH(\mathfrak{S}_j) \geq \rho$ or $capacity(\mathfrak{S}_j) + AT(\mathfrak{S}_j) + RT^{\mathfrak{R}}(\mathfrak{S}_j) + PH(\mathfrak{S}_j) \geq \rho$. For the services that do not satisfy the constraint, sponsor rejects them and removes them out of set Γ .

Step12: For each decision sub-problems sub_S_i , sponsor selects the services \mathfrak{S}_j as the final victor with the maximum reputation values of CPSSP $RR(SP_k)$. If the selected service is a recommended one, sponsor computes its confidence factor $\phi(SP)$. If $\phi(SP)$ is acceptable, sponsor ascertains that the recommended service is victor. Otherwise, sponsor selects the second highest value of reputation.

6. Proposed Framework

In summary, we propose a framework of service selection for CDM in CPS. Our service selection mechanism is shown in Figure 2. In the figure, blue lines indicate the releasing of sponsor decision making task semantics, the dashed lines indicate the negotiation between sponsor and CPSSPs, and red lines indicate the services from CPSSPs in selection process. As shown in Figure 2, there are three phases, which are the semantic based capacity evaluation for CDM sponsor, trust computation of CPS, and the negotiation selection of CPSSP. In the first phase, the formal semantic of complex

A Novel Capacity and Trust Based Service Selection Mechanism for Collaborative Decision Making in CPS

decision task is described through ontology in sponsor machine. Each CPSSP analyzes the decision task semantic and generates participator instruction according to its service capacity and physical environment automatically. Moreover, CPSSP sends the participator instruction to the CDM sponsor. In the second phase, trust computation is launched when the sponsor receives all the participator instructions from CPSSP. Trust computation consists of two steps: trust evaluation for virtual software service and trust evaluation for physical service objects. In the last phase, CDM sponsor would negotiate with CPSSPs and identify the most competent CPSSP participants through trust and capacity criterions.

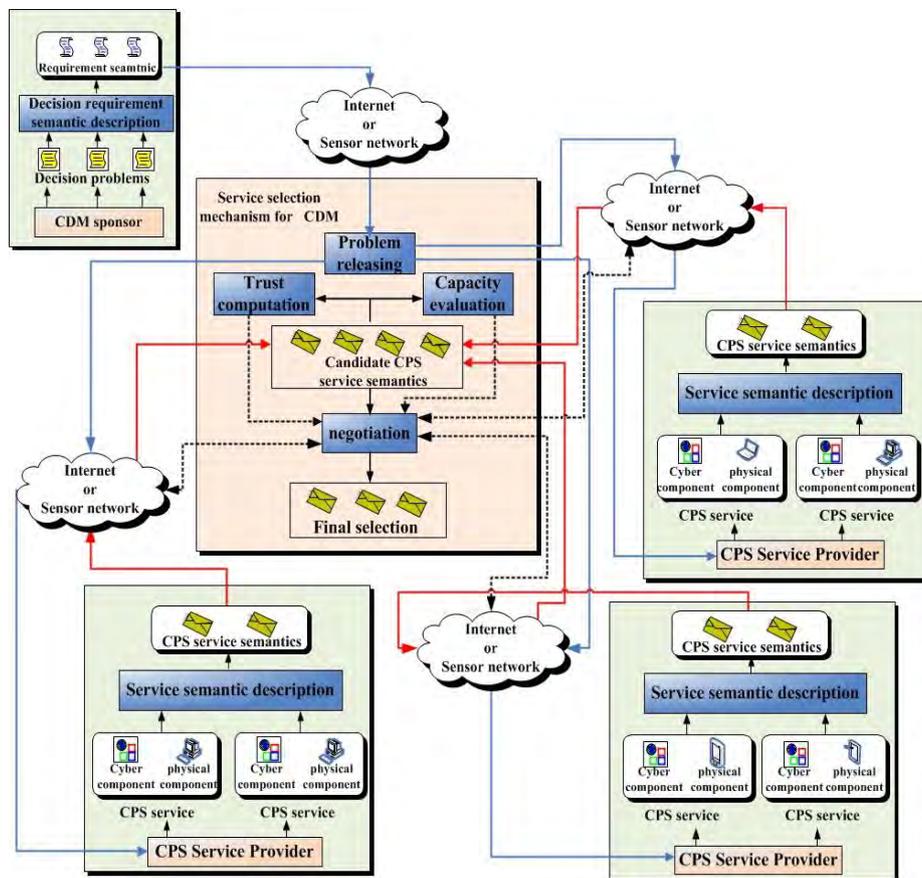


Fig.2. Service selection mechanism from CPSSP for CDM

CPS service selection framework aims to enable CDM sponsor to identify their desirable service from candidates automatically in virtual and physical environment. It comprises 6 elements as follows.

- 1) Semantic description is responsible for representing semantic of sponsor's requirements and CPS's services.

2) Asynchronous message communication is responsible for the e-communication among organizations.

3) Capacity evaluation of CPS's service provides the capacity reference value.

4) Trust computation offers trust reference value.

5) Negotiation activities are the protocols for negotiation during decision making. The negotiation is divided into 6 steps: handshake negotiating, releasing of user's requirements, negotiating conformation of requirements, negotiating exchange of service plans, and negotiating revising of plans, final service selection.

6) Other functions mainly include service rules, data storage, physical environments watch, and device log maintenance.

Currently, in order to exhibit and examine the effects of our framework, we are working on the implementation of a real-world application: smart connected cars. We utilize the capacity and trust evaluation of cyber software and physical device proposed in our framework of this paper to select CPSSP, such as car or on board device that best fits the service requirements. This work can be used in smart vehicle scheduling or smart traffic controlling.

7. Conclusion

CDM-based CPS service now faces the embarrassment to identify the most competent services from candidate sets due to insufficient prophetic knowledge for a specified decision making. In this paper, we utilize the capacity and trust computation for the service selection. Our methodology comprises three phases. First, capacity evaluation of decision service is achieved based on formal semantic description of decision problem and CPS service's software as well as physical characteristics. Second, we address the trust computation composed by service's software activity, subjective belief trust, objective reputation, physical trust and recommended trust. Based on above two criteria, we present an automatic negotiation framework for service selection. Our future works will focus on the challenges that have not been discussed in this paper as following:

- Measure the trust value for the dynamic environments and statuses of cyber and physical components in CPS.
- Measure the service's activity trust based on the nature of activity, just like logic, motivation or consistency of activity.
- Define how trust evolves in a dynamic setting.

Acknowledgements. This work is funded by "China Postdoctoral Science Foundation funded project" (20110490697). This work is also supported by the National Natural Science Foundation of China (61103069, 60903032, and 71171148). We appreciate all reviewers for their comments. We would thank for Dr. Ziming Zheng of Illinois Institute of Technology, for providing useful insights.

References

1. Arribas, J., Cid-Sueiro, J.: A Model Selection Algorithm for a Posteriori Probability Estimation With Neural Networks. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, Vol. 16, No. 4, 799-809. (2005)
2. Vasant, D., Roger, S.: Intelligent Decision Support methods. The science of Knowledge Work. Printice Hall, New Jersey, USA, 171-180. (1997)
3. Herbert, J. P., Yao, J.: Rough Set Model Selection for Practical Decision Making. In *Proceedings of Fourth International Conference on Fuzzy Systems and Knowledge Discovery*. Haikou, Hainan, China, 203 – 207. (2007)
4. Tan, Y., Goddard, S., Perez, L. C.: A prototype architecture for cyber-physical systems. *ACM SIGBED Review*, Vol. 5. No.1, 1-2. (2008)
5. Balasubramanian, J., Tambe, S., Gokhale, A., Dasarathy, B., Gadgil, S., Schmidt, D.: A model-driven QoS provisioning engine for cyber physical systems. Technical Report, Vanderbilt University and Telcordia Technologie. (2010).
6. Tan, Y., Vuran, M. C., Goddard, S.: Spatio-temporal event Model for cyber-physical systems. In *Proceedings of the 29th IEEE IntConf on Distributed Computing Systems Workshops*. Montreal, Québec, Canada. 44-50. (2009)
7. Tidwell, T., Gao, X., Huang, H.: Towards configurable real time hybrid structural testing: A Cyber-Physical Systems approach. In *Proceedings of IEEE International Symposium on Object Component Service-Oriented Real-Time Distributed Computing*, Tokyo, Japan. (2009)
8. Lee, E.: Cyber Physical Systems: Design Challenges. In *Proceedings of 2008 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing*, Orlando, Florida, USA, 363-369. (2008)
9. Dabholkar, A., Gokhale, A.: An approach to middleware specialization for cyber physical systems. In *Proceedings of 29th IEEE International Conference on Distributed Computing Systems Workshops*. Montreal, Québec, Canada, 73-79. (2009)
10. Antsaklis, P.: On control and cyber-physical systems: challenges and opportunities for discrete event and hybrid systems. *9th International Workshop on Discrete Event Systems*, Goeteborg, Sweden. (2008)
11. Ahmadi, H., Abdelzaher, T. F., Gupta, Indranil.: Congestion control for spatio-temporal data in cyber-physical systems. In *Proceedings of 1st ACM/IEEE International Conference on Cyber-Physical Systems*, Stockholm, Sweden, 89-98. (2010)
12. Cardenas, A. A., Amin, S., Sastry, S.: Secure control: Towards survivable cyber-physical systems. In *Proceedings of the 28th Int Conf on Distributed Computing Systems Workshops*. Beijing, China. 495-500. (2008)
13. Adam, N.: Cyber-physical systems security. *Proc of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*. Oak Ridge, TN, USA. (2009)
14. Tang, H., Mcmillin, B. M.: Security property violation in CPS through timing. In *Proceedings of the 28th Int Conf on Distributed Computing Systems Workshops*. Beijing, China, 519-524. (2008)
15. Xia, F., Ma, L.H., Dong, J.X., Sun, Y.X.: Network QoS management in cyber-physical systems. In *Proceedings of Int Conf on Embedded Software and Systems Symposia*. Chengdu, Sichuan, China, 302-307. (2008)
16. Kottenstette, N., Karsai, G., Sztipanovits, J.: A passivity-based framework for resilient cyber physical systems. *Proc of the 2nd Int Symp on Resilient Control Systems*, Idaho Falls, Idaho, USA, 43-50. (2009)

17. Kang, K. D., Son, S. H.: Real-time data services for Cyber Physical Systems. In Proceedings of 28th International Conference on Distributed Computing Systems Workshops. Beijing, China. (2008)
18. Carlson, M.: Exploiting the opportunities of collaborative decision making: A model and efficient solution algorithm for airline use. *Transportation Science*, Vol. 34, No. 4, 381-393. (2000)
19. Eriksen, P.: Collaborative decision making information management in airports. In Proceedings of AIAA/IEEE Digital Avionics Systems Conference. Irvine, CA, 7B41-7B48. (2002)
20. Boroushaki, S., Malczewski, J.: Measuring consensus for collaborative decision-making: A GIS-based approach. *Computers, Environment and Urban Systems*, Vol. 34, No. 4, 322-332. (2010)
21. Dreelin, A., Rose-Joan, B.: Creating a dialogue for effective collaborative decision-making: A case study with Michigan stakeholders. *Journal of Great Lakes Research*, Vol. 34, No. 1, 12-22. (2008)
22. Chim-Mei, Y., Anumba-Chimay, J., Carrillo-Patricia, M.: Internet-based collaborative decision-making system for construction. *Advances in Engineering Software*, Vol. 35, No. 6, 357-371. (2004)
23. Wang, L., Cheng, Q.: Web-based collaborative decision support services for river runoff and flood risks prediction in the Oak Ridge Moraine Area, Canada. In Proceedings of International Society for Optical Engineering. Orlando, FL, USA. (2006)
24. Indiramma, M., Anandakumar, K.R.: Collaborative decision making framework for multi-agent system. In Proceedings of the International Conference on Computer and Communication Engineering 2008, Kuala Lumpur, Malaysia, 1140-1146. (2008)
25. Barhamgi, M., Benslimane, D., Ouksel-Aris, M.: PWSMS: A peer-to-peer web service management system for data sharing in collaborative environments. *Computer Systems Science and Engineering*, Vol. 23, No. 2, 89-106. (2008)
26. Mou, Y., Cao, J., Zhang, S., Zhang, J.: Interactive Web service choice-making based on extended QoS model. *Journal of Zhejiang University: Science*, Vol. 7, No. 4, 483-492. (2006)
27. <http://www.w3.org/TR/ws-gloss/>
28. Jim, P., Joannis, H.: Integrating Hybrid Rule-Based with Case-Based Reasoning, Proceedings of Advances in Case-Based Reasoning 6th European Conference, Aberdeen, Scotland, UK, 336-349. (2002)
29. Yang, S., Ni, Z.: *Machine Learning and Intelligent Decision Support System*. Science Publisher, Beijing, China. (2004)
30. Malcolm, R.F.: Key Concepts in Model Selection: Performance and Generalizability. *Journal of Mathematical Psychology*, Vol.44, No.1, 205-231. (2000)
31. Hammadi, A., Hussain, F.K., Chang, E., Dillon, T.S., Ali, S.: Ontological framework for trust & reputation for DBE. In Proceeding of 2nd IEEE International Conference on Digital Ecosystems and Technologies. Phitsanulok, Thailand, 650-653. (2008)
32. Vijayakumar, V., WahidhaBanu, R. S. D.: Trust and Reputation Aware Security for Resource Selection in Grid Computing. In Proceeding of International Conference on Security Technology, Sanya, Hainan Island, China, 121-124. (2008)
33. De, O., Albuquerque, R., Cohen, F.F., Mota, J., de Sousa, R.T.: Analysis of a Trust and Reputation Model Applied to a Computational Grid Using Software Agents. In Proceeding of International Conference on Convergence and Hybrid Information Technology. Busan, Korea, 196-203. (2008)

34. Jøsang, A., S. Lo, P.: Analysing the Relationship between Risk and Trust. In Proceedings of the Second International Conference on Trust Management. Oxford, UK. (2004).
35. Resnick, P., Zeckhauser, R., Friedman R., Kuwabara, K.: Reputation Systems. Communications of the ACM, Vol.43, No.12, 45-48. (2000)
36. Schneider, J.: Disseminating Trust Information in Wearable Communities. In Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing, London, UK. (2000)
37. Withby, A., Jøsang, A., Indulska, J.: Filtering Out Unfair Ratings in Bayesian Reputation Systems. In Proceedings of the 7th Int. Workshop on Trust in Agent Societies, New York, USA. (2004)
38. Zhang, B., Xiang, Y.: Trust and Reputation based Model Selection Mechanism for Decision-making, In Proceedings of 2th International Conference on Networks Security, Wireless Communications and Trusted Computing. Wuhan, Hubei, China, Vol.2, 14-17. (2010)
39. Zhang, B., Xiang, Y., Xu, Q.: Evaluation Mechanism for Web Service based on Semantics and Reputation. In Proceeding of 6th International Conference on Wireless Communications, Networking and Mobile Computation, Chengdu, Sichuan, China. (2010)
40. La, H., Kim, S.: A Service-Based Approach to Designing Cyber Physical Systems. In Proceedings of 2010 IEEE/ACIS 9th International Conference on Computer and Information Science, Kaminoyama, Japan, 895-900. (2010)
41. Chakraborty, S., Ray, I.: p-Trust: A New Model of Trust to Allow Finer Control over Privacy in Peer-to-Peer Framework. Journal of Computers, Vol.2, No.2, 13-24. (2007)
42. Beth, T., Borcherding, M., Klein, B.: Valuation of Trust in Open Networks. In Proceedings of the 3rd European Symposium on Research in Computer Security, Lecture Notes in Computer Science, vol. 875 Springer-Verlag, Brighton, UK. (1994)
43. Bertino, E., Ferrari, E., Squicciarini, A.: Trust-X: A Peer to Peer Framework for Trust Establishment. IEEE Transactions on Knowledge and Data Engineering, Vol.16, No.7, 827-842. (2004)
44. Yu, T., Winslett, M.: A Unified Scheme for Resource Protection in Automated Trust Negotiation. In Proceedings of the IEEE Symposium on Security and Privacy. Berkeley, CA, USA. (2003).
45. English, C., Nixon, P., Terzis, S., McGettrick, A., Lowe, H.: Dynamic Trust Models for Ubiquitous Computing Environments. In Proceedings of the 4th International Conference on Ubiquitous Computing, Göteborg, Sweden. (2002)
46. Kagal, L., Finin, T., Joshi, A.: Trust Based Security in a Pervasive Computing Environment. IEEE Computer, Vol. 34, No. 12, 154-157. (2001)
47. Chakraborty, S., Poolsappasit, N., Ray, I.: Reliable Delivery of Event Triggered Obligation Policies from Sensors to Actuators in Pervasive Computing Environments. In Proceedings of the 21st Annual IFIP TC-11 WG 11.3 Working Conference on Data and Applications Security, Redondo Beach, CA, USA. (2007)
48. Ganeriwal, S., Srivastava, M.: Reputation-Based Framework for High Integrity Sensor Networks. In Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks. Washington, DC, USA., 66-77. (2004)
49. Jin, J., Ahn, G. J.: Role-based Access Management for Ad-hoc Collaborative Sharing. In Proceedings of the 11th ACM Symposium on Access Control Models and Technologies. Lake Tahoe, California, USA, 200-209. (2006)
50. Jøsang, A., Ismailb, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems, Vol. 43, No. 2, 618-644. (2007)

Bo Zhang, Yang Xiang, Peng Wang, and Zhenhua Huang

Bo Zhang is a postdoctoral in the College of Electronics and Information Engineering at Tongji University. His research interests include cyber-physical system, wireless network, trust computation, semantic web, and data mining. He has a PhD in computer science from the Tongji University. He is now a director of trust computation project, which is funded by National Nature Science Foundation of China.

Yang Xiang is a professor of intelligent information system in the College of Electronics and Information Engineering at the Tongji University. His research interests include cyber-physical system, trust computation, intelligent decision support system, semantic web, and data mining. He is director of data mining project, which is funded by National Nature Science Foundation of China.

Peng Wang is a PHD candidate in the College of Electronics and Information Engineering at Tongji University. His main research interests include cyber-physical system and web service.

Zhenhua Huang is a senior lecture in the College of Electronics and Information Engineering at Tongji University. His main research interests include web service, database and data mining. He is director of web service project, which is funded by National Nature Science Foundation of China.

Received: February 25, 2011; Accepted: April 25, 2011.

Privacy Preserving in Ubiquitous Computing: Classification & Hierarchy

Tinghuai Ma¹, Sen Yan², Jin Wang³, and Sungyoung Lee³

¹Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology,
210044 Nanjing, China
thma@nuist.edu.cn

²School of Computer & Software, Nanjing University of Information Science & Technology,
210044 Nanjing, China
perhaps2532@gmail.com

³Department of Computer Engineering, Kyung Hee University,
446701, Suwon, South Korea
{wangjin,sylee}@oslab.khu.ac.kr

Abstract. In this paper, we adopt the classification of personal information and hierarchy of services to build a privacy system, in which one communicates with each other via pipes with different security levels. In each level, one has the corresponding rights to access each other. The requesters are not able to be infringed based on the personal information that service obtains from service providers. Privacy system can decrease the interaction, while in other circumstance the system strengthens and enhances the privacy preserving. Thus we strike a balance between two goals of Ubiquitous Computing: interaction and privacy preserving.

Keywords: ubiquitous computing, privacy preserving, classification, hierarchy.

1. Introduction

Ubiquitous computing represents the concept of seamless „everywhere“ computing and aims at making computing and communication essentially transparent to users. In ubiquitous computing, we will be surrounded by a comfortable and convenient information environment that merges physical and computational infrastructures into an integrated habitat [1]. Context-awareness will allow this habitat to take on the responsibility of serving users by tailoring itself to their preferences as well as performing tasks and group activities according to the nature of the physical space. So, the more an application is aware of the user’s context, the better it can adapt itself to assist him. Richer contextual and deeper personal information facilitates better automation and adaptation [2], [33]. Unfortunately, transparent soliciting, acquiring and

handling of personalization constructs raise privacy concerns [33]. Thus, privacy in ubiquitous computing has been a contentious issue. The privacy concerns have been raised to suggest that privacy may be the greatest barrier to the long-term success of ubiquitous computing [3]. The privacy preserving in ubiquitous computing becomes a hot topic recently [25], [30].

There are two privacy preserving techniques in ubiquitous computing, one is the anonymous / pseudonym oriented, and the other is the policy based.

For the anonymous based privacy enhancing technology (PET), there are k -anonymities that they can not be distinguished from each other. In other words, there are k entities having the same ID or in the same location at the same time. The higher value of k , the higher level of anonymity is [4]. The service provider doesn't want to misuse of data because there is a mixture of true and $k-1$ sensor data. Even though there are k IDs, the privacy sensitive information may be revealed to an attacker through track analysis. The combination of frequently varying pseudonyms and dummy traffic is used to prevent it [5]. For some ID based services, the virtual identities are used to conceal the user's real identity [6]. Anonym and authentication are always conflict. Diep et al [7] present a scheme using anonymous user ID, sensitive data sharing method, and account management to provide a lightweight authentication while keeping users anonymously interacting with the services in a secure and flexible way. He et al [8] use the blind signature to encrypt anonymous ID, which can support the anonymous ID's authority.

For the policy based privacy enhancing technology, the main concept is to store privacy-compliant rules to process personal information. The stored privacy policies describe the allowed recipients, uses, and storage duration of users' data. Also, a policy engine is used to reason the compatible privacy policy. Now, privacy policy is described in XML [9], [10] and XACML [11], [27], [28], [30], [31]. Most privacy policy based PET are focus on one or more scenarios. The configuration of policy for each scenario is a huge work [12]. The management and deducing are the main concerns in policy based system [13], [14], [15], [16]. Actually, the policy based PET is using access control model to make sure the security of personal data and service. Policy based PET is simply, intuitive, but not easy to deployment in wide area.

Using XML or XACML to describe accessing control rule is the main idea of policy based PET's. It focuses largely on conventional data management schemes to support user's privacy preserving. With different situations and user-specific privacy granularity, the privacy preserving rules configuration is exhaustively. Reducing the rule number is helpful to configure the whole privacy preserving system.

The goal of this paper is to design a possible way to accomplish least interactions between services and users, and thus accentuate "seamlessness" between virtual world and physical world, making services transparent to users anywhere anytime, making users immerse to services anywhere anytime. The proposed system in this paper offers a new secure approach to decrease the interactions. Compared to the existed ones, this system is suitable for a small-scaled area. In this paper, we classify the user and service into three

types separately. Then, the rules, which indicate what data can be accessed, will be simpler. It will simplify huge rule configuration into several rules setting.

The remainder of this paper is organized as follows: Section 2 presents the related work about policy based PETs. Then we present our new mechanism about how to tackle privacy in the system in section 3. We introduce some policies and strategies that will be of great help and integrate them as a policy-making system in Section 4, also the practical analyses are conducted. In section 5, an integrated privacy system is described, and an example system is implemented to demonstrate the performance. Finally, we concluded the whole paper in Section 6.

2. Related Work

There have been many works in the area of privacy preserving in ubiquitous computing [17], [23], [24]. They emphasized the importance of privacy preserving, clarified the problems that we must not ignore. Most of the work focuses on how to express privacy preserving policies and the architectures. But these solutions have not addressed the policies configuration automation and the configuration efficiency.

Approaches closely related to our work have been investigated in two different areas: XACML based policy configuration and access classification.

XACML based policy. XACML is an XML-based language which specifies access control policies, which is called the extensible Access Control Markup Language. Policy based PET focuses on controlling the data leakage, in other words, configuring the rule what data can be accessed.

In service access control model, Dai et al. [31] propose an access control model based on XACML in web services, which provides the process of access control for the requesters and services in web services. Malik et al. [32] use XACML to describe the access control rules which require prioritization and conflict handling mechanism. It is used for control web service sharing, which preserving the privacy of personal context and shared context.

For data access control model, XACML also acts an important role, especially for health data. Arunkumar et al. [30] use XACML policy to control the privacy and data access through handheld devices. Ardagna et al. [28] combine the XACML and SAML to enable privacy-preserving and credential-based access control. Kodeswaran et al. [29] propose a framework to allow users to control how their data is used, and extend existing access control languages to enable researchers to use the aggregate data avoiding privacy leakage.

Access classification. Policy configuration is a huge work [12]. Automated policy configuration is the destination of policy based PET. But it is on the way for this scenario coming. Decreasing the rule number may reduce configuration pressure. First, the privacy data are classified into groups. Ning et al. [26] design a documents broadcasting approach, which is based on access control policies specifying which users can access which documents, or subdocuments. Second, Kodeswaran et al. [29] classify the access process into complete access, abstract access and statistical access. Then, policy infrastructure is described, which proves individual privacy is protected.

3. Classification & Hierarchy

In accordance with the previous fundamentals, we advance some new concepts about privacy preserving in Ubiquitous Computing.

3.1. Classification of Personal Information

Definition 1: Classification of Personal Information. Classification is taxonomy. According to the relevance to a real entity, information of such entity can be classified as:

1. Direct information, the part which can be directly referred to a person without any reasoning, such as name, ID number, address.
2. Indirect information, the part which might be referred to a person with reasoning, or have some relationship with a person's privacy, such as medical history.
3. Relevant information, the part which cannot be referred to a person, such as hobbies.

Most of us are familiar with Object Oriented Programming languages (OOP), one of its core ideas is that it introduces a concept of Class. The objects are encapsulated as three types or levels: Public, Protected and Private. Our idea of Classification is derived from this. From the previous work we have done, personal data comprises the identity and the profile two parts. We might categorize the information as Private Level (PriL), Protected Level (ProL) and Public Level (PubL), respectively. Here we draw an intuitive figure to illustrate our structure as Fig.1.

On the basis of Fig.1, Table I is listed, which presents the rights of corresponding service type. Full means accessing information without interaction is allowed. Limited means when necessary, it will need mandatory interaction to get permission to proceed. No means no permission and interaction, individual devices deny automatically. The data flow arrows in the Fig.1 represent the directions of access rights, that is, the back-end of the arrow can access the fore-head of the arrow. Generally speaking, the system grants high level rights to access the low level. In the system defined area column, since the data is strictly defined by the system, the access rights are

thus strictly granted from high to low. In the user defined area column, there are some distinctions: considering the personalization of individual database, between the Private and Protected level, we grant the latter one the limited rights to access the former one with some kind of responses to the entity it affiliates such as alarm, vibration, sound, to remind the user to decentralize the accessing rights, so that it can proceed and collect the necessary information.

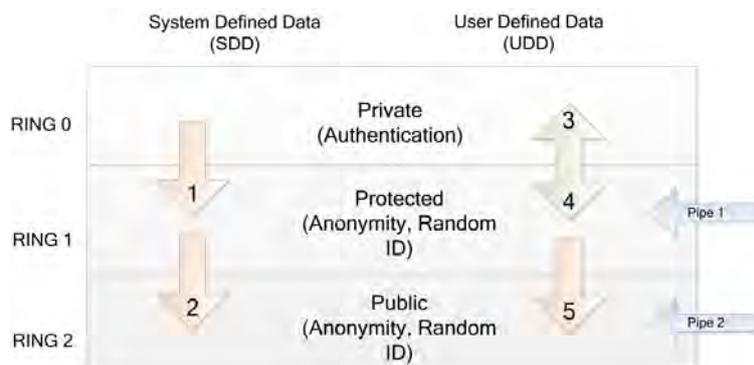


Fig.1. Structure of privacy system. The arrow means the accessible directions of the data flow. Privacy system is divided into two parts: the System Defined Data (SDD) area and the User Defined Data (UDD) Area.

Table 1. Rights Table of Accessing. The accessing types are classified according to SDD, UDD respectively.

Service Type	Public UDD	Public SDD	Protected UDD	Protected SDD	Private UDD	Private SDD
Service Private	Full	Full	Full	Full	Limited	No
Service Protected	Full	Full	Limited	Limited	Limited	No
Service Public	Full	Full	No	No	No	No

Public Service. After a piece of ubiquitous terminal enters the specific district of ubiquitous environment, the public service can contact directly with the terminal, read the whole information of public level without notifying the person who holds it, so there is no interaction. If the service matches with the preference of device and the policies of privacy system, the terminal notifies the user that a service called him. If one of the following conditions is satisfied:

1. The service is not satisfied the preference or policies;
2. The authorized service is trying to read the information of high level;
3. The existing preference is not enough to judge;
4. The terminal rejects all public service;

5. The terminal set the value that always rejects this kind of service.

The terminal rejects automatically without notifying.

Protected Service. It has the similar performance as the above. The protected service can not communicate directly with the terminal. Both of them must register in the environment. If a protected service needs to read the information in the protected level, the privacy system must notify the user. But this doesn't mean that every such kind of services need the permission to access, some of them just notify the user what they are doing. If the service meet the above five conditions listed in the public service part, the same operation will be made here.

Private Service. It is quite similar with the protected service. The main difference between them is the rights of accessing information of users.

As it can be seen from Fig.1, the whole system consists of three rows and two columns. Let us analyze them one by one in the following.

Row. PriL : Marked as "Ring 0 Level" in Fig.1. In this level, the identity includes name, ID number and so forth. The system provides no rights and direct access to external Service Providers (SP) for the protection of authentic ID information. The service that needs the information of this level must request from pipe 1. Depending on the willingness of entity, the system denies or accepts the requests.

ProL : Marked as "Ring 1 Level" in Fig.1. In this level, some aggregation of the profiles such as medical history is stored that are related to the privacy.

PubL : Marked as "Ring 2 Level" in Fig.1. In this level, the other types of the profiles are stored, mainly including some unimportant data that are generally irrelevant to the identity of a certain individual. That means, if the real identity is masked, no entity could utilize any mechanism to infer the real entity from the total profile it got.

Column. There are two reasons for system being divided into SDD and UDD: first, we strike the balance between security and personalization here, and the dotted line separates it as two independent storage space; second, since they are independent, the fails or threats in the User Defined Data Area have limited influence on the System Defined Data Area that is the core area of the system. In SDD Area, we can pre-define the following sets:

Private Data Entries (*PriDEs*) contains the crucial information such as name and ID, which are relevant to the identity of a certain person. Usually, only the Authentication System has the rights to read it for authentication in some certain circumstances. It is can be represented as follows.

$$PriDEs = \{name, IDnumber, Sex, \dots: \text{Items related to identity}\} \quad (1)$$

Protected Data Entries (*ProDEs*) are some data that are accessible in some circumstances by some objectives, but not accessible in other circumstances by some other objective. Take medical history for an example, not every hospital is trustworthy enough to read you medical history that exists in the Ubiquitous Computing Environments. It is a mathematical set:

$$ProDEs = \{medical\ history, \dots; \text{Items partially or partly related to privacy}\} \quad (2)$$

Public Data Entries (*PubDEs*) are some data that are related to hobbies, interests and so on. This information can be read by any service providers (SP). Since the random ID stream displaces the real identification immediately when it is in the range of available services, it is hard to trace the entity. We present it by the following set:

$$PubDEs = \{(Potato, Love), (Tomato, Dislike), \dots; \text{Items irrelevant to privacy}\} \quad (3)$$

3.2. Hierarchy of Services

Definition 2: Hierarchy of Services. In this paper, hierarchy of services refers to different service types that require different data type:

1. Privacy-based services, requiring private data and requesting the personal data via pipe 1, such as bank services, which need ID;
2. Protection-based services, requiring protected data and requesting the personal data via pipe 1, such as hospital services, which need Medical history;
3. Public-based services, requiring public data and requesting the personal data via pipe 2, such as weather services, which need temperature data.

Table 2. Basic needs of different services

Service Type	Security Agent	Authentication	Alarm
Service Private	Need	Need	Yes
Service Protected	Need	Need	When necessary
Service Public	No	No	No

Categorizing the services obviously facilitates the management of the services. Table 2 shows the basic needs for different sorts of services. In the ubiquitous computing environments, even in a district area, there could probably be many services. Since quite a lot of services just require the public information, it is feasible to make these services contact with entities directly without security agent and authentication towards public service. We also have

to clarify the worry about possible leakage of personal information. Tinghuai MA et al. [18] have utilized a spatiotemporally-based anonymity method and demonstrated that real identity can be masked. On one hand, it can light the system burden in the trusty area; on the other hand, it lowers the doorsill of service-providing and booms the passion of service provider to provide abundant services. The entity can avoid harassment from those services irrelevant or not interested by configuring the terminals using preference, combining the system's strategy (we will discuss a strategy named *N*th push strategy in the following), the entity can efficiently avoid the majority of unwelcome services and spam services. However, for the services needing protected information or private information, authentication and security agent are needed for entities to make sure the service he will receive is genuine and valid and the security agent authenticates the entities as well in the same time, confirming neither of two parties will jeopardize each other.

3.3. General Process

The three level services are donated as $Service_{PriL}$, $Service_{ProL}$ and $Service_{PubL}$ correspondently;

Thus we have the following process, the "have rights" means have rights to access information that is relevant to the respective service, or essential information that the service needs.

1. The Service Provider (SP) of $Service_{PubL}$ communicates with the entity directly through pipe 2, no intermediate Access Point (AP) needed, $Service_{PubL}$ are pushed to the entity without interaction, and no rights of accessing Ring 1 and Ring 0 are granted.

2. The Service Provider (SP) of $Service_{ProL}$ communicates with the entity through pipe 1. It has full rights to access Ring 2 via arrows 2 and 5, limited rights to access Ring 1, and strict rights to access Ring 0 via arrow 3, depending on the user's response, mandatory interaction needed.

3. The Service Provider (SP) of $Service_{PubL}$ communicates with the entity through pipe 1 too. It has full rights to access Ring 2 via arrow 2 and 5, full rights to access Ring 1 via arrow 1 and 4, and limited rights to access Ring 0, mandatory interaction needed.

4. Filter Policy

A satisfactory system consists of not only high-security architecture but also reliable policies to safeguard users' privacy. In this paper, we call the aggregation of those policies as Policy-making System.

4.1. Policy-making System

Here we introduce a system called policy-making system, its duty is to predicate the validation of present action via comparing with the former actions and determine the appropriate authorization and authenticating the entities when necessary.

The Fastest Velocity Policy (FVP). This policy can be expressed as follows: The system records location and time information of every access of an individual in Private Level and compares the corresponding information. Furthermore, if necessary, it calculates the fastest velocity and compare with the possible velocity. Let us see the exact mathematical table of certain service and interpret it firstly. Here we present policies of abnormality detecting and the details of Table 3.

Number. To record the sequence of the entry.

Access Time. It refers to the starting time and aborting time of a certain kind of service. Since it is the table of a certain kind of service, comparing with each starting time and aborting time makes some sense in some circumstances. With the increasing records, system will compute the time distribution to record or filter abnormal accessing requests, and raise the risk level to a certain degree.

Location. It refers to the contemporary location when one is requiring a kind of service, the location set is made up of three elements: x, y and z, representing location coordinates, respectively. We can easily calculate the distance between two places.

Velocity. The velocity can be calculated via the displacement and time interval between two neighboring entries. In practice, we have the possibly fastest velocity between the two neighboring place coordinates, when the calculated velocity value is beyond the possibly fastest velocity, the system can block this request because it must be an unauthentic or unauthorized request. The highest value of velocity can vary according to different areas and districts, and this causes different pre-set value.

Risk Level. The risk level is evaluated by the former three columns: Access Time, Location and Velocity. The initial value of risk level, of course, can be set as 0, that is, when the risk level is 0, the request is authentic and secure. When the security lever is below 0, it causes an alarm and the alarm level varies based on the degree.

Based on the former time distribution possibility and location distribution possibility, the present access time and access location can be detected and

located in the probability area in accordance with problematic judgment, namely, a present access time and location can be compared with the former access time and location habits, as shown in Table 3. Its time probability distribution chart can be described as Fig.2.

Table 3. Recording Table of a Certain Service

ID	Access Time	Location	Velocity	Risk Level	Remarks	Normal
1	[t1]	[l1]	N/A	N/A	start	N/A
2	[t2]	[l2]	N/A	N/A	abort	N/A
3	[t3]	[l3]	(l3-l2) / (t3-t2)	evaluation	start	evaluation
4	[t4]	[l4]	N/A	evaluation	abort	evaluation
5	[t5]	[l5]	(l5-l4) / (t5-t4)	evaluation	start	evaluation
...
...
...
2n-2	[t2n-2]	[l2n-2]		evaluation	start	evaluation
2n-1	[t2n-1]	[l2n-1]	(l2n-1-l2n-2) / (t2n-1-t2n-2)	evaluation	abort	evaluation
2n	[t2n]	[l2n]	N/A	evaluation	start	evaluation
2n+1	[t2n+1]	[l2n+1]	(l2n+1-l2n) / (t2n+1 - t2n)	evaluation	abort	evaluation
2n+2	[t2n+2]	[l2n+2]	N/A	evaluation	start	evaluation

According to the probability distribution, the risk level can be assessed in the following way: if a certain present action is in the most probability area, it marks itself as value 0, if in the second most probability area, it marks itself as value -1, and the rest can be done in the same manner. The lower the score is, the higher the risk is. The grade of probability area can vary in line with the practical needs.

Remarks. To remark the system state.

Normal. To assess whether the present request is valid and authentic via the former judgment and assessment, and present crucial information via a kind of human-readable way in this column.

In the same way, we can also construct a space table: we demarcate a certain area into small ones, record the login places vaguely or precisely, compute the probability, and assign each one an addition value. So we have the following mathematical expression to determine the risk level preliminarily.

$$Compare((V_{time} + V_{space} + \dots) | (V_{threshold} \{...\})) \tag{4}$$

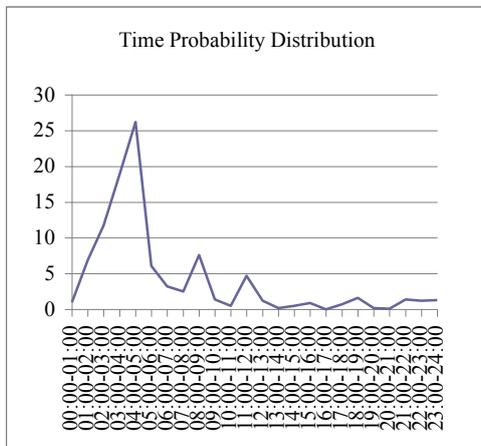


Fig. 2. Line chart of time probability distribution

If the value of former one is lower than the latter threshold value, it alarms. $V_{threshold} \{...\}$ can have multiple values, and we can set alarming level, respectively.

Algorithm of Comparison. Suppose an entity requires logging in when it is entering a certain area:

- Step 1: Record the present location and time information;
- Step 2: Compare the former recordings, evaluate the values and add them;
- Step 3: Compare the aggregation of the present value with the threshold;
- Step 4: ($Value_{Present} \leq threshold$) system denies the login request and informs the entity of authentication;
- Step 5: ($Value_{Present} \geq threshold$) system accepts the login request.

N-th Push Strategy. Nth Push Strategy is a Service-Orient Strategy, the purpose of which is to reduce the unnecessary interaction, which is also one of the purposes of Ubiquitous System. We assume there are many kinds of services in a certain area, based on the profile and preference of an individual who enters the area the services can be categorized into several sorts as follows:

1. The services that one is interested in;
2. The services that one is not interested in;
3. The services that one is uncertain or that one doesn't mention.

Since it is a strategy, it can vary itself in accordance with the relevant ambience and relevant entities. Here we just adopt a simple scenario that consists of two simple groups, and the strategy is defined as follows:

1. Define the services that one interests in (judged by use frequency and individual preference) or concerns to the data of private level or protected level have rights to push 2 times.

$$(Service_{interests} + Service_{PrL} + Service_{ProL} | Times_2) \quad (5)$$

2. Define the services that one does not interest in have rights to push 0 times.

$$(Service_{no-interests} | Times_0) \quad (6)$$

3. Define the services that one is uncertain have rights to push 1 times.

$$(Service_{uncertain} | Times_1) \quad (7)$$

4. If the services that one is uncertain are relevant to his existing preference, set an additional times, that is 1+1 times

$$(Service_{uncertain-relevant} | Times_{1+1}) \quad (8)$$

5. If the services that one is uncertain are irrelevant to his existing preference, set subtraction times, that is 0 times.

$$(Service_{uncertain-irrelevant} | Times_{1-1}) \quad (9)$$

6. If the services that one is uncertain and cannot be judged through his existing preference, remains.

The user preference is defined by him as follows:

1. Accept the interested N th times-based services, $N \geq 1$;
2. Filter the services not interested in to reduce the unnecessary interaction.

Suppose there is an entity who is entering the effective area, system generates a new random ID to identify the entity:

$$Entity \left(\begin{array}{l} Property \{ ID_{random}, \dots \} \\ Preference \{ \dots \} \end{array} \right) \quad (10)$$

And there are services categorized by level:

$$Service \left(\begin{array}{l} Service \{ Service_{private\ level} \} \\ Service \{ Service_{protected\ level} \} \\ Service \{ Service_{public\ level} \} \end{array} \right) \quad (11)$$

Or categorized by user's preference

$$Service \left(\begin{array}{l} Service\{Service_{interest}\} \\ Service\{Service_{not\ interest}\} \\ Service\{Service_{uncertain}\} \end{array} \right) \quad (12)$$

So the services that may be useful to the entity are $Service_{useful}$, M means Match.

$$Service_{useful} = Service(\dots) \quad (13)$$

$$M$$

$$Entity(Preference\{\dots\})$$

In this system, we separate the authentication and delivery of a certain service, though the integration may be more simple and convenient for us to build. The direct advantage is that even the Service Delivery fails because of the system’s heavy burden, or something else, it cannot influence the system’s authentication function. This can avoid one situation that we cannot even register.

4.2. Practical Analyses

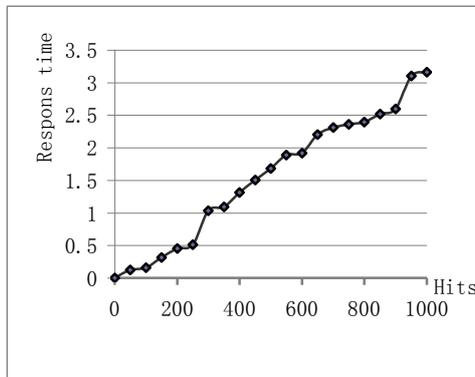


Fig. 3. Line chart of hit probability distribution

We conducted a simulation on our computers to prove our policy-making system’s efficiency. The computer simulates the hit probability trend when the times user used increase as shown in Fig.3.

The X axis represents the frequency in which service has been called. The Y axis represents the system’s response time of matching the suitable service. From the chart we can see that when a certain service had been used for initial 100 times, the policy-making system cannot work efficiently because of lacking of adequate data and distribution trend. The situation changed a lot when the

service had been used more than 100 times, the policy-making system takes effect. It demonstrates that this system is not suitable for contemporary services or services that one doesn't use so much.

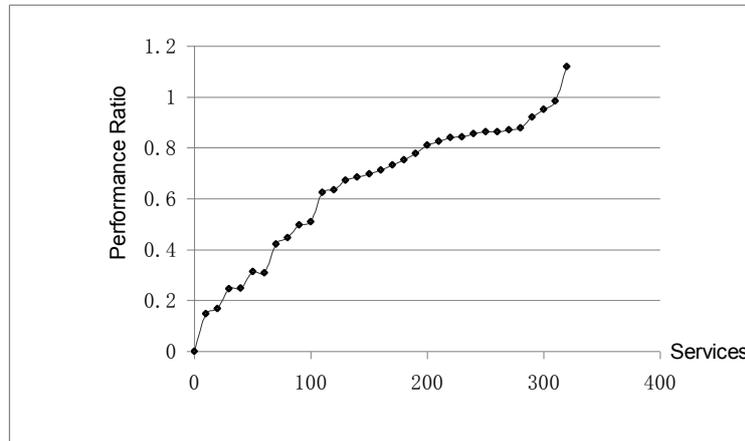


Fig. 4. The performance of system according the service number

We also examine the performance of policy-making system, which shown in Fig. 4.

We set the Performance Ratio $R = \tau_{actual} / \tau_{desire}$, if τ_{desire} is set to 1 second, the response time τ_{actual} should be below 1 second in order to get better performance. X axis is the number of available services.

According to our simulation, we can estimate that if there are about 320 services, the system works very well. However, if there are more than 320 services, the system is not able to respond in time. This is just a simulation, and later the mathematical calculation will be given to prove the simulation from another aspect.

5. System Overview

In this section, we integrate Policy-making System into Privacy System, and then give mathematical analyses. Finally we show a practical example to examine the whole system.

5.1. The Infrastructure

According to our previous design, the system is a critical part of infrastructure in a ubiquitous environment, which is in charge of access control of services,

and the policy-making system as a supplementary part and the discrimination of services. The whole infrastructure can be illustrated as Fig.5.

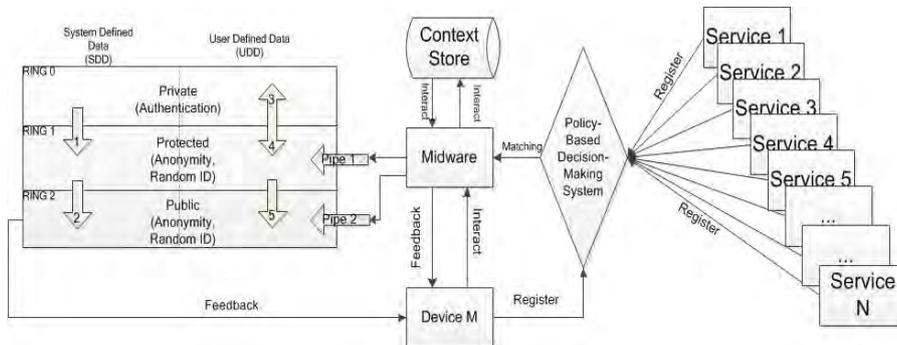


Fig. 5. Infrastructure of the privacy system

The infrastructure works as follows:

1. Available Services register in the Policy-Based Policy-Making System to inform the system of its existence and availability;
2. Device M enters the ubiquitous environment, registers anonymously in the Policy-Based Policy-Making System to inform the system of its existence and availability;
3. Policy-Based Policy-Making System synthesizes both service profile and device profile, and pushes the matching services to middleware;
4. Middleware check the Context Store and records the activities of Device M. Context Store records the history of Device M. If the current activities of Device M are abnormal, for example, alarm Device M to verify identity;
5. Middleware also separates the matching services, push them via Pipe 1 or Pipe 2, according to the information the services need;
6. Services are pushed to Device M.

5.2. XACML Description

XACML is adopted to express fine grained access data control, which describe the access process of Fig.1. Standard XACML includes three basic elements: <Rule>, <Policy> and <PolicySet>. <Rule> has two elements, <Condition> and <Target>, and an Effect attribute. The rule means: if the condition of the rule evaluates to be true, then the access control decision to perform <Actions> by the <Subjects> on the <Resources> are given by the Effect attribute.

Fig.6 shows an example XACML policy that specifies a permission to get data. Lines 3-9 define the policy’s target, which indicates that this policy is applicable to Private subject requesting permission to execute any action on UDD and SDD private data. Lines 11-37 indicate applicable requests to those

requesting accesses to the resource data with the action GetData. Lines 38-44 indicate that if the data does exist, the request should be permitted.

```

1.<Policy PolicyId="GetPrivateData"
2.  RuleCombiningAlgId="permit-overrides">
3.<Target>
4.    <Subjects ServiceType= Private></Subjects>
5.    <Resources >
6.    <UDD Type=Private><SDD Type=Private>
7.    </Resources>
8.    <Actions><AnyAction/></Actions>
9.  </Target>
10.<Rule RuleId=" GetPrivateData " Effect="Permit">
11.  <Target>
12.    <Subjects><AnySubject/></Subjects>
13.    <Resources>
14.      <Resource>
15.        <ResourceMatch MatchId="string-equal">
16.          <AttributeValue DataType="string">
17.            data
18.          </AttributeValue>
19.          <ResourceAttributeDesignator
20.            AttributeId="resource-id"
21.            DataType="string"/>
22.        </ResourceMatch>
23.      </Resource>
24.    </Resources>
25.    <Actions>
26.      <Action>
27.        <ActionMatch MatchId="string-equal">
28.          <AttributeValue DataType="string">
29.            GetData
30.          </AttributeValue>
31.          <ActionAttributeDesignator
32.            AttributeId="action-id"
33.            DataType="string"/>
34.        </ActionMatch>
35.      </Action>
36.    </Actions>
37.  </Target>
38.  <Condition FunctionId="Yes">
39.    <Apply FunctionId="data-exist">
40.      <ResourceAttributeDesignator
41.        AttributeId="new-data-id"
42.        DataType="data"/>
43.    </Apply>
44.  </Condition>

```

45. </Rule>
 46. </Policy>

Fig. 6. Get private data policy based on XACML

5.3. Cost of Evaluation

We assume C as cost of evaluation, m as the number of devices, n as the number of services, $p_{1,j}$ as failure probability of service from service 1 to service j , $p_{2,j}$ as the failure probability of system from service 2 to service j , i and j as two random services, $t(0 \leq t \leq 1)$ as the adjustment parameter set by system, then the following equation could be inferred:

$$C = \left[1 + \sum_{i=1}^{n-1} \prod_{j=1}^i p_{1,j} \right] + \prod_{i=1}^n p_{1,i} * \left[1 + \sum_{i=1, i \neq n-1}^{m-1} \prod_{j=1, j \neq n}^i p_{2,j} \right] * t \quad (14)$$

$$+ \left(1 - \prod_{i=1}^{n-1} p_{1,i} \right) * \left[1 + \sum_{i=1}^{m-1} \prod_{j=1}^i p_{2,j} \right] * t$$

The typical system can be inferred as:

$$C_{typical} = \left[1 + \sum_{i=1}^{m-1} \prod_{j=1}^i p_{2,j} \right] + \prod_{i=1}^n p_{2,i} * \left[1 + \sum_{i=1, i \neq n-1}^{n-1} \prod_{j=1, j \neq n}^i p_{1,j} \right] * t \quad (15)$$

$$+ \left(1 - \prod_{i=1}^{m-1} p_{2,i} \right) * \left[1 + \sum_{i=1}^{n-1} \prod_{j=1}^i p_{1,j} \right] * t$$

Since we want to compare the cost, we have:

$$F = C / C_{typical} \quad (16)$$

It is a large equation, but lucky enough if we just want to prove that our system is more efficient at a certain value range, we do not necessarily have to calculate all its values. We can just demonstrate $F \leq 1$.

We set $t = 1$ and omit the process of verification here. Based on accurate calculation, the threshold (M, N) should be (136, 314). That means, in this ubiquitous environment, under ideal conditions, there could be at most 136 devices and 314 services existed. If more, the system is not advantageous compared to a typical system existed before.

6. Conclusions & Future Work

We present a privacy-preserving architecture utilizing the classification of personal information and hierarchy of services, which are derived from the concept of Class in the Object Orient Programming. Based on such concept, personal information is manually or automatically categorized into Private (it can infer the real identity of an entity), Protected (it may infer the real identity of an entity) and Public (it cannot infer the real identity of an entity). All the three kinds of services have two independent storage spaces logically in order to make sure they would not be able to affect one another even some part is cheated by malicious users. Exposing the public information is somehow no harm to the entities, at the same time it could lower the doorsill of service-providing and ultimately booms the quantity of services. The categorized information of entity and categorized services are concerted, and the latter require information via respective pipes, one has lower security, suitable for services that need public information only without interactions, one has higher security, suitable for the rest two types. When it comes to the information that is not public, mandatory interactions are needed to alarm the entity and request the permissions to proceed. Besides, we combine some filter policies which is probability-based policy aiming to execute the preliminary judgment to judge whether the present register is suspicious or not, then choose appropriate actions, alarm, vibration, and require entity authentication in another way, including password, pre-designed questions or other security mechanisms. The other is called *N*th Push Strategy, aiming to filter the services that one does not need to combine with users' preferences, avoiding unwelcome services and unessential interactions, to meet the requirements of least interactions in the Ubiquitous Computing.

Our system has the following significant advantages compared to other similar system:

1. Higher efficiency. Our system is highly efficient, especially when it comes to ubiquitous terminals, most of which are battery-powered with slower CPUs. Through executing a simple set of rules, system can block most of services that are irrelevant to users as well as spam services, although not all of them.

2. Highly customization. Our system allows users to specify their preferences in a very simple manner, matching and discarding most services without keeping alarming the users what it is doing.

3. Security. The system adopts the hierarchy and classification to ensure that compromise of a single part will not jeopardize the whole system, especially the SDD area.

4. Compatibility. Ubiquitous environment is a device-rich environment, which means compatibility is a major issue to a variety of different devices. Basically speaking, any device that is applied to this system can be compatible with one another.

5. Adaptability. The system allows the same task to be performed differently in different environments. The only thing that one should do is to achieve this and to adjust the system configurations in various environments.

Though our system can work appropriately and satisfactorily, there still exist many problems compared with other existing systems. Basically speaking, it is a behavior-based system. When it comes to a service that a user has used many times, with the former behaviors profile, the system can discriminate and adjudge the present behaviors appropriately. However, when it comes to a new service, the policy-making part of whole system is nearly useless. So one emergent problem we must solve is that we should accomplish how to make this part of system work appropriately and satisfactorily even when it confronts a new service. The other work we must solve is that we should optimize the algorithms to fit with the needs of the large scale computing and accurate computing.

Acknowledgement. This work was supported in part by the National Science Foundation (61173143), and also supported by A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions. This research was also supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency)"(NIPA-2009-(C1090-0902-0002)).

References

1. Weiser, M.: The future of ubiquitous computing on campus. *Communications of the ACM*, Vol. 41, No. 1, 41-42. (1998)
2. Tent ori, M., Favela, J., Gonzalez, V.: Quality of Privacy (QoP) for the Design of Ubiquitous Healthcare Applications. *Journal of Universal Computer Science*, Vol. 12, No. 3, 252-269. (2006)
3. Hong, J. I., Ng, J. D., Lederer, S., Landay, J. A.: Privacy Risk Models for Designing Privacy-Sensitive Ubiquitous Computing Systems. In *Proceedings of ACM conference on Designing Interactive Systems (DIS2004)*. ACM, Cambridge, Massachusetts, 91-100. (2004)
4. Sweeney, L.: k-Anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, Vol. 10, No. 5, 557-570. (2002)
5. Cheng, H. S., Zhang, D., Tan, J. G.: Protection of privacy in pervasive computing environments. In *Proceedings of International Conference on Information Technology: Coding and Computing, (ITCC 2005)*. IEEE Computer Society, Washington, USA, Vol. 2, 242-247. (2005)
6. Papadopoulou, E., McBurney, S., Taylor, N., Williams, M. H., Dolinar, K., Neubauer, M.: Using User Preferences to Enhance Privacy in Pervasive Systems. In *Proceedings of Third International Conference on Systems (ICONS 08)*. IEEE Computer Society, Cancun, Mexico, 271-276. (2008)
7. Diep, N. N., Lee, S. Y., Lee, Y. K., Lee, H. J.: A Privacy Preserving Access Control Scheme using Anonymous Identification for Ubiquitous Environments. In *Proceedings of the 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*. IEEE Computer Society, Daegu, Korea, 482-487. (2007)
8. He, Q., Wu, D., Khosla, P.: The quest for personal control over mobile location privacy. *Communications Magazine, IEEE*, Vol.42, No.5, 130-136. (2004)

9. Langheinrich, M.: A Privacy Awareness System for Ubiquitous Computing Environments. In Proceedings of the 4th International Conference on Ubiquitous Computing(UbiCom2002). Springer-Verlag, Goteborg, Sweden, 237-245. (2002)
10. Myles, G., Friday, A., Davies, N.: Preserving privacy in environments with location-based applications. *Pervasive Computing, IEEE*, Vol.2, No.1, 56-64. (2003)
11. Zheng, Y., Chiu, D.; Wang, H., Hung, P.: Towards a Privacy Policy Enforcement Middleware with Location Intelligence. In Proceedings of 11th IEEE International Enterprise Distributed Object Computing Conference. IEEE Computer Society, Maryland, USA, 97-104. (2007)
12. Pallapa G., Roy, N., Das, S. K.: A scheme for quantizing privacy in context-aware ubiquitous computing. In Proceedings of IET 4th International Conference on Intelligent Environments. IET, Seattle, USA, 1-8. (2008)
13. Yee, G.: Using privacy policies to protect privacy in UBICOMP. In Proceedings of 19th International Conference on Advanced Information Networking and Applications. IEEE Computer Society, Taiwan, Vol.2, 633-638. (2005)
14. Kang, Y., Lee H., Chun, K., Song, J.: Classification of Privacy Enhancing Technologies on Life-cycle of Information. In Proceedings of The International Conference on Emerging Security Information, Systems, and Technologies. IEEE Computer Society, Valencia, Spain, 66-70. (2007)
15. Lee, B., Kim, H.: Privacy Management for Medical Service Application Using Mobile Phone Collaborated with RFID Reader. In Proceedings of Third International IEEE Conference on Signal-Image Technologies and Internet-Based System. IEEE Computer Society, Shanghai, China, 1053-1057. (2007)
16. Martimiano, L. A. F., Goncalves, M. R. P., dos Santos Moreira, E.: An ontology for privacy policy management in ubiquitous environments. In Proceedings of IEEE Network Operations and Management Symposium. IEEE Computer Society, Bahia, Brazil, 947-950. (2008)
17. Ma, T. H., Kim, S. D., Wang, J., Zhao, Y. W.: Privacy Preserving in Ubiquitous Computing: Challenges & Issues. In Proceedings of IEEE International Conference on e-Business Engineering. IEEE Computer Society, Xi'an, China, 297-301. (2008)
18. Ma, T. H., Yang, S., Tian, W., Liu, W. J.: Privacy Preserving In Ubiquitous Computing: Architecture. *Information Technology Journal*, Vol. 8, No.6, 910-916. (2009)
19. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkatasubramaniam, M.: I-Diversity: Privacy Beyond k-Anonymity. In Proceedings of 22nd IEEE International Conference on Data Engineering. IEEE Computer Society, Atlanta, USA. (2006)
20. Kaya, S. V., Savaş, E., Levi, A., Erçetin, Ö.: Privacy-Aware Multi-Context RFID Infrastructure Using Public Key Cryptography. In: Ian F. Akyildiz, Raghupathy Sivakumar, etc (Eds.): *Networking 2007. Lecture Notes in Computer Science*, Vol. 4479. Springer-Verlag, Berlin Heidelberg New York, 263-274. (2007)
21. Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., Ribagorda, A.: RFID Systems: A Survey on Security Threats and Proposed Solutions. In Proceedings of the 11th IFIP International Conference on Personal Wireless Communications (PWC'06). Springer-Verlag, Albacete, Spain, 159-170. (2006)
22. KIM, I., LEE, B., KIM, H.: Privacy Protection Based on User-defined Preferences in RFID System. In Proceedings of International Conference on Advanced Communication Technology-ICACT'06. IEEE, Phoenix Park, Korea, 858-862. (2006)
23. ROBINSON, P., BEIGL, M.: Trust Context Spaces: An Infrastructure for Pervasive Security. In Proceedings of First International Conference on Security in Pervasive Computing. Springer-Verlag, Boppard, Germany, 157-172. (2003)

24. ORTMANN, S., Langendörfer, P., MAASER, M.: A Self-Configuring Privacy Management Architecture for Pervasive Systems. In Proceedings of 5-th ACM International Workshop on Mobility Management and Wireless Access (MobiWAC). ACM New York, Chania, Crete Island, Greece, 184-187. (2007)
25. Moncrieff, S., Venkatesh, S., West, G.: A Framework for the Design of Privacy Preserving Pervasive Healthcare. In Proceedings of the IEEE International Conference on Multimedia and Expo. IEEE, Cancun, Mexico, 1696-1699. (2009)
26. Ning, S., Nabeel, M., Paci, F., Bertino, E.: A Privacy-Preserving Approach to Policy-Based Content Dissemination. In Proceedings of the IEEE 26th International Conference on Data Engineering. IEEE, Long Beach, California, USA, 944-955. (2010)
27. Xu, M., Wijesekera, D., Zhang, X.: Runtime Administration of RBAC Profile for XACML. IEEE Transactions on Services Computing, 1-1, Vol.3, No. 1. (2010)
28. Ardagna, C. A., De Capitani di Vimercati, S., Neven, G., Paraboschi, S., Preiss, F.-S., Samarati, P., Verdichio, M.: Enabling Privacy-Preserving Credential-Based Access Control with XACML and SAML. In Proceedings of the IEEE 10th International Conference on Computer and Information Technology (CIT). IEEE, Bradford, UK, 1090-1095. (2010)
29. Kodeswaran, P., Viegas, E.: Towards A Privacy Preserving Policy Based Infrastructure for Social Data Access To Enable Scientific Research. In Proceedings of the 8th Annual International Conference on Privacy Security and Trust (PST). IEEE Computer Society, Ottawa, Ontario, Canada, 103-109. (2010)
30. Arunkumar, S., Rajarajan, M.: Healthcare Data Access Control using XACML for Handheld Devices. In Proceedings of the Developments in E-systems Engineering (DESE). IEEE, London, UK, 35-38. (2010)
31. Dai, C. Y., Gong, W. T., Liu, J.: The Research of Access Process in Web Services Based on XACML. In Proceedings of the 2nd International Workshop on Database Technology and Applications (DBTA). IEEE, Wuhan, China, 1-4. (2010)
32. Malik, A. K., Dustdar, S.: A Hybrid Sharing Control Model for Context Sharing and Privacy in Collaborative Systems. In Proceedings of the IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA). IEEE, Biopolis, Singapore, 879-884. (2011)
33. Oyomno, W., Jäppinen, P., Kerttula, E.: Privacy Preservation for Personalised Services in Smart Spaces. In Proceedings of the Baltic Congress on Future Internet Communications (BCFIC Riga). IEEE, Latvia, Riga, 181-189. (2011)

Tinghuai Ma is an associate professor in Computer Sciences at Nanjing University of Information Science & Technology, China. He received his Bachelor (HUST, China, 1997), Master (HUST, China, 2000), PhD (Chinese Academy of Science, 2003) and was Post-doctoral associate (AJOU University, 2004). From Nov.2007 to Jul. 2008, he visited Chinese Meteorology Administration. From Feb.2009 to Aug. 2009, he was a visiting professor in Ubiquitous computing Lab, Kyung Hee University. His research interests are in the areas of Data Mining and Privacy Protected in Ubiquitous System, Grid Computing. His research interests are data mining, grid computing, ubiquitous computing, privacy preserving etc. He has published more than 70 journal/conference papers. he is principle investigator of several NSF projects. He is a member of IEEE.

Tinghuai Ma, Sen Yan, Jin Wang, and Sungyoung Lee

Sen Yang is a graduate at School of Informatics and Computing, Indiana University – Bloomington. He received his Bachelor degree in NUIST in 2009. His research interests are in the related areas of Human-Centered Computing, including Human-Centered Interaction / design, Ubiquitous / Pervasive Computing and Data Mining.

Jin Wang received the B.S. and M.S. degree in the Electronical Engineering from Nanjing University of Posts and Telecommunications, China in 2002 and 2005, respectively. He received Ph.D. degree in Ubiquitous Computing laboratory in the Computer Engineering Department of Kyung Hee University Korea. Now, he is a professor in Nanjing university of Information Science & technology. His research interests include routing protocol and algorithm design, analysis and optimization, and performance evaluation for wireless ad hoc and sensor networks.

Sungyoung Lee received his B.S. from Korea University, Seoul, Korea. He got his M.S. and Ph.D. degrees in Computer Science from Illinois Institute of Technology (IIT), Chicago, Illinois, USA in 1987 and 1991 respectively. He has been a professor in the Department of Computer Engineering, Kyung Hee University, Korea since 1993. He is a founding director of the Ubiquitous Computing Laboratory, and has been a.iliated with a director of Neo Medical ubiquitous-Life Care Information Technology Research Center, Kyung Hee University since 2006. Before joining Kyung Hee University, he was an assistant professor in the Department of Computer Science, Governors State University, Illinois, USA from 1992 to 1993. His current research focuses on Ubiquitous Computing and applications, Context-aware Middleware, Sensor Operating Systems, Real-Time

Received: March 15, 2011; Accepted: May 2, 2011.

TRM-IoT: A Trust Management Model Based on Fuzzy Reputation for Internet of Things

Dong Chen¹, Guiran Chang², Dawei Sun¹, Jiajia Li¹, Jie Jia¹, and
Xingwei Wang¹

¹ College of Information Science and Engineering, Northeastern University,
110004 Shenyang, China
chend.2008@gmail.com

² Computing Center, Northeastern University,
110004 Shenyang, China
chang@neu.edu.cn

Abstract. Since a large scale Wireless Sensor Network (WSN) is to be completely integrated into Internet as a core part of Internet of Things (IoT) or Cyber Physical System (CPS), it is necessary to consider various security challenges that come with IoT/CPS, such as the detection of malicious attacks. Sensors or sensor embedded things may establish direct communication between each other using 6LoWPAN protocol. A trust and reputation model is recognized as an important approach to defend a large distributed sensor networks in IoT/CPS against malicious node attacks, since trust establishment mechanisms can stimulate collaboration among distributed computing and communication entities, facilitate the detection of untrustworthy entities, and assist decision-making process of various protocols. In this paper, based on in-depth understanding of trust establishment process and quantitative comparison among trust establishment methods, we present a trust and reputation model TRM-IoT to enforce the cooperation between things in a network of IoT/CPS based on their behaviors. The accuracy, robustness and lightness of the proposed model is validated through a wide set of simulations.

Keywords: Internet of Things, Cyber Physical System, Wireless Sensor Network, Trust, Reputation, Fuzzy Sets.

1. Introduction

Cyber-Physical Systems (CPS) are systems deployed in large geographical areas and generally consist of a massive number of distributed computing devices tightly coupled with their physical environment [1]. CPS and Internet of Things (IoT) have always been closely related, since both of them employ physical objects and events, including WSNs, RFID-based systems, mobile phones, etc. Cyber-Physical Internet [1], which can roughly be viewed as a

large-scale universal network that interconnects several heterogeneous CPS in order to ensure worldwide interoperability of cyber-physical devices. Therefore, we argue that the proposed fuzzy theory based trust and reputation model is not only suitable for CPS, but also suitable for IoT.

IoT and CPS cannot perceive physical information from physical world themselves. Intelligent things are usually labeled with RFID tags or equipped with sensors and sensors are widely regarded as the nerve endings of IoT/CPS [2] [3]. Sensors or sensor embedded things can usually form a wireless multi-hoc network-WSN employing ZigBee, Wi-Fi, Bluetooth and etc. In a future IoT/CPS, a large number of embedded, possibly mobile computing devices will be interconnected through WSNs, constituting various autonomous subsystems that provide intelligent services for end users. IoT/CPS can benefit from WSNs from the perspective that so far, sensors and RFID readers are the most efficient tools to obtain sensed data from the physical world, turning ubiquitous computing of IoT/CPS into a reality. Therefore, Internet connectivity in WSNs of the IoT/CPS is highly desirable, featuring sensing services at a global scale all over the world [4].

However, such networks present some new challenges when compared with traditional computer networks, namely in terms of smart node hardware constraints, very limited computing and energy resources. Unlike other networks using dedicated nodes to support basic functions like packet forwarding, routing, and network management, in WSNs of IoT/CPS, those functions are carried out by all available nodes. This significant difference is at the core of the increased sensitivity to node misbehavior. Due to the wireless nature of this kind of WSNs, it is also quite possible that a node could be captured by an adversary, which may lead to its non-cooperative behavior or misbehavior with the rest of the nodes in the network and even become a malicious node. Malicious nodes aim at damaging other nodes by causing network outage by partitioning.

In order to facilitate the detection of untrustworthy nodes, and assist decision-making process of various protocols in a WSN which is vital for carrying out specific tasks as it aids sensors establish collaborations, it is necessary to provide a trust and reputation mechanism for WSNs of IoT/CPS. One strategy to improve the security of WSNs is to develop trust mechanisms that allow a node to evaluate trustworthiness of other nodes [5] [6]. Such trust and reputation systems not only help in node behavior detection, but also improve network performance since honest nodes can avoid working with untrustworthy nodes [7].

The measurement and computation of trust and reputation to secure interactions between sensor nodes in IoT/CPS is crucial for the development of trust and reputation management mechanisms. The calculation and measurement of trust and reputation in a supervised ad-hoc environment involves complex aspects such as credible rating for opinions delivered by a node, the honesty of recommendations provided by a sensor node, or the assessment of past experiences with the node one wishes to interact with. The deployment of suitable algorithms and models imitating fuzzy logic can help to solve these problems. Therefore, the focus of this paper is to develop

a fuzzy theory based trust and reputation model for IoT/CPS environment. The proposed theoretical models are then applied to improve the performance of routing algorithms and detect node behaviors of WSNs in IoT/CPS.

The contribution of this paper can be categorized as follows. (1) Analysis of special features and unique trust challenges of IoT/CPS; (2) Concepts of trust and reputation and discussion of the relationship between trust and reputation in IoT/CPS; (3) A novel fuzzy theory based trust and reputation management model towards IoT/CPS; (4) Trust evaluation metrics, local trust relationship evaluation and global trust relationship evaluation; (5) A wide set of simulations, performance evaluations of the proposed fuzzy trust and reputation management model.

The remainder of the paper is organized as follows. We give an overview of related influential works in Section 2. In Section 3, a novel trust and reputation model for choosing trusted source nodes base on the fuzzy relationship theory in fuzzy mathematics TRM-IoT is proposed and further discussed in detail in WSNs of IoT/CPS. The simulation results in Section 4 show that TRM-IoT model can effectively prevent malicious and selfish nodes. TRM-IoT scheme can promote data forwarding and cooperation between nodes and improve the performance of the entire network, followed by the conclusion and future work of the paper in Section 5.

2. Related Works

Establishing security communication channels based on trust and reputation models among sensor nodes is an important consideration when designing a secure routing solution in IoT/CPS.

ATRM [8] is an agent-based trust and reputation management scheme for WSNs where trust and reputation management is carried out locally with minimal overhead in terms of extra messages and time delay. However, since mobile agents are designed to travel over the entire network and run on remote nodes, they must be launched by trusted entities. An agent-based trust model for WSN is presented in [9] using a watchdog scheme to observe the behavior of nodes and broadcast their trust ratings. Sensor nodes receive the trust ratings from the agent nodes, which are responsible for monitoring the former and computing and broadcasting those trust ratings. In [10], a reputation-based scheme called DRBTS is proposed to provide a method by which beacon nodes, *BN*, can monitor each other and provide information so that sensor nodes, *SN*, can choose who to trust, and based on a quorum voting approach. However, in order to trust a *BN*'s information, a sensor must get votes for its trustworthiness from at least half of their common neighbors. BTRM-WSN [11] is a bio-inspired trust and reputation model for WSN aimed to achieve to the most trustworthy path leading to the most reputable node in a WSN offering a certain service. Each node must maintain a pheromone trace for each of its neighbors. CONFIDANT [12] is proposed to extend

reactive routing protocols with a reputation-based system in order to isolate misbehaving nodes. Each node monitors the behaviors of its next hop neighbors. Trust relationships and routing decisions are based on experienced, observed, or reported routing and forwarding behavior of other nodes. SORI [13] scheme is proposed to encourage packet forwarding and discipline selfish behavior. The reputation of a node is quantified by objective measures, and the propagation of reputation is efficiently secured by a one-way-hash-chain-based authentication scheme. Watchdog and Pathrater mechanisms [14], are just two extensions to the DSR algorithm.

However, not all of the most known works take into account the strong restrictions about processing, storage or communication capabilities. Some of them rely on a watchdog mechanism with or without using a multi-agent system. IoT/CPS assumes that trillions of things which are used on a daily basis will eventually be connected to the Internet employing 6LoWPAN [15] protocol and provide intelligent service through cooperating with each other. Most things have the following significant characteristics [16] [17], limited power capability, wireless receivers and transmitters with limited range facing the use of multi-hop communication, mobility (things will move, possibly become disconnected) and violability (things may be switched on and off frequently). All the above issues raise the need for the development of a novel management model, different from those being in use today. Based on the research of characteristics of IoT/CPS and in-depth understanding of ATRM [8], ATSN [9], DRBTS [10], BTRM-WSN [11], CONFIDANT [12], SORI [13] and WP [14], we propose a novel trust and reputation model TRM-IoT to enforce the cooperation between things in a network of IoT/CPS based on their behaviors.

3. TRM-IoT: A Trust Model for IoT/CPS

The trust between sensor nodes cannot be set up simply by using the traditional trust mechanisms. In a human social community, trust between two individuals is developed based on the reputation evaluation of their actions over time. When faced with uncertainty, individuals will trust and rely on the actions and evaluations of others who have behaved well in the past.

Trust is one of the most fuzzy, dynamic and complex concepts in both social and business relationships. The difficulty in measuring trust and predicting trustworthiness in service-oriented network environments leads to many problems. These include issues such as how to measure the willingness and capability of individuals in the trust dynamics and how to assign a concrete level of trust to an individual. Wireless networks of IoT/CPS have several salient characteristics, such as dynamic topologies, bandwidth constraints, variable capacity links, energy constrained operation, and limited physical security. Due to these features, WSNs of IoT/CPS are particularly vulnerable to all kinds of attacks launched through malicious nodes. Unreliable wireless links are vulnerable to jamming and

eavesdropping. Constraints in bandwidth, computing power, and battery power in mobile devices may lead to their trade-offs between security and resource consumption. Dynamics make it hard to evaluate node behaviors, because routes in this kind of networks change frequently. Sensors or sensor embedded things are more likely to form a wireless multi-hoc network. Therefore, they cannot rely on central authorities and infrastructures for key management.

In this paper, we propose a generalized and unified mechanism to address the trust and reputation issue by developing a community of sensor nodes in the WSNs of IoT/CPS. Our motivation is to develop a similar behavior and fuzzy theory-based trust and reputation model for sensor nodes or sensor-embedded nodes, where each node develops a direct reputation for each other node by making direct observations and indirect reputation between individuals set up upon recommendations of other individuals about these other nodes in the neighborhood. The two kinds of reputations are used together to help a node evaluate the trustworthiness of other sensor nodes, detect the malicious nodes, and assist decision-making within the wireless network. The proposed scheme can be employed in any WSNs routing protocol to enforce cooperation among nodes and counter with non-cooperative nodes in IoT/CPS infrastructure.

The resource constraints of WSNs such as limited battery lifetime, memory space and computing capability in IoT/CPS make it easy to attack and fairly hard to protect. Therefore, it is fairly critical to detect the compromised nodes in order to avoid being misled by those compromised or malicious nodes. However, malicious nodes are so difficult to detect even a cryptography mechanism is applied, since most low-cost tiny sensor nodes are not tamper-resistant and easy to be cracked by the adversary. Therefore, in this paper we argue that behaviors-based trust and reputation mechanism can be used to resolve this problem efficiently. Based on this motivation, this paper proposes a novel behavior-based trust and reputation for IoT/CPS. The management model of trust and reputation is related to the creation, update and deletion of trust and reputation degree.

First, an effective lightweight authentication mechanism must exist to ensure all the identities are trustworthy [18] [19] [20]. That means the identity of each sensor node is unique and trustworthy, on the basis of cryptographic primitives [21] [22] [23]. In fact, we have proposed a novel lightweight pairwise key management scheme towards IoT/CPS in a previous paper before this one. Second, the task evaluation component evaluates the performance of the nodes, including sensors nodes and sensor-embedded device nodes. The tasks here include data processing and routing. Third, evaluation combination component is in charge of the result combination of the old trust degree and the indirect information from the third node in order to form the new trust degree which is used in future task allocation and evaluation.

Throughout this paper we assume a scenario where a WSN of IoT/CPS is composed of hundreds of sensor nodes with relatively high sensor activity. Without loss of generality, we consider some sensor nodes requesting lightweight common services and some nodes providing these services. We

also assume that every sensor node in the WSN only knows its neighbors and nothing else about the whole topology of the WSN. Additionally, the topology is considered to be relatively highly dynamic, with many nodes joining or leaving the community. The contribution of the proposed model is aimed to help a sensor node requesting a specific service to find the most trustworthy route leading to another sensor node providing the corresponding service. An untrustworthy node in this paper can be considered either because it intentionally provides a fraudulent service or because it provides a wrong service due to hardware failures or performance deterioration.

In this paper, taking costly decisions depends on the expectations created according to past behavior of others. Usually, this kind of information is called reputation and it is one of the most significant factors to trust merchants and recommenders towards IoT/CPS.

3.1. Definitions of Trust and Reputation

Although we experience and rely on trust in everyday life, it is so challenging to define trust accurately. The literature on trust is also quite confusing, since it manifests itself in fairly different forms. In this paper, we adopt the following definitions for trust and reputation.

Definition 1. Trust is the subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends [24].

Definition 2. Reputation is what is generally said or believed about a thing's character or standing [25].

Reputation exists only in a community which is observing its members in one way or the other. Accordingly, reputation is the collected and processed information about one partner's former behavior as experienced by others.

Josanga [25] gives a survey of trust and reputation systems and points out that there is significant difference between trust and reputation. Trust is a subjective phenomenon which is based on various factors or evidences. In fact, firsthand experience always carries more weight than the secondhand trust recommendation or reputation. Since the nodes in the data collection layer of IoT/CPS usually are heterogeneous and mobile, trust establishment model can significantly stimulate collaboration among distributed computing and communication entities, facilitate the detection of untrustworthy entities, and assist decision-making process of various protocols.

Based on [24] and [25], we try to give the following more detailed trust and reputation definitions towards IoT/CPS.

Definition 3. In a wireless network of IoT/CPS, a node *S*'s trust in another node *P* is the subjective expectation of node *S* receiving positive outcomes through the transactions with node *P*.

Definition 4. In IoT/CPS, a node *S*'s reputation is the global perception of its trustworthiness in the wireless network. Furthermore, the trustworthiness can be evaluated from its past and current behaviors.

Trust in this paper describes the relying node's trust in a service or resource provider node and it is relevant when the relying party is a user seeking protection from malicious or unreliable service providers.

3.2. Relationship between Trust and Reputation

The term 'trust' and 'reputation' have strongly linked meanings. Especially in WSNs of IoT/CPS, trust is often defined as an abstract acquired attribute relative to some sensor nodes which is due to the amount of reputation held by such sensor nodes.

By making full use of observing good long-term behavior, reputation ratings can be improved; therefore, trust relationships will be easily established [5]. In real-life communities, trust is the consequence of the satisfaction of certain desired properties [26].

As discussed in [25], the concept of reputation is closely linked to that of trust; however, there is a clear and significant difference. A node *S* can trust in another node *P* because of its good reputation. Likewise, node *S* can also trust in node *P* in spite of its bad reputation. Reputation is usually inspired by the past behaviors observed. Trust reflects the relying party's subjective view of an entity's trustworthiness, whereas reputation is a score which can be seen by the whole community.

Note that, in this paper, trust is considered as a subjective probability value while reputation is regarded as an objective and acknowledged value in a specific community context.

3.3. Fuzzy Trust Model Description

An entity's trustworthiness is the quality indicator of the entity's services, which is used to predict the future behavior of the entity (stored in sensors or sensor-embedded things). Intuitively, if it is trustworthy enough, the entity will provide good services for future transactions. In most trust models, the domain of trustworthiness is assumed to be $[0, 1]$.

Since the key issue in investigating fuzzy problems is to establish membership functions (membership degrees) by employing the fuzzy set theory, we have to create the mathematical model of fuzzy trust firstly [27].

Suppose that $SN = \{SN_1, SN_2, \dots, SN_n\}$ is a problem domain of the fuzzy trust model. Note that, $SN_i (i = 1, 2, \dots, n)$ is a subset in the corresponding domain. Then we can get the following mapping,

$$\begin{cases} MappingFuction : SN \times SN \rightarrow [0, 1], \\ (SN_i, SN_j) \rightarrow \psi(SN_i, SN_j) \in [0, 1]. \end{cases} \quad (1)$$

where $\psi(SN_i, SN_j)$ represents the degree of trust relationship between SN_i and SN_j . *MappingFuction* is a fuzzy relation mapping from $SN \times SN$ to $[0, 1]$.

In the proposed scheme, a neighbor monitoring process is used to collect information of the package forwarding behaviors of the neighbors. Each sensor node in the network maintains a data forwarding transaction table as follows,

$$DFT_{i,j} = \langle Source, Destination, RF_{i,j}, F_{i,j}, TTL \rangle \quad (2)$$

where *Source* is the trust and evaluation evaluating nodes, *Destination* is the evaluated destination nodes, $RF_{i,j}$ denotes the times of successful transactions which node SN_i has made with node SN_j , and $F_{i,j}$ denotes the positive transactions.

3.4. Trust Evaluation Metrics

Within the realm of IoT/CPS security, we interpret the concept of trust as a relation between entities stored in sensor nodes that participate in various protocols. Trust relations are based on evidence or reputation created by the previous interactions of entities within a protocol. Each node employs a neighbor monitoring process in order to collect information about the packet forwarding behaviors of the neighbor nodes. Furthermore, each node is capable of overhearing the transmissions of its neighbors in the promiscuous mode. Each node independently overhears its neighboring nodes packet forwarding activities. This overhearing is related to the proportion of correctly forwarded packets with respect to the total number of packets to be forwarded during a fixed time window. Then, each node in the network maintains a data forwarding information table. The table includes only the data forwarding transaction information by overhearing its neighboring nodes.

In the proposed model, we consider the following trust evaluation metrics for the establishment and validation of the proposed trust management model.

(1) End-to-end packet forwarding ratio (EPFR). EPFR is defined as the ratio between the numbers of packets received by the application layer of destination nodes to the numbers of packets sent by the application layer of the source node. The EPFR can be calculated by

$$EPFR = \frac{\sum_i^k RECV_i}{\sum_i^n SEND_i}, 0 \leq k \leq n. \quad (3)$$

where $RECV_i$ and $SEND_i$ denote the packages received and sent by the i -th destination node and the i -th source node, respectively. And k denotes the successful receiving times, while n denotes the total times of packages sending.

(2) AEC. The key criterion for the design of a WSN in IoT/CPS Infrastructure is the energy consumption. In order to research and analyze the energy consumption of our TRM-IoT model, we define the energy consumption metric as follows,

$$AEC = \frac{\sum_{i=1}^n consume_i}{\sum_{i=1}^n send_i + recv_i + \tau} \quad (4)$$

where $send_i$ and $recv_i$ denotes the energy consumption when the i -th sensor node sending and receiving messages, respectively. $consume_i$ denotes the total energy cost of consumption the trust and reputation values of the corresponding sensor node. And τ represents the other energy consuming which is used to maintain the normal running of the node itself.

(3) PDR. In fact, the package delivery ratio (PDR) is affected by the packet loss and packet retransmissions. Packet loss may occur for many reasons. In this paper we only focus on the behavior that an intermediate node intentionally drops the received data packets instead of forwarding them to the next hop node.

3.5. Reputation Evaluation

Node SN_i evaluates the reputation of node SN_j with which it tries to make transactions by rating each package forwarding process as either positive or negative, depending on whether SN_j has completely done the transaction correctly.

As discussed above, we use Con to describe the evaluation of the whole metrics in order to judge whether this transaction is successful. The Con can be computed by

$$Con = [EPFR, AEC, PDR] \cdot \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \alpha \cdot EPFR + \beta \cdot AEC + \gamma \cdot PDR \quad (5)$$

where α, β, γ represent the corresponding aspect weights of the different resources. We also define a parameter $Sat_{Threshold}$ to describe the satisfaction degree. That means, if $Con < Sat_{Threshold}$, then it indicates that node SN_i get a negative reputation evaluation to node SN_j ; if $Con \geq Sat_{Threshold}$, it indicates that node SN_i gets a positive reputation evaluation to node SN_j .

The reputation evaluation of all interactions from node SN_i to node SN_j is defined as follows,

$$\delta = \frac{F_{i,j}}{RF_{i,j}} \in [0,1]. \quad (6)$$

Reputation evaluation is the basis of trust management. In our trust model, the reputation is evaluated considering three metrics, EPFR, AEC and PDR. Compared with other reputation evaluation methods, we consider more factors which can more accurately evaluate the behaviors of nodes according to specific characteristics of IoT/CPS.

3.6. Local Trust Evaluation

From different points of view, trust can usually be categorized into different classes: direct trust and indirect trust. When we say node SN_j is trustworthy or untrustworthy for the node SN_i , it means that there must be a trust and reputation model between node SN_j and node SN_i . If a trust relationship statement is based on the reputation of direct observations on node SN_j , the corresponding model mentioned above is the direct trust model.

Since the direct trust relationship also has some significant fuzzy properties, we can describe the direct trust model employing the fuzzy theory. According to the data forwarding transaction table, fuzzy reputation membership based direct trust model can be defined as

$$T_{i,j}^d = \frac{\delta}{\delta + \alpha(1-\delta) + \frac{\lambda}{RF_{i,j}}} \quad (7)$$

where α denotes the weight of the past negative behavior that can be regulated to punish the malicious node action. λ represents the uncertainty trust for the weight value α .

Since the behavior of a node is not always constant but often changes in time and volatility, it is significant that the recent events are more credible than the historical events. Let $T_{i,j}^d(t-1)$ be the most recent trust evaluation and $T_{i,j}^d(\Delta t)$ be the past trust evaluation during a time interval Δt . We combine the recent events and historical events to update $T_{i,j}^d(t)$:

$$\begin{cases} T_{i,j}^d(t) = \omega_1 \cdot T_{i,j}^d(t-1) + \omega_2 \cdot T_{i,j}^d(\Delta t), \\ \omega_1 = 1 - \frac{1}{2} \cdot \zeta, \forall \zeta \in [0,1] \\ \omega_1 + \omega_2 = 1. \end{cases} \quad (8)$$

Therefore, the new trust of $T_{i,j}(t)$ is dependent on the three factors, $T_{i,j}^d(t-1)$, $T_{i,j}^d(\Delta t)$ and ζ . Then we can get the local trust updating equation,

$$T_{i,j}^d(t) = (1 - \frac{1}{2} \cdot \zeta) \cdot T_{i,j}^d(t-1) + \frac{1}{2} \cdot \zeta \cdot T_{i,j}^d(\Delta t). \quad (9)$$

However, it is arbitrary and difficult to decide whether a mobile sensor node' behavior is good or bad only based on a few interactions. Therefore, we must have an interaction threshold value of interaction times $C_{threshold}$. Consequently, the fuzzy direct trust evaluation can be computed by

$$T_{i,j}^d = \begin{cases} \frac{1}{2} \times (1 + \frac{\delta}{C_{threshold}}), & RF_{i,j} < C_{threshold} \\ \frac{\delta}{\delta + \alpha(1-\delta) + \frac{\lambda}{RF_{i,j}}}, & RF_{i,j} \geq C_{threshold} \end{cases} \quad (10)$$

When node SN_i and node SN_j has no direct relationship and cannot establish direct communication channel to exchange data, node SN_i can evaluate the trust of node SN_j based on the recommendation trust of a third party node SN_k .

As is discussed in [28], the recommendation trust and reputation model can be divided into two categories, transitivity and consensus recommendation trust and reputation management models.

The fuzzy transitivity recommendation trust and reputation model defines a degree of recommending relationship between node SN_i and node SN_j . $RR_{i,j}$ denotes the number of request recommendations, and $HR_{i,j}$ represents the number of the positive recommendations. $CR_{threshold}$ is defined as threshold value of the recommendation times. Therefore, the membership function for fuzzy recommendation trust model is defined as:

$$T_{k,j}^r = \begin{cases} \frac{1}{2} \times (1 + \frac{\eta}{CR_{threshold}}), & RR_{i,j} < CR_{threshold} \\ \frac{\eta}{\eta + \alpha(1-\eta) + \frac{\lambda}{RR_{i,j}}}, & RR_{i,j} \geq CR_{threshold} \end{cases} \quad (11)$$

where $\eta = \frac{HR_{i,j}}{RR_{i,j}} \in [0,1]$.

The different sensor nodes may provide diverse recommendations on the same nodes. That means, different nodes may have the different or even opposite trust evaluations towards the same sensor node. Assume that node SN_k gives the recommendation trust evaluation of $T_{k,j}^r$ and node SN_t provides the recommendation trust evaluation of $T_{t,j}^r$ to node SN_j . Also there have two direct trust relationships between node SN_i and node SN_k , node SN_i and node SN_t , respectively.

Here we combine the two recommendation trust evaluation and the two direct trust evaluations to make a relatively objective assessment for node SN_j ,

$$T_{i,j}^i = (D(SN_i, SN_k) \wedge R(SN_k, SN_j)) \cup (D(SN_i, SN_t) \wedge R(SN_t, SN_j)), \forall SN_k, SN_t \in SN. \quad (12)$$

Therefore, in a similar way, the fuzzy membership function of n -level fuzzy consensus recommendation trust and reputation model can be defined as

$$T_{i,j}^i = \underbrace{(R \circ D) \cup (R \circ D) \cup \dots \cup (R \circ D)}_n \quad (13)$$

In conclusion, the fuzzy local trust relationship can be calculated through the combination based on direct and indirect trust evaluation by

$$\begin{cases} T_{i,j} = X \cdot T_{i,j}^d + Y \cdot \sum_k (T_{i,k}^d \cdot T_{k,j}^r), 1 < Y < X < 0 \\ X + Y = 1 \end{cases} \quad (14)$$

where X, Y denotes the weight of direct trust value and indirect trust value in the whole fuzzy local trust value, respectively. Note that, $1 < Y < X < 0$ means that compared with the indirect recommendations, our fuzzy local trust evaluation is more focused on the direct observations. Since nodes in IoT/CPS may dynamically join in the WSNs and quit the WSNs, it stands to reason that the long historical recommendations should have relatively small weights in the Equation (14).

3.7. Global Trust Evaluation

In fact, node SN_i may have not only the direct observation on the node SN_j , but also indirect experiences by asking its acquaintances. Therefore, there are two fuzzy trust models between node SN_i and node SN_j , fuzzy direct trust model and fuzzy indirect trust model.

Obviously, if a node wants to obtain more accurate trust value with another node, it must integrate more direct and indirect experiences. Note that, the direct trust may vary with time. In order to get the most accurate trust value, we must discover the most wide indirect trust set. In this paper, the fuzzy global trust relation is defined as a union of fuzzy direct trust relation, 1-level fuzzy indirect trust relationship, 2-levels indirect trust relationship, and n -levels fuzzy indirect trust relation ($n \rightarrow \infty$).

Let us consider an example of the fuzzy trust relationship evaluation between node SN_i and node SN_j in a community of $(n+16)$ nodes, as shown in Fig.1. In the example the source node SN_i has five routes to the destination node SN_j . If we want to obtain the most accurate trust evaluation between them, all of the five routes must be contained and evaluated. Therefore, the fuzzy global trust relationship evaluation can be calculated by

$$\begin{aligned}
 T_{i,j} &= D + R \circ D + R^2 \circ D + R^3 \circ D + R^n \circ D \\
 &= (SN + R + R^2 + R^3 + R^n) \circ D.
 \end{aligned}
 \tag{15}$$

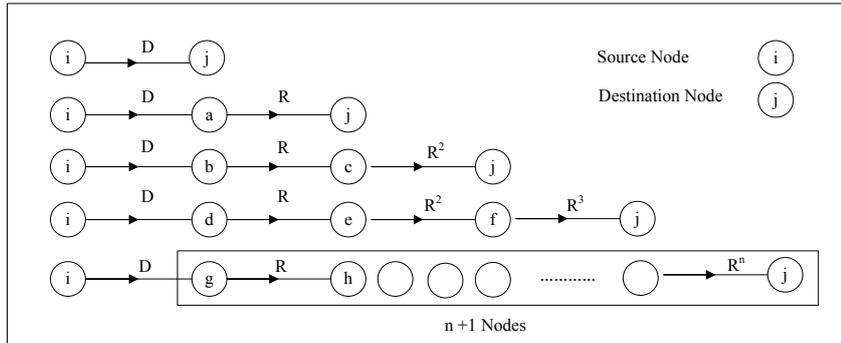


Fig. 1. Illustration of the fuzzy global trust relationship evaluation between node SN_i and node SN_j

Obviously, node SN_i can make the fuzzy global trust evaluation to node SN_j which is computed as,

$$T_{i,j} = \lim_{n \rightarrow \infty} [(SN \cup R \cup R^2 \cup \dots \cup R^n) \circ D].
 \tag{16}$$

WSNs of IoT/CPS have dynamic topologies, bandwidth constraints, variable capacity links, energy constrained operation, and limited physical security. Dynamics make it hard to evaluate behaviors, because routes in this kind of network change frequently. In this case, fuzzy global trust evaluation reflects the past interactions of the community with the corresponding node being evaluated. This evaluation is globally available to all member nodes of the community and updated each time a member node issues a new evaluation of a sensor node.

4. Simulation and Discussion

4.1. NS-3 Setup

In this paper, we perform our simulation on a NS-3 simulator [29]. Every plot is taken as an average of ten different runs. And each run is executed with source and destination pairs selected randomly from the WSN.

Since we rely on TCP acknowledgments and retransmission as indications of successful and failed package delivery events, respectively, we employ AODV protocol [30] as the communication protocol in our simulation. The

NS-3 setup parameters and model configuration parameters are listed in Table 1 and Table 2.

Table 1. NS-3 Setup Parameters.

Parameter	Value
Simulator	NS-3
MAC Layer	IEEE 802.11
Nodes Number	300
Node Placement	Random, uniform
Package Size	512 bytes
Maximum Connection	30
Transmission Range	250
Application Traffic	CBR

Table 2. Model Configuration Parameters.

Parameter	Value
ζ	0.7
α	0.75
λ	4.6
Δt	0.5
$C_{threshold}$	12
$CR_{threshold}$	12
Reply Delay	60ms

Note that, since the maximum connection number of a service node is no more than 30, $C_{threshold}$ and $CR_{threshold}$ have to be initialized as a value which is no more than $0.5 \times (0 + MAX_Connections) = 15$. Higher value of the two parameters will reduce the success rate of recommendations from neighbor nodes.

In this simulation experiment, we divide the sensor nodes into two types, good nodes and malicious nodes. Moreover, according to the behavior in route discovery, route maintenance and data forwarding, malicious nodes can be divided into two categories further. For the first type (Type 1): the malicious nodes do not perform the package forwarding function; for the second type (Type 2), the malicious nodes do not participate in the route discovery phase. Those malicious nodes are selected randomly in each run according to the setup percentage, as shown in Fig.2.

The trust and reputation relationship is initialized randomly at the very beginning of simulation. Therefore, after several rounds, we establish a similar behavior and fuzzy theory-based trust and reputation model for WSNs of IoT/CPS, where each node develops a direct reputation for each other node by making direct observations and indirect reputation between

individuals which are set up on recommendations of other individuals about these other nodes in the neighborhood.

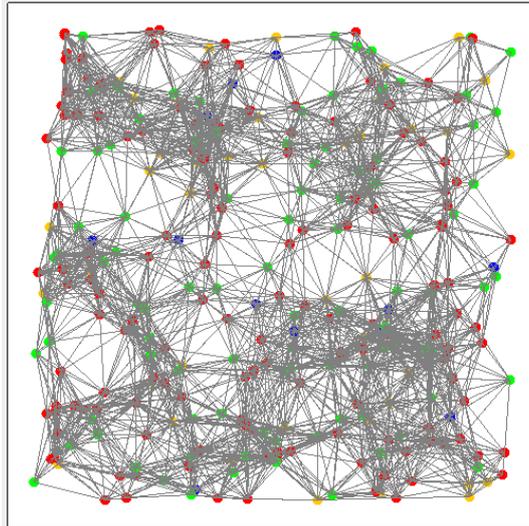


Fig. 2. The random distribution of malicious and misbehavior nodes in the simulations.

4.2. EPFR

End-to-end packet forwarding ratio (EPFR) is defined as the ratio between the number of packets received by the application layer of destination nodes to the number of packets sent by the application layer of the source node. As discussed in [28], this parameter significantly reflects the effect on the drop ratio, the path interruption repair, sending buffer overflow, interface queue overflow, the conflict MAC packet and end-to-end packet in the process of data packet. The lost packets cover all packet losses due to drops, route failures, congestion and wireless channel losses.

As shown in Fig. 1, some sensor nodes are set to be malicious nodes randomly. The percentage of malicious sensor nodes is increased and taken values from 10% to 60%, while other nodes of the network behave benevolently. The results indicate that some individual selfish nodes obviously result in the linear regression of EPFR.

Therefore, the secure mechanisms mainly focus on Type 1 to correctly perform the packet forwarding function. When 60% of the nodes follow Type 1 and Type 2, EPFR degrades by 53% and 82%, respectively. However, when the number of normal nodes becomes so smaller to a certain degree, such as 50%, the corresponding EPFR will decrease significantly.

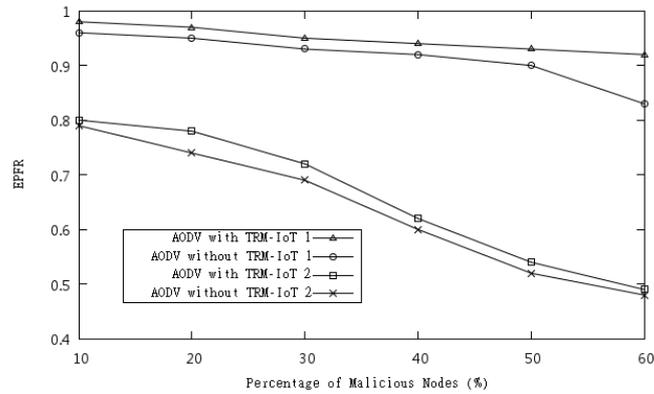


Fig. 3. The relationship between EPFR and different percentage of malicious nodes.

As shown in Fig. 3, EPFR can be degraded by malicious nodes. Through employing the proposed behavior-based trust and reputation model, the WSNs of IoT/CPS performance can be enhanced, since it enables the best loyal route selection process to avoid asking the less trustworthy nodes to forward messages.

By examination of EPFR, we can see improvements by BRM-IoT under attacks of type 1 and 2, compared to the original AODV protocol. Moreover, as the percentage of malicious nodes increases, Type 2 has a less obvious influence on EPFR than Type 1.

4.3. AEC

As shown in Fig. 4, we make malicious nodes which change between 10% and 60% of the sensor nodes in the network, increasing 10% for each running of the experiment, while the other nodes of the network behave virtuously. Since any malicious node does not participate in the route discovery phase of the AODV protocol or it not be honestly execute data packets forwarding, AEC of malicious nodes is less than that of other normal nodes.

The experimental results show that even if individual malicious nodes of type 1 seriously affect the network performance, the trust and reputation mechanism, which prompts the times of nodes transmitting data packets, is the basic security need for the non-malicious routing in WSNs. TRM-IoT model effectively cubes the malicious nodes, and significantly reduces the energy consumption of good sensor nodes.

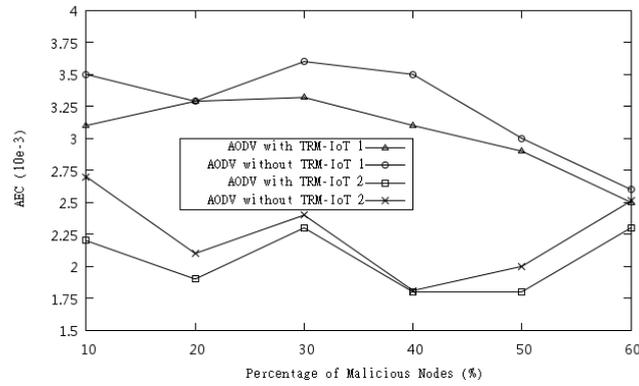


Fig.4. The relationship between AEC and different percentages of malicious nodes.

4.4. Package Delivery Ratio

Fig. 5 shows a comparison between the proposed trust and reputation scheme for IoT/CPS, TRM-IoT, and two existing trust models based on reputation mechanisms, namely DRBTS [10] and BRTM-WSN [11], in terms of PDR. In fact, package delivery ratio (PDR) is affected by packet loss and packet retransmissions. Packet loss may occur for many reasons. In this paper, we focus on the behavior that an intermediate node intentionally drops received data packets instead of forwarding them to the next hop node. From Fig. 5, we can see that the proposed trust and reputation model outperforms the other two schemes especially at higher loads on the network.

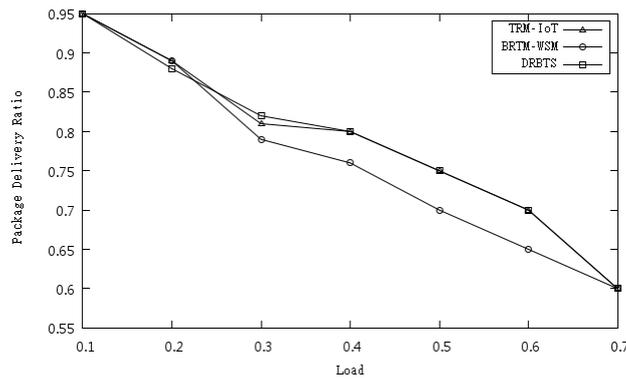


Fig. 5. The relationship between load and package delivery ratio.

4.5. Convergence Speed

Convergence speed (CS) is defined as the least number of cycles required making the number of the failed data forwarding transaction. That is, the greater of the CS, the more unfair represents that if a trust model works, the good nodes can be differentiated from the misbehavior nodes by their trust values after a few transaction cycles [31]. At the beginning, all sensor nodes have the same initial trust value, and the source sensor nodes randomly select a node for data packet forwarding. After a small numbers of transactions, the good nodes can get the higher trust value than the other bad malicious nodes.

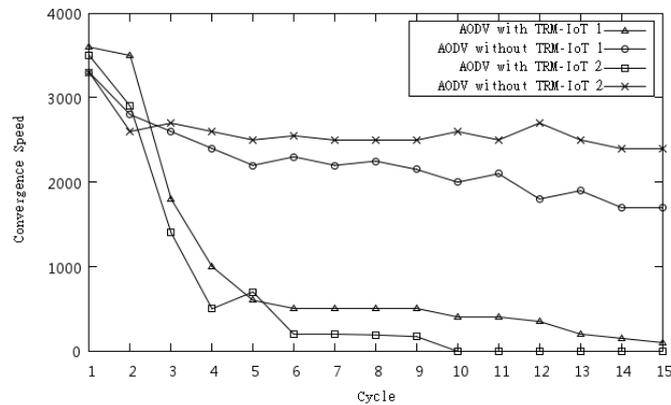


Fig. 6. The relationship between cycles and convergence speed.

The failure numbers of all data forwarding packets of the normal nodes reflect CS with the change of the simulation cycles. Since nodes always select the nodes with the higher trust values, the fewer cycles the faster the convergence of the model.

Fig. 6 describe TRM-IoT almost completely eliminates the failure of data packet forwarding after the first eight cycles in WSNs.

However, selfish nodes of Type 1 intentionally drop the received packets instead of forwarding them, and increase in the failure ratio of the normal packet forwarding increasing. The system is not the very good convergence, and has slow convergence speed in comparison with the selfish nodes of Type 2.

4.6. Detection Probability

Detection Probability (DP) indicates that whether a trust and reputation model can better handle incorrect recommendations from the third party. In Fig. 7, BRTM-WSN [11] model performs better than DRBTS [10] model. This is

because BTRM-WSN model can better handle incorrect recommendations from the third party.

Moreover, TRM-IoT model performs well than the other two existing models. This is mainly because TRM-IoT model considers the possible estimation error when evaluating the trust and reputation values. Therefore, compared with the two other existing models, our model, TRM-IoT, has better performance.

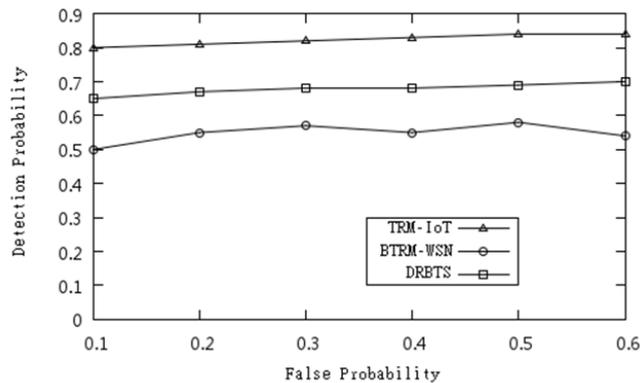


Fig. 7. The relationship between false probability and detection probability.

5. Conclusion and Future Works

Since WSNs are to be completely integrated into Internet or Next Generation Internet as a core part of IoT/CPS, it is necessary to consider various security challenges that come with IoT/CPS, such as the detection of malicious attacks.

A trust and reputation model is recognized as an important approach to defend a large distributed sensor networks in IoT/CPS against malicious node attacks, since trust establishment mechanisms can stimulate collaboration among distributed computing and communication entities, facilitate the detection of untrustworthy entities, and assist the decision-making process of various protocols.

Based on in-depth understanding of trust establishment process and quantitative comparison among trust establishment methods, this paper present a trust and reputation model TRM-IoT to enforce things cooperation in a WSN of IoT/CPS based on their behaviors. The potential benefits of employing fuzzy sets to manage trust and reputation relationships are analyzed according to the excellent NS-3 simulations.

Although the proposed model TRM-IoT has better performance compared with two other existing models, we have increasingly aware of the necessity

Dong Chen, Guiran Chang, Dawei Sun, Jiajia Li, Jie Jia, and Xingwei Wang

of eliminating the influence upon the evaluation results affected by malicious recommendation and defamation behaviors of the third party. The mechanism by which global trust is updated while local trust changes can be improved in order to be more efficient in future works.

Acknowledgement. This work is supported by the National Natural Science Foundation of China under Grant No. 60903159 and the Fundamental Research Funds for the Central Universities under Grant No. N100604012.

References

1. Wolf, W.: Cyber-Physical Systems. *Computer*, Vol. 42, No. 3, 88-89. (2009)
2. Zhu, Q., Wang, R. C., Chen Q., Liu Y., Qin W. J.: IOT Gateway: Bridging Wireless Sensor Networks into Internet of Things. In Proc. of 2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, USA, CA, Los Alamitos, 347-352. (2010)
3. Khoo, B.: RFID- from Tracking to the Internet of Things: A Review of Developments. In Proc. of 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing, Hangzhou, 533-538. (2010)
4. Joel, J. P. C. Rodrigues, Paulo, A. C. S. Neves.: A survey on IP-based wireless sensor network solutions. *International Journal of Communication Systems*, Vol. 23, Issue 8, 963-981. (2010)
5. Sun, Y. L., Yang, Y.: Trust establishment in distributed networks: Analysis and modeling. In Proc. of 2007 IEEE International Conference on Communications, ICC'07, United Kingdom, Glasgow, 1266-1273. (2007)
6. Sun, Y., Yu, W., Han, Z., Liu, K.: Trust modeling and evaluation in ad-hoc networks. In Proc. of Global Telecommunications Conference 2005 GLOBECOM'05, Vol. 3, 1862-1867. (2005)
7. Buttyan, L., Hubaux, J. P.: Stimulating cooperation in self-organizing mobile ad hoc networks. *Mobile Networks and Applications*, Vol. 8, No. 5, 579-592. (2003)
8. Boukercha, A., Xua, L., EL-Khatibb, K.: Trust-based security for wireless ad hoc and sensor networks. *Computer Communications* Vol. 30, Issues 11-12, 2413-2427. (2007)
9. Chen, H. G., Wu, H. F., Zhou, X. and Gao, C. S.: Agent-based Trust Model in Wireless Sensor Networks. In Proc. of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 119-124. (2007)
10. Srinivasan, A., Teitelbaum, J. and Wu, J.: DRBTS: Distributed Reputation-based Beacon Trust System. In Proc. of 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC'06), 277-283. (2006)
11. Marmol, G., Perez, M.: Providing trust in wireless sensor networks using a bio-inspired technique. *Telecommunication Systems*, Vol. 46, Number 2, pp 163-180. (2010)
12. Buchegger, S., Boudec, J. Y. L.: Performance analysis of the confidant protocol. In Proc. of MobiHoc'02: Proceedings of the 3rd ACM international symposium on Mobile Ad-hoc networking & computing, ACM, New York, NY, USA, 226-236. (2002)

13. He Q., Wu D., Khosla P.: Sori: A secure and objective reputation-based incentive scheme for ad-hoc networks. In Proc. of 2004 Wireless Communications and Networking Conference, Vol. 2, 21-25, 825-830. (2004)
14. Zhong, S., Chen, J., Yang, Y.: Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks. In Proc. of INFOCOM 2003, Twenty- Second Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 3, 1987-1997. (2003)
15. Hui, J. W., Culler, D. E.: Extending IP to Low-Power Wireless Personal Area Networks. IEEE Internet Computing, Vol. 12, No. 4, 37-45. (2008)
16. Liekenbrock, D.: The Internet of Things State-of-the-Art and Perspectives for Future Research. Communications in Computer and Information Science, Vol. 32, No. 2, 10-15. (2009)
17. Luigi, A., Antonio, I., Giacomo, M.: The Internet of Things: A survey. Computer Networks, Vol. 54, No. 15, 2787-2805. (2010)
18. Eshenauer, L., Gligor, V. D.: A Key-Management Scheme for Distributed Sensor Network. In Proc. of 9th ACM Conf. Computer and Comm. Security (CCS'02), United states, Washington, DC, 41-47. (2002)
19. Liu, D., Ning, P., Rongfang, L. I.: Establishing Pairwise Keys in Distributed Sensor Networks. ACM Transactions on Information and System Security, Vol. 8, No. 1, 41-77. (2005)
20. Zhu, S., Seia S., Jajodia S.: LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks. ACM Transactions on Sensor Networks, Vol. 2, 500-528. (2006)
21. Liu, F., Cheng, X., Ma, L., Xing, K.: SBK: A Self-Configuring Framework for Bootstrapping Keys in Sensor Network. IEEE Transactions on Mobile Computing, Vol. 7, No. 7, 858-868. (2008)
22. Loree, P., Nygard, K., Du, X. J.: An Efficient Post-Deployment Key Establishment Scheme for Heterogeneous Sensor Networks. In Proc of 2009 Global Telecommunications Conference, GLOBECOM 2009, United states, HI, Honolulu, 1-6. (2009)
23. Sun, Y., Trappe, W., Liu, K. J. R.: A scalable multicast key management scheme for heterogeneous wireless networks. IEEE/ACM Transactions on Networking, Vol. 12, No. 4, 653-666. (2004)
24. Gambetta, T.: Can we trust trust? In: D. Gambetta (Ed.), Trust: making and Breaking Cooperative Relations, Basil Blackwell, Oxford, 213-238. (1990)
25. Josang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decis. Support Syst, Vol. 43, No. 2, 618-644. (2007)
26. Better Business Bureau. [Online]. Available: <http://www.bbb.org>.
27. Azzedin, F., Ridha, A., Rizvi, A.: Fuzzy trust for peer-to-peer based systems. In Proc. of World Academy of Science, Engineering and Technology, Vol. 21, 123-127. (2007)
28. Luo, J. H., Liu, X., Fan, M. Y.: A trust model based on fuzzy recommendation for mobile ad-hoc networks. Computer Networks, Vol. 53, 2396-2407. (2009)
29. NS-3. [Online]. Available: <http://www.nsnam.org/index.html>.
30. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc on-demand distance vector (AODV) routing, IETF RFC 3561, July 2003.
31. Griffiths, N., Chao, K. M., Younas, M.: Fuzzy trust for peer-to-peer systems. In Proc. of 26th IEEE International Conference on Distributed Computing Systems Workshop, Portugal, Lisboa, 73-73. (2006)

Dong Chen, Guiran Chang, Dawei Sun, Jiajia Li, Jie Jia, and Xingwei Wang

Dong Chen is a PhD candidate at the school of Information Science and Engineering, Northeastern University, Shenyang, China. He received his MSc in Computer Science from Northeastern University in 2010. His current researches interests include Internet of Things, Cyber Physical System.

Guiran Chang received his Ph.D. degree in electrical engineering from the University of Tennessee, Knoxville, Tennessee in 1991. He is currently a Professor at the computing center of Northeastern University, China. His current research interests include computer networks, Internet of Things and information security.

Dawei Sun is a PhD candidate at the school of Information Science and Engineering, North-eastern University, China. He received his MSc in Computer Science from Northeastern University in 2009. His current researches interests include cloud computing and virtualization technology.

Jiajia Li is a PhD candidate at the school of Information Science and Engineering, Northeastern University, China. He received his MSc in Computer Science from Northeastern University in 2010. His current researches interests include Spatial-Temporal Database and XML Database.

Jie Jia received her Ph.D degree in computer science from Northeastern University, China. She is currently an Associate Professor at the School of Information Science and Engineering, Northeastern University, China. Her research interests are mainly on RFID systems and Wireless Sensor Network.

Xingwei Wang received his Ph.D degree in computer science from Northeastern University, China in 1998. He is currently a Professor at the School of Information Science and Engineering, Northeastern University. His research interests are mainly on routing algorithms and protocols, mobility management in NGI.

Received: March 3, 2011; Accepted: April 22, 2011.

A Reusable Agent Design Pattern with Flexibility and Extensibility

Hao Lan Zhang¹, Wenhua Zeng², and Christian Van der Velden³

¹ School of Management, NIT, Zhejiang University,
No. 1 Qianhu South Road, Ningbo 315100, China
haolan.zhang@nit.zju.edu.cn

² School of Software, Xiamen University,
361005 Xiamen, China.
whzeng@xmu.edu.cn

³ BAE Systems Australia, 40 River Boulevard
Richmond, VIC, 3121, Australia
christian.vandervelden@baesystems.com

Abstract. Intelligent agent-based systems are regarded as the promising technology in bridging the gap between the physical world and cyber-applications. In spite of the rising demands for reusable information systems; current designs are still insufficient in providing efficient reusable mechanisms for system design. One of the major problems hinders the development of information reuse in most traditional systems is the lack of the autonomous character among system modules or subsystems. The emergence of agent technology is able to solve the problem plaguing many traditional systems. Existing agent design models create an agent as a sole system with built-in domain-specific capabilities. However, this design pattern causes several problems while matching and updating agents' capabilities due to the built-in design pattern in these models decreases agents' extensibility, flexibility and reusability. In this paper we introduce a novel design for agent-based systems, which is able to provide an efficient design pattern for improving the reusability, extensibility and flexibility of agent design. The novel agent capability design offers an open and flexible structure; and implements several practical algorithms that can improve the system performance. An experimental program based on several practical cases has been developed to evaluate the performance of the proposed design. The empirical results reveal the efficiency of the new agent design pattern.

Keywords: Agent Capability Design, Agent Reusability, Domain Specific Components, Agent Design.

1. Introduction

Information reuse has become a key issue for information system designers in order to reduce information redundancy and system development expenses. Various systems have implemented reusable mechanisms; agent-based systems are among these systems with fast-growing demands for information reuse. In this paper we introduce a component-based design for agent capability reuse, i.e. the Domain Specific Component (DSC) design. The mechanisms used in this design can be generally applied to various agent-based systems for capability reuse.

Many existing agent-based systems have been focusing on developing service-oriented agents or component-based agents for complex problem-solving processes [1, 2, 3, 4]. These systems adopt various mechanisms to enhance the system reusability and flexibility. However, agent capabilities developed in these systems are generally integrated to agents and together form a complete component. This design pattern decreases the flexibility and reusability of agents. Under the circumstances, this paper proposes a new mechanism that could efficiently improve the reusability of task-oriented agents.

Unlike traditional agent design models, the DSC structure design deploys the novel slot-item structure, which is supported by several practical algorithms in order to improve the reusability and flexibility. Figure 1 shows the design pattern for a DSC-based agent. The DSC items are the elements of agent functionalities. In other words, an agent's capabilities are based on the DSC items that it carries.

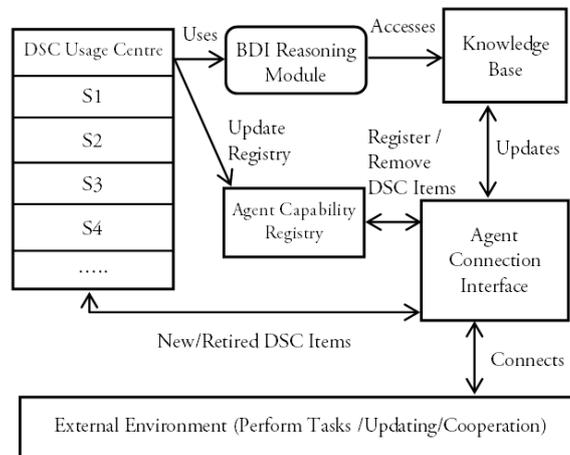


Fig. 1. General agent design pattern of DSC-based agents

A centre is deployed in this design to coordinate various agents. This centre maintains a large DSC warehouse, which enables agents to upgrade their capabilities through receiving the latest DSC items from the DSC warehouse. This design incorporates agent-agent communication (through

the agent-agent connection interface in Figure 1) and centre-agent communication (through the centre-agent connection interface in Figure 1) to form a hybrid structure, which combines decentralised and centralised framework. The performance of the hybrid structure for agent cooperation has been proven as a superior solution [5].

The DSC-based agent design adopts Beliefs-Desire- Intentions (BDI) model for agent reasoning as agent capabilities can be formalised in a framework [6].

A DSC-based agent can efficiently update its capabilities through updating its DSC items. The DSC-based agents are also efficient for agent matching since DSC items provide explicit descriptions about agent capabilities. The retired or dated DSC items are returned to the DSC warehouse; however they can be reused anytime through plugging back to the agent's DSC slots. In general, the DSC-based agent design is able to improve the efficiency of agent capability reuse and provide a flexible and upgradeable structure for integration. This design overcomes the adaptation and integration problems that plague existing software reuse systems [7, 8]; it provides a predefined input and output structure to minimise the costs of components standardisation that plagues many systems [9]. In addition, this design cost-effectively recycles dated DSC items rather than eliminating them from systems.

2. Related Work

Several mechanisms have been suggested to improve the reuse of capabilities and tasks in agent-based systems. For instance, the major mechanisms of describing agent capabilities for information reuse include the Language for Advertisement and Request for Knowledge Sharing (LARKS), Agent Capability Description Language (ACDL) [10], and Interface Communication Language (ICL) [11].

ACDL is introduced to maximise the reuse of agent capabilities over new application domains. It is based on the Knowledge Modelling Framework (KMF). The LARKS allows agents to advertise their capabilities for both syntactic and semantic matching processes. LARKS-based agents are able to use application domain knowledge in any advertisement and request [12].

Modelling of component-based systems is still regarded as a largely unresolved problem in many object-oriented systems according to [13]. The emergence of intelligent agents is helpful to solve the problems in object-oriented systems since agents can be specified on a conceptual level instead of an implementation level. Moreover, agents have strong adaptability and self-learning capability, which make the component integration process in agent-based systems much efficient than in object-oriented systems. The design principles for building component-based agents have been described in [3], which provide several preliminary mechanisms to enhance the reusability of agent design. The „agent specific task“ component described in

their study is similar to the DSC concept. However, it does not provide more concrete mechanisms about the design procedures and the performance of the design.

Previous research on developing an open and comprehensive agent structure has addressed some fundamental issues including the reusability issue for agent design [11, 14, 15]. The reusability issue for agent design is also related to the other issues, such as agent matchmaking, service advertisement, learning and adaptivity, etc. These issues help to draw the basic guidelines for designing DSC-based agents.

3. DSC Usage Centre Overview

3.1. DSC Usage Centre – A Slot Container

Each agent has a DSC usage centre, which provides information resources for agents. This component distinguishes the DSC-based agents from existing middle agents or agent facilitators that mainly play as the role of an agent coordinator. The DSC usage centre upgrades the agents' functionalities through acquiring information from the external environment and sending the acquired information in DSC formats back to the centre. The DSC usage centre manages the unused and active DSC items for agents to update their capabilities. Each DSC item is executable and has standardised and predefined inputs and outputs.

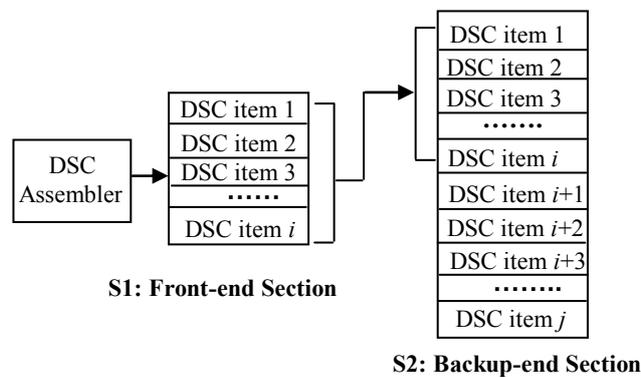


Fig. 2. Slot design of a DSC usage centre

Each DSC usage centre in an agent is a slot container, which offers numerous slots for containing the specialising domain components as shown in Fig 2. Each specialising domain component possesses some special capabilities. Each domain component can plug into a DSC slot to perform specific tasks as an item, for example a domain-component can connect the

agent knowledge base to a stock market database to acquire useful knowledge for the agent.

3.2. Inputs/Outputs of DSC Items

The inputs and outputs of a DSC item contain information as shown in Fig 3.

Input Section: [Input 1: String (Compulsory); Input 2: Integer; Input 3: Float; Input 4: Float (Compulsory).] Output Section: [Output 1: String; Output 2: Float.]

Fig. 3. Inputs and outputs in a DSC item

In the input section, the information includes the input content, data type, and constraints. The input content indicates the value of the inputs such as a string or a number. The data type indicates the input content's data type, which include string, integer, float, double, etc. The constraint indicates whether the input content can be null. If the constraint value is „compulsory“ then the input content must not be null. Otherwise the input content can be null. The output section consists of the output content and data type; they basically have the same meaning as the input section.

A DSC assembler is employed in the DSC usage centre to evaluate the suitability of a DSC item for an agent through comparing the inputs and outputs from both sides. It also plays the role of extracting top-value of DSC items from the backend section of the DSC usage centre. The comparison between a DSC item and an agent's request is mainly based on the data types and the number of inputs and outputs. For instance, an agent requests a DSC item that can provide two main outputs including the total sales amount and the average salary. Then a selected DSC item must provide these two outputs. The following procedures show the comparison process:

Step 1: If the agent's request does not specify inputs then skip to the next step, otherwise we compare the length of the inputs of the agent's request and the DSC item. We eliminate all the non-compulsory inputs in both DSC items and agent's requests. If the length of compulsory inputs (*LC*) of both sides is equal then we continue the comparison process, otherwise the DSC item is not appropriate for fulfilling the agent's request. The following example illustrates the verification process.

Agent's request inputs:	1	0	0	0	1
DSC item inputs:	1	1	0	0	1
			↓		
Agent's compulsory inputs:	1	1			
DSC compulsory inputs:	1	1	1		

„0“ denotes that the input is non-compulsory;
„1“ denotes that the input is compulsory.

According to the above example, we have: $LC_a = 2$ and $LC_d = 3$, where LC_a denotes the length of the compulsory inputs of the agent's request; LC_d denotes length of the compulsory inputs of the DSC items. Here, $LC_d > LC_a$. Hence, the selected DSC item is not appropriate for the agent's request.

If $LC_d = LC_a$, then we compare the data types of the compulsory inputs from both sides. If $DIA_i = DID_i$, then go to the next step, otherwise the selected DSC item is not appropriate for the agent's request. DIA_i denotes the input data types of the agent's request; DID_i denotes the input data types of the DSC items; i is from 1 to LC_d , with the incremental change = 1.

Step 2: If the agent's request does not specify outputs then the verification process is accomplished. Whether the DSC item is appropriate for the agent's request is based on the semantic matching of the capabilities descriptions between the two sides, and the comparison process performed in the previous step. If the agent's request specified the output, then we compare the length of the outputs of both sides. The outputs do not distinguish the compulsory and non-compulsory values; therefore, it is not necessary to perform the elimination procedure of non-compulsory values.

If $LP_a \neq LP_d$, then the DSC item is not appropriate for the agent's request. If $LP_a = LP_d$, then we compare the data types of the outputs from both sides. If $DOA_i = DOD_i$ then the DSC item can be selected as an item for the agent's request, otherwise the selected DSC item is not appropriate for the agent's request. LP_a denotes the length of the outputs of the agent's request; LP_d denote the length of the outputs of the DSC item; DOA_i denotes the output data types of the agent's request; DOD_i denotes the output data types of the DSC items; i starts from 1 to LC_d , with the incremental change = 1.

3.3. Functionality Redundancy Calculation

The DSC normally communicates with the environment through the agent-to-agent (or central component-to-agent) interfaces; it also can establish communication with the environment directly. A problem arises when some plugged components are rarely used or never used. To solve this problem, a component usage evaluation mechanism is used to examine the usage efficiency of a plugged component. There are two major factors affecting a DSC's usage efficiency, which include the usage frequency factor and functionality similarity factor. The usage frequency indicates the total number of visits to/from DSC items. The functionality similarity indicates the possible redundancy of a plugged item's functionality with the other plugged items' functionalities. In this section, we introduce the mechanism for calculating functionality redundancy for DSC items. The usage frequency calculation process will be introduced in the next section.

We deploy a three-type semantic relationship model, which is used in WordNet [16], to describe the similarity relationships between two words. The three types of relationships [17] are:

- (i) Synonym: two words are synonymous.
- (ii) IS-a: two words are in a superset and subset relationship.
- (iii) Has-a: One word has ownership of another word. Also known as part-whole relationship between words.

If two words are synonym relationship then their similarity value is A_1 ; if two words are IS-a relationship then their similarity value is A_2 ; if two words are Has-a relationship then their similarity value is A_3 ($A_1 > A_2 > A_3$). Therefore, we have the following equation for similarity calculation:

$$F = \sum_{i=1}^n A_i \tag{1}$$

where F denotes the DSC functionality redundancy value of a slot; A_i the similarity value; n denotes the total slot number of a DSC; F can be further expanded through decomposing the task description of a plugged item into several key terms and comparing these key terms with other plugged items' key terms, see Fig 4.

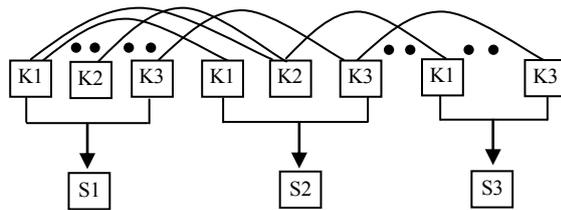


Fig. 4. Functionality similarity calculation based on Cartesian-Product method

In Fig 4, the functionality-similarity value of S_1 and S_2 , namely A_i , is based on the *Cartesian-Product* [18] method. In this method, A_i is calculated by exhausting all combinations of choosing one key term from S_1 and one key term from S_2 . A_j represents the functionality-similarity value of S_1 and S_3 . Therefore, the overall functionality-similarity value of S_1 is the composition of A_i and A_j .

$$F = \sum_{\substack{m,n \in C_1 \\ m \neq n}} \sum_{i,j \in C_2} S(L_m(K_i), L_n(K_j)), \tag{2}$$

where m and n denote the DSC slot number (e.g. S_1, S_2, S_3 in Fig 4), m must not equal to n because the key terms are only calculated with similarity to other slots but not the same slot; i and j denote the key term number (e.g. K_1, K_2, K_3 in Fig 4); $S(x, y)$ is a function to calculate functionality similarity between x and y ; $L_x(K_y)$ denotes key term K_y in slot L_x ; C_1 denotes the total slot number of the DSC; C_2 denotes the set of decomposed key terms.

If a plugged DSC item has high F value compared with other items, and there is no slot for a new item then the high F item will be unplugged from the DSC and sent to the central component (the DSC learning centre) for reuse because this item is highly redundant in the DSC compare with other items. A new DSC item will be selected and plugged to the vacant DSC slot. The new DSC item will also be evaluated its similarity to other slots. If the new average similarity value is smaller than the previous average similarity value (before the previous DSC item is removed), then the new item will be plugged in. Otherwise, another DSC item will be selected until it can decrease the average similarity value. This design can enhance the reusability and efficiency of an agent-based system, and it also reduces the redundancy in the DSC.

4. Usage Frequency Calculation for Agent Capability Reuse in DSC Usage Centre

4.1. Front/Back End Structure

The DSC usage centre consists two sections, which include the front-end section and the back-end section. The front-end section extracts the DSC items from the backend section, which are the most frequently used DSC items. This two-section-based structure adopts the methodology of the CPU cache design in operating systems, which stores copies of the data from the most frequently used main memory locations. The reason for dividing the DSC usage centre into two sections is that: the number of the DSC items in the DSC usage centre could be large; however, there are only a number of DSC items are used frequently within a certain period. Therefore, it can improve the system efficiency through deploying a section with a relatively smaller size, which contains the most frequent used items within a certain period.

Fig 5 illustrates the structure the DSC usage centre. S_1 denotes the front-end section; S_2 denotes the item backend section; all the DSC items in S_1 can be found in S_2 , which can be expressed as: $S_1 \subseteq S_2$. The DSC items stored in S_1 are obtained from S_2 , which are the most frequently used items in S_2 . The DSC assembler is responsible for searching a DSC item and importing it to the requesting agent. The following calculation processes illustrate how to extract the DSC items in S_2 and store to S_1 .

Step 1: Calculating the recent visit factors in S_2 within time period $[t_a, t_b]$.

$$V(t_i) = \begin{cases} 1, & \text{the DSC item is visited,} \\ 0, & \text{the DSC item is not visited.} \end{cases}$$

where t_i is the time and $V(t_i)$ denotes whether a DSC item is visited at time t_i ; i is a variable, which denotes different visiting time within a time period.

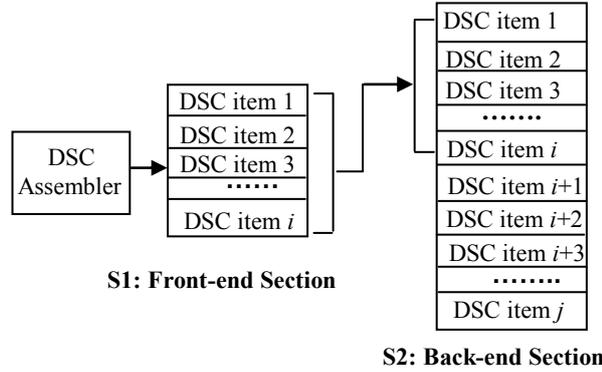


Fig. 5. Two-section-based structure of the DSC usage centre

The total number of visits to a DSC item is calculated as:

$$n = \sum_{i=a}^b V(t_i), \text{ and } V(t_i) = 1. \tag{3}$$

where a, b denote the starting point and the ending point of the time period respectively; n is the total number of visits to a DSC item within the time period $[t_a, t_b]$; t_a is the starting point; t_i is the time point that the DSC item is visited within the time period. If a DSC item is visited at time t_i within $[t_a, t_b]$, then this DSC item is the i^{th} visit to the DSC item and $1 \leq i \leq n$. For instance: there are a total of 100 visits within $[t_a, t_b]$, i.e. $n = 100$. The DSC item is visited at time t_{20} ($t_a \leq t_{20} \leq t_b$) and is the 20th visit to the DSC item, then $t_i = t_{20}$. All t_i and t_b are the converted values, which are subtracted by t_a and convert into seconds or a user-defined time scale.

Therefore, a DSC item's recent visit factor can be calculated as the following:

$$\begin{aligned} E_i &= \sum_{i=1}^n \left(V(t_i) + \frac{t_b - t_i}{t_i} \times V(t_i) \right)^{\log_n \left(\frac{t_b \times n}{t_i} \right)} \\ &= \sum_{i=1}^n \left(\frac{t_b}{t_i} \times V(t_i) \right)^{\log_n \left(\frac{t_b \times n}{t_i} \right)} \\ &= \sum_{i=1}^n \left(\frac{t_b}{t_i} \times V(t_i) \right)^{(\log_n t_b - \log_n t_i + 1)} \end{aligned} \tag{4}$$

where E_i denotes the recent-visit-factor value of a DSC item at time t_i , which indicates a DSC item's usage within the period $[t_a, t_b]$; n is the total number of visits to the DSC and $1 \leq n$, (Eq. 3, 4 only calculate the recent-visit-factor value when there is visit to the DSC item, if $n = 0$ then $E_i = 0$.); the exponent in Eq. 4, i.e. $(\log_n t_b - \log_n t_i + 1)$, is to enlarge the recent-visit-factor value. If t_i is more recent then its E_i value is greater, meanwhile log function is used to limit the exponent value. Eq 4 indicates that: when t_i increases (more recent) then E_i increases. In other words, if a visit to the DSC item is more recent then its E_i is greater. The DSC-based system developers or users can define the time period (i.e. $[t_a, t_b]$), which is configurable, to initialise and update the DSC items in S_1 .

The DSC items in S_2 are ranked according to E_i . The DSC items with higher E_i scores are listed on the top of S_2 ; S_1 extracts a number of the DSC items that are listed on the top of S_2 . The next step calculates the number of the DSC items that S_1 extracts from S_2 (i.e. the size of S_1).

Step 2: Calculating the size of S_1 .

The size of S_1 is based on the usage frequency of the target system's DSC item; it should also take the miss-rate into consideration. We use a dynamic alteration method to determine S_1 's size. An empty S_1 extracts the maximum number of the DSC items with top E_i values from S_2 within S_1 's predefined size limit. After a period $([t_0, t_s])$ of running (initial running period), some DSC items in S_1 will be replaced by the items in S_2 and some will be removed from S_1 because of low-usage-efficiency (refer to B section). We first calculate the total number of DSC items in S_2 as its size, i.e. A_2 . The initial step of this process is set a target miss-rate value and according to the miss-rate value and A_2 to calculate the preliminary size of A_1 .

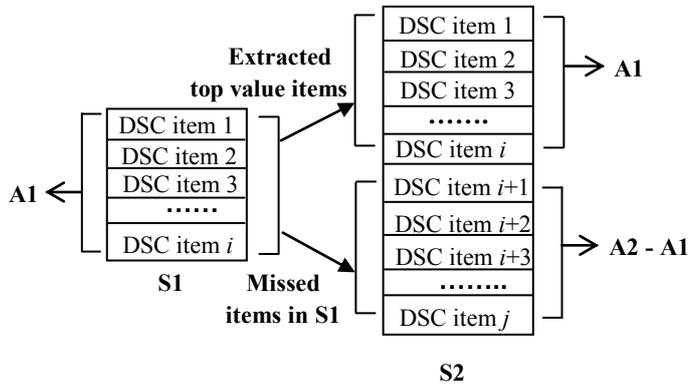


Fig. 6. Dynamic alteration process

Once E_i , A_1 , and A_2 are calculated, the DSC usage centre starts to dynamically alter S_1 's size according to the miss-rate of searching DSC items in S_1 within a user defined period, i.e. the alteration period. This is because the E_i value of each DSC item is normally changing during the alteration

period and the change affects the size of S_1 . Fig 6 illustrates the dynamic alteration process.

In the dynamic alteration process, A_1 is affected by E_i values and the miss-rate of searching DSC items in S_1 . The following equation describes the calculation process.

$$\begin{aligned} \frac{\sum_{x=1}^{A_1} E_x}{\sum_{y=1}^{A_2-A_1} E_y} &\approx \frac{A_1}{(A_2 - A_1)} \times \frac{1}{ms} \\ \Rightarrow A_1 &\approx \frac{\sum_{x=1}^{A_1} E_x \times (A_2 - A_1) \times ms}{\sum_{y=1}^{A_2-A_1} E_y} \\ \Rightarrow A_1 &\approx \frac{ms \times \sum_{x=1}^{A_1} E_x \times A_2}{\sum_{y=1}^{A_2-A_1} E_y + \sum_{x=1}^{A_1} E_x \times ms} \end{aligned} \quad (5)$$

where ms denotes the miss-rate of searching DSC items in S_1 within the alteration period (this period can be various according to different system applications), and it is always ≤ 1 ; E_x and E_y denote the E_i values of the DSC items in A_1 and $(A_2 - A_1)$ sections, respectively.

This equation indicates that: A_1 should be approximately equal to $(A_2 - A_1)$ multiplied by the miss-rate, and the ratio of the total E_i values of all the DSC items in A_1 section to the total E_i values of all the DSC items in $(A_2 - A_1)$ section. In this equation, A_1 is alterable to satisfy the target ms value.

If initial A_1 does not satisfy this equation, then the DSC usage centre needs to alter A_1 size until it approximately equals the right side of Eq.5. The system users can define the alteration period based on different application requirements. For instance, the alteration period could be longer in some applications because their DSC items require more processing time. Moreover, the process is dynamic, which allows the DSC usage centre to alter the S_1 size regularly according to the computation results based on Eq.5.

The above two steps explain the operating process of the DSC usage centre. The system users can configure the DSC usage centre through adjusting the processing time and the actual database size of the DSC usage centre.

4.2. Reuse / Dismissal of DSC Items

The DSC items in the DSC usage centre that have low usage efficiency will be sent to the DSC warehouse for reuse. A DSC item is regarded as low usage efficiency when this DSC item is constantly on the bottom of S_2 within a period of time $[t_s, t_{re}]$, which is called the critical period for removal and it can be configured by system users.

The usage efficiency is calculated through combining the probability of the visits to a specific item with the recent-visit-factor value E_i . We have assumptions as follows:

There are a total of x DSC items in an agent.

All the DSC items in the agent have been visited the total of m times within period $[t_a, t_b]$.

There is no concurrent visit to the DSC items. In other words, only one DSC item can be visited at the same time.

Value m is based on Eq. 3. Therefore, $m = \sum_{i=1}^b V(t_i)$, and $V(t_i) = 1$. Based on Eq. 4, the average value of all the DSC items' E_i values, i.e. $AVG(E)$, within $[t_a, t_b]$ is:

$$AVG(E) = \frac{\sum_{i=1}^m \left(\left[\frac{t_b}{t_i} \times V(t_i) \right]^{(\log_n t_b - \log_n t_i + 1)} \right)}{m} \quad (6)$$

The standard deviation [19] of all the DSC items' E_i value is:

$$SDV(E_i) = \sqrt{\frac{1}{m} \sum_{i=1}^m \left(\left(\frac{t_b}{t_i} \times V(t_i) \right)^{(\log_n t_b - \log_n t_i + 1)} - AVG(E) \right)^2}$$

$$= \sqrt{\frac{1}{m} \left(\sum_{i=1}^m \left(\frac{t_b}{t_i} \times V(t_i) \right)^{2(\log_n t_b - \log_n t_i + 1)} - 2AVG(E) \sum_{i=1}^m \left(\frac{t_b}{t_i} \times V(t_i) \right)^{(\log_n t_b - \log_n t_i + 1)} + m(AVG(E))^2 \right)}$$

$$= \sqrt{\frac{1}{m} \left(\left(\sum_{i=1}^m \left[\frac{t_b}{t_i} \times V(t_i) \right]^{2(\log_n t_b - \log_n t_i + 1)} \right) - m(AVG(E))^2 \right)}$$

$$= \sqrt{\frac{1}{m} \sum_{i=1}^m \left(\frac{t_b}{t_i} \times V(t_i) \right)^{2(\log_n t_b - \log_n t_i + 1)}} - (AVG(E))^2 \quad (7)$$

We consider removing the DSC item with an E_i value far below $AVG(E)$. Therefore, the lowest E_i value is:

$$E_k = MIN(E_1, E_2, \dots, E_m),$$

where dataset (E_1, E_2, \dots, E_m) denotes all the DSC items' E_i values within the period $[t_a, t_b]$; E_k denotes the lowest E_i value in the dataset. If we have:

$$\frac{|E_k - AVG(E)|}{SDV(E)} \geq 3 \quad (8)$$

We define this DSC item as low usage efficiency. This definition is based on *Chebyshev's* theorem [20]:

"If μ and σ are, respectively, the mean and the standard deviation of the distribution of the random variable x , then for any positive constant k the probability that x will take on a value which is at most $\mu - k\sigma$ or at least is less than $\mu + k\sigma$ or equal to $1/k^2$."

This theorem is also expressed as: $P(|x - \mu| \geq k\sigma) \leq 1/k^2$. According to *Chebyshev's* theorem, at least 89% of the E_i values disperse within 3 standard deviations from the $AVG(E)$ value. Therefore, any value, which is greater than this range, is considered as an outlier that has low-usage-efficiency.

If a DSC item in an agent's DSC usage centre is identified as a low-usage-efficiency item, then it will be sent to the DSC warehouse. These low-usage-efficiency DSC items can be discovered and re-deployed by other agents, or can be updated by the DSC warehouse through information updating.

The DSC-based agents also use this evaluation methodology to identify whether a DSC item is low-usage-efficiency. If a DSC item is identified as low-usage-efficiency in an agent then it will be unplugged from the DSC container and sent to the DSC usage centre. This reuse and dismissal method is also used to remove the low-efficient items in the front-end section of the DSC usage centre; it can reduce the section size and improve the system efficiency.

5. Experimental Results

Many existing agent-based systems have the difficulties in designing efficient processes for agent capability matching and reuse. The DSC-based design aims to provide efficient mechanisms for agent capability matching and reuse. In order to evaluate the performance of the DSC-based mechanisms, we conducted a set of experiments based on a real case scenario for solving

the automated feature recognition problem in aerospace component design process, as shown in Fig 7 [21].

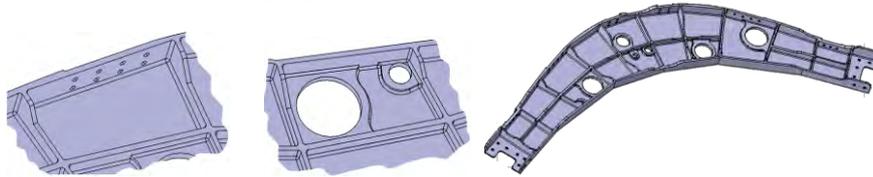


Fig. 7 Typical CAD models of stiffener-panel design.

These experiments demonstrate how the agent-based technology, in particular the DSC mechanism, can be used in the real world engineering design processes.

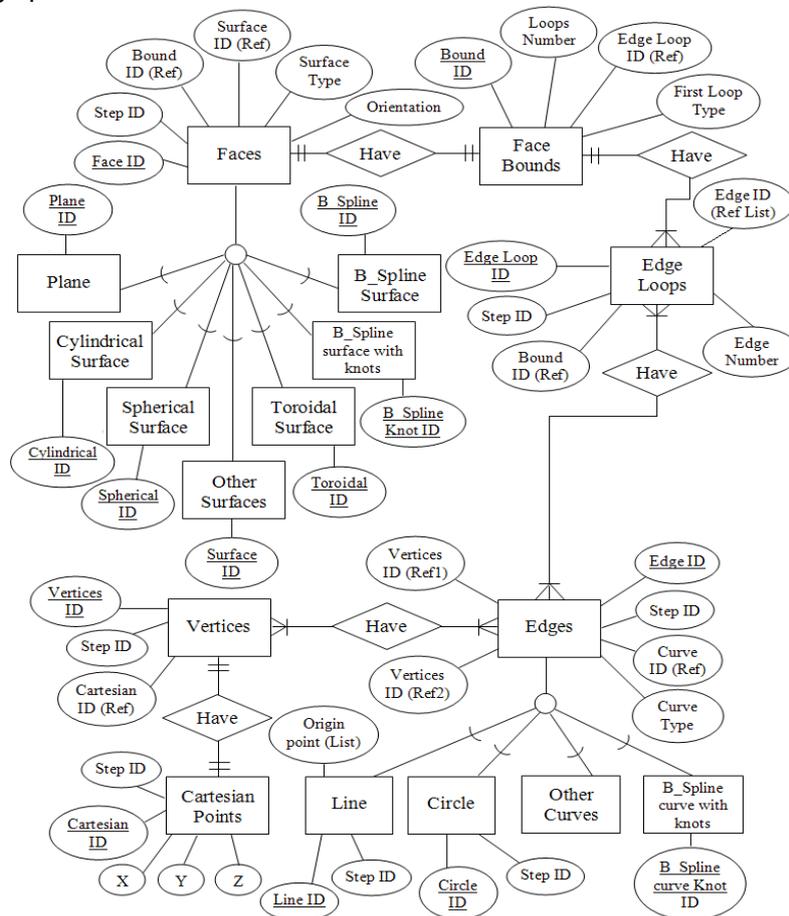


Fig. 8. Entity relationship diagram based on common CAD models [21].

In this case, agents extract key components from CAD models such as the models listed below. There are a number of domain specific agents with different capabilities including the Face agent, Panel agent, Stiffener agent, Hole agent, etc. These agents are responsible for extracting their specified components.

The experimental program matches the user requests against the information from the CAD-based database. The functionalities of the automated feature recognition system are transformed into DSC-based items, i.e. agent capabilities. The experimental program consists two parts: the first part evaluates the performance of the functionality redundancy calculation mechanism; the second part evaluates the performance of the two-section-based structure.

The real case scenario is based on several CAD models for aerospace component design. Domain agents using the DSC-based design can improve the efficiency of finding appropriate agent capabilities for performing the feature recognition process. An entity-relationship diagram based on common CAD models is shown in Fig 8.

An example table describes the three-type relationship is deployed in the functionality redundancy calculation process. The example relationships are shown in Table 1.

Table 1. Similarity relationships used in the FRC experiment

Comparison words	Similarity relationships
Plane Face, Face	IS-a
Cylindrical Face, Face	IS-a
Face, Face Bound	Has-a
Advance Face, Face	Synonym
Face Bound, Edge Loop	Has-a
Edge Loop, Edge	Has-a
Line, Edge	IS-a
Circle, Edge	IS-a
B Spline curve, Edge	IS-a
.....
Panel, Panel Face	Has-a

In the first 16 sets of the *Functionality Redundancy Calculation* (FRC) experiments, we evaluated the impact of request number on the success rate. Among the 16 sets, 8 sets based on the FRC mechanism are called the FRC sets; another 8 sets without using the FRC mechanism are called the NFRC sets. In each experiment, a request is generated randomly based on a knowledge base, which holds around 103 capability descriptions. The DSC usage centre is also generated based on the knowledge base. The DSC slot number is alterable; system users can increase and decrease the DSC slot

number according to specific system requirements. In the first part of the FRC experiment, the default DSC slot number in an agent is 3 and there are 5 agents to deal with a request corporately. As shown in Table 1, the relationship table contains 39 three-type relationship descriptions.

The FRC-based sets produce the similarity score of each DSC item compared with other agents' DSC items. If the DSC item has the highest similarity score and is 1.5 times greater than average score of all the participated agents, then this DSC item will be replaced by a new DSC item from the DSC warehouse. The boundary value of trigger a DSC replacement is 1.5 times, which is based on experimental experience and the relevance of the requests.

The NFRC-based sets only match the requests with the DSC item descriptions. The DSC items in NFRC sets will not be replaced or changed in the experimental process. The following figures show the experimental results.

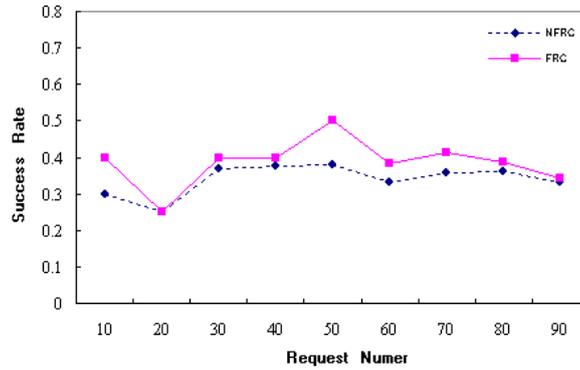


Fig. 9. Success rate comparison based on FRC and NFRC (15 DSC items).

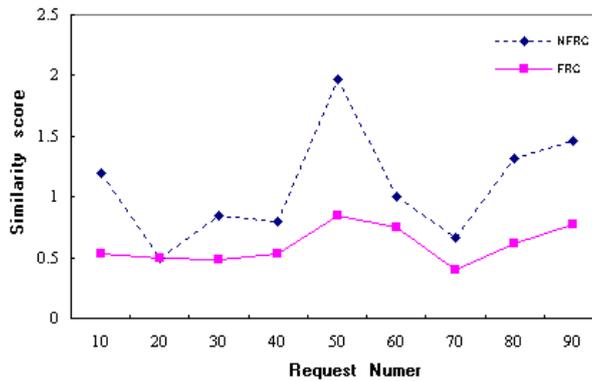


Fig. 10. Similarity score comparison based on FRC and NFRC (15 DSC items).

Fig 9 shows the impact of the FRC method on the success rate of matching requests in comparison with the experimental sets without using

FRC method (i.e. NFRC sets). Fig 10 presents the relationship between the average similarity score and requests number based on FRC and NFRC. The requests matching process in FRC is more efficient than NFRC since the all FRC-based success rates based on different request number are greater than the NFRC-based success rates. The improvement on success rates for a single agent is not enormously large based on FRC. However, this improvement ought to be enormous based on a large number of agents. We also noticed that the success rate is improved significantly when total requests number is 50. The reason for the increase at 50 point is still under investigation. We presume the reason could be: 50 requests for 15 DSC items is a balanced ratio for matching in a DSC usage centre.

Figure 10 shows the average similarity scores based on the NFRC sets is clearly greater than the FRC sets. The average score for NFRC is 1.8 times higher than FRC, which reflects the functionality redundancy in the NFRC sets is much higher than the FRC sets. The functionality redundancy in the DSC usage centre could cause the low efficiency in terms of capability matching and memory consumption for a multi- agent system.

To further evaluate the performance of FRC, we altered the DSC item number (i.e. the DSC slot number) in a DSC usage centre to observe its impact on the success rates of request matching based on 200 requests and remain the other parameters unchanged that are: 39 relationship descriptions and 103 capability descriptions. The results are shown in Fig 11 and 12.

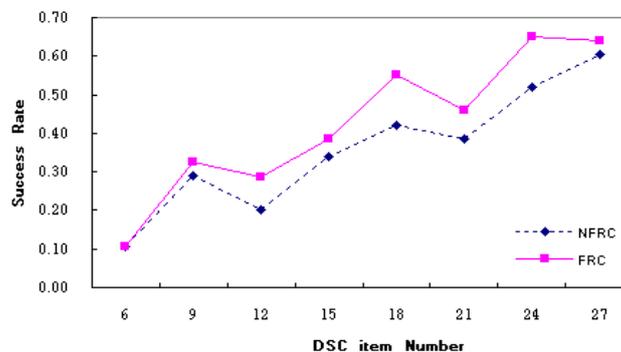


Fig. 11. Success rate comparison based on NFRC and FRC (200 requests)

Fig 11 indicates that the success rates are increasing in both FRC and NFRC sets. Overall, the success rates in the FRC sets are still higher than the NFRC sets. The disadvantage of the FRC method is that: the time consumption of the FRC sets is higher than the NFRC sets as shown in Figure 12. The time consumption in the FRC sets is caused by the processes of calculating the similarity scores, selecting an appropriate DSC item from the DSC warehouse, and replacing the low- efficiency DSC item.

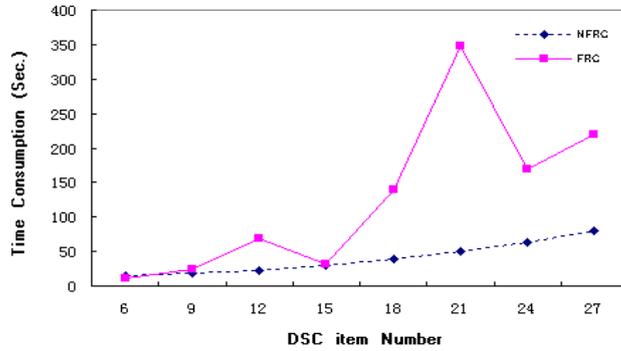


Fig. 12. Time consumption based on NFRC and FRC (200 requests)

In this experimental design, the requests are generated randomly; and this makes the miss-match happens more often than real applications. In other words, the frequency of requesting for a DSC item in a system has a routine pattern in practical. For instance, a „connecting to the Internet“ DSC item might be used most frequently in many organisations, but the random request generation process in this experimental design does not take this factor into account. Therefore, the miss- match will be highly decreased in real cases.

In the second part of the experiments, we evaluate the performance based on the two-section (TS) and Non-two- section (NTS) structures. The database and parameters used in this part are identical to the FRC part.

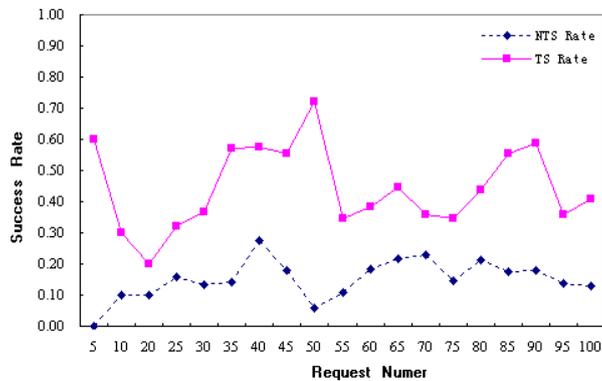


Fig. 13. Success rate comparison based on TS and NTS sets (15 DSC items)

Figure 13 shows that the success rate of matching requests with DSC items in the TS sets are much better than it is in the NTS sets. The average success rate of the TS sets is 2.75 times higher than NTS sets. This result provides a solid proof that TS based mechanism has superior performance for improving matching success rates.

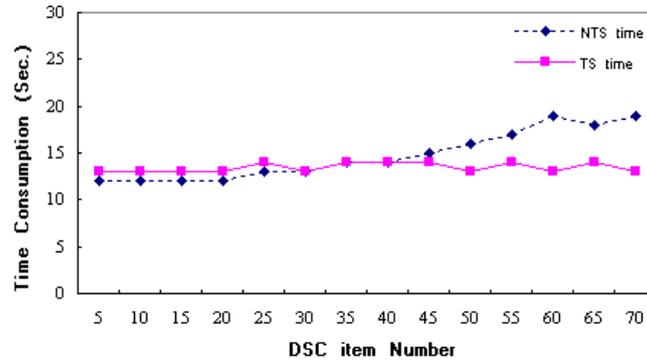


Fig. 14. Time consumption based on TS and NTS sets (100 requests).

To evaluate the time consumption factor of the TS method, the experimental program randomly generated 100 requests and modified the DSC item number to observe the results. Figure 14 indicates that the time consumption of matching requests with DSC items in the TS sets is steady. However, the time consumption increases gradually in the NTS sets when the DSC item number increases.

To further evaluate the results in Figure 14, we increased the requests number to 200. The results shown in Figure 15 are very similar to Figure 14, except that it takes more DSC items in this experiment to increase the time consumption in the NTS sets. Nevertheless, the time consumption of the NTS sets increased from 22 seconds to 34 seconds when DSC items number increased from 5 to 90. However, the TS sets almost remained unchanged.

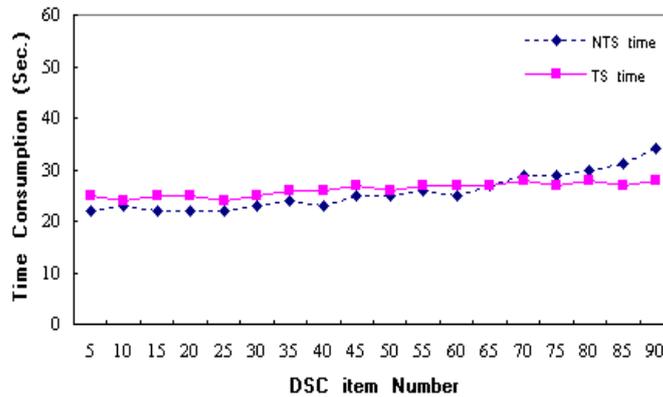


Fig. 15. Time consumption based on TS and NTS sets (200 requests).

6. Conclusion

The DSC-based agent design offers an efficient and cost-effective solution to enhance agent capability reuse and integration. It can be further implemented to various domain agents, such as Web wrapper agents [22], negotiation agents [23], decision support agents [24, 25], etc.

The DSC design mechanism adopts several traditional methods including the cache model, three-type relationship, Cartesian product, and Chebyshev's theorem. These traditional methods are endowed a new meaning when they are applied to the DSC design for agent-based systems. In particular, the novel DSC slot design structure can significantly improve the reusability of agent capabilities.

A DSC-based agent can update its capability through removing the dated DSC items and inserting new DSC items from the DSC warehouse [26]. Instead of destroying retired or dated components in many traditional systems, the DSC-based agent returns the retired or dated DSC items to the DSC warehouse. These returned DSC items can be reused when they can fulfil the users' requests.

The functionality redundancy calculation and two-section- based structure deployed in the DSC-based design are able to improve the success rate of matching DSC items with user requests. The experimental results show that both FRC and TS methods can improve the success rate of request matching. Particularly, the success rate is improved significantly in the TS-based experiments. The similarity scores in the FRC sets are much lower than they are in the NFRC sets. This reflects that the DSC item redundancy is reduced in the FRC-based experiments. The TS sets require less time consumption than NTS sets for matching DSC items, on the contrary, the FRC sets require more time than NFRC sets for matching.

In general, the experimental results based on the CAD models indicate that the TS and FRC methods used in the DSC-based systems improve the system performance in terms of matching requests and reusing DSC components. Hence, the DSC-based agent design offers a reusable and extensible solution, which demonstrates the superior system performance.

The current agent design pattern is based on the system prototype and agent simulation processes. To further improve the DSC-based design, we plan to develop a complete DSC-based multiagent system to solve some complex problems based on industrial cases, such as the automated feature recognition problem described in Section V. Thus, the DSC-based agent design can be further evaluated and improved systematically.

References

1. Lesser, V., Decker, K., Wagner, T., Carver, N., Garvey, A., Horling, B., Neiman, D., Podorozhny, R., Nagendra Prasad, M., Raja, A., Vincent, R., Xuan, P.,

- Zhang, X.Q.: Evolution of the GPGP/TAEMS Domain-Independent Coordination Framework. *Journal of AAMAS*, Vol. 9, 87-143. (2004)
2. Hutzschenreuter, A. K., Bosman, P., Blonk-Altena, I., Aarle, J. V., Poutre, H. L.: Agent-based Patient Admission Scheduling in Hospital. *Proc. of AAMAS*. 45 – 52. (2008)
 3. Wang, M., Wang, H., Xu, D., Wan, K.K., Vogel, D.: A Web-service Agent-based Decision Support System for Securities Exception Management. *Expert Systems with Applications*. Vol. 27, Elsevier Publication, 439 – 450. (2004)
 4. Brazier, F.M.T., Jonker, C.M., Treur, J.: Principles of Component- Based Design of Intelligent Agents. *Data & Knowledge Engineering*, Vol. 41, Issue 1, Elsevier Press, 1-27, (2002)
 5. Zhang, H.L., Leung, C. H. C., Raikundalia, G. K.: Topological analysis of AOCD-based agent networks and experimental results. *Journal of Computer and System Sciences*, Vol. 74, Elsevier Press, 255–278 (2008)
 6. Braubach, L. and Pokahr, A.: Representing Long-Term and Interest BDI Goals. *Proc. of ProMAS, AAMAS Foundation Press*. (2009)
 7. Mili, H., Mili, A., Yacoub, S. and Addy, E.: *Reuse-Based Software Engineering: Techniques, Organization, and Controls*. John Wiley & Sons Press. (2002)
 8. Ghijssen, M., Jansweijer, W., Wielinga, B.: Towards a framework for agent coordination and reorganization, *AgentCoRe*. *Proc. of COIN, IEEE Press*. (2007)
 9. Mohagheghi, P., Conradi, R.: An Empirical Investigation of Software Reuse Benefits in a Large Telecom Product. *ACM Transactions on Software Engineering and Methodology*, Vol. 17, 3. (2008)
 10. Gomez, M., Plaza, E.: Extending matchmaking to maximize capability reuse. *Proc. of AAMAS, New York*. (2004)
 11. Cheyer, A., Martin, D.: The Open Agent Architecture. *Journal of AAMAS*, Volume 4 Issue 1-2. (2001)
 12. Sycara, K., Widoff, S.: LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. *Journal of AAMAS*, 5, 173-203. (2002)
 13. Vitharana, P., Zahedi, F., Jain, H.: Design, retrieval, and assembly. *Communications of the ACM*, Vol. 46, No. 11, 97 – 102. (2003)
 14. Haigh, K., Phelps, J., Geib, C.: An Open Agent Architecture for Assisting Elder Independence, *Proc. of AAMAS*, pp. 578 – 586. (2002)
 15. Jennings, N. R.: An agent-based approach for building complex software systems. *Communications of the ACM*, Vol. 44, No. 4. (2001)
 16. Li, Y., Bandar, Z. A., McLean, D.: An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transaction on Knowledge and Data Engineering*, Vol.15, No.4, 871-882. (2003)
 17. Gangemi, A., Navigli, R., Velardi. P.: The OntoWordNet Project: Extension and Axiomatization of Conceptual Relations in WordNet. *Proc. of ODBASE*, pp. 820-838. (2003)
 18. Hein, J. L.: *Discrete Mathematics*, Jones and Bartlett Press, 74-75. (2002)
 19. Venables, W. N., Ripley, B. D.: *Modern Applied Statistics with S-PLUS*, Springer-Verlag Press. (1999)
 20. Hogg, R.V. and Craig, A.T.: *Introduction to Mathematical Statistics (5th Edition)*, Prentice Hall Press. (2007)
 21. Zhang, H. L., Van der Velden, C., Yu, X., Jones, T., Fieldhouse, I., Bil, C.: Developing A Rule Engine for Automated Feature Recognition from CAD Models. *Proc. of IEEE IECON*, 3925 – 3930. (2009)
 22. Chang, C., Siek, H., Lu, J., Hsu, C., Chiou, J.: Reconfigurable Web Wrapper Agents. *IEEE Intelligent Systems*, 34 – 40. (2003)

Hao Lan Zhang, Wenhua Zeng, and Christian Van der Velden

23. Fischer, K., Chaib-draa, B., Muller, J. P., Pischel, M., Gerber, C.: A Simulation Approach Based on Negotiation and Cooperation between Agents - A Case Study. *IEEE Transaction on Systems, Man and Cybernetics - Part C*, Vol. 29, Issue 4, 531-545. (1999)
24. Vahidov, R., Fazlollahi, B.: Pluralistic multi-agent decision support system: a framework and an empirical set. *Information & Management*, Vol. 41, Elsevier Press, 883 – 898. (2004)
25. Turban, E., Aronson, J. E., Liang, T.: *Decision Support Systems and Intelligent Systems* (7th edition), Prentice-Hall Press, 223,. (2005)
26. Zhang, H. L., Leung, C.H.C., Yu, X., He, J.: An Optimised Design for Agent Capability Reuse. *Proc. of IEEE/WIC/ACM WI-IAT*, Vol.3, 231-234. (2010)

Dr. Hao Lan Zhang is an Associate Professor and Deputy Director of IS discipline at NIT, Zhejiang University. Dr. Zhang received a PhD from Victoria University, Australia. He was with RMIT University as a Research Fellow from 2008–2010. Prior to that, he worked as a research assistant/tutor at Victoria University. His research interests include: multi-agent systems, knowledge-based systems, intelligent information systems, e-health systems, database, etc. He has published over 30 research papers in various journals, conferences and workshops. Dr. Zhang received an outstanding student award from the Chinese Scholarship Council and Chinese Ministry of Education in 2006.

Dr. Wenhua Zeng is a Professor and Vice Dean of Software School at Xiamen University. He obtained a PhD degree from Zhejiang University in 1989. He has been active in areas of Software Engineering, Grid Computing, Cloud Computing.

Dr. Christian Van der Velden is a Control System Engineer at BAE Systems Australia. He received his PhD in 2009 from RMIT University, Australia in the field of Knowledge Based Engineering. He worked as a Research Fellow at RMIT University from 2008 to 2010 working with GKN Aerospace and the Cooperative Research Centre for Advanced Automotive Technology (AutoCRC) in the area of Automated Feature Recognition.

Received: March 4, 2011; Accepted: May 4, 2011.

Quantitative Analysis for Symbolic Heap Bounds of CPS Software

Renjian Li¹, Ji Wang¹, Liqian Chen¹, Wanwei Liu², Dengping Wei²

¹ National Laboratory for Parallel and Distributed Processing
410073 Changsha, China

² School of Computer, National University of Defense Technology
410073 Changsha, China

{li.renjian@gmail.com, wj@nudt.edu.cn, lqchen@nudt.edu.cn,
wwliu@nudt.edu.cn, dpwei@nudt.edu.cn}

Abstract. One important quantitative property of CPS (Cyber-Physical Systems) software is its heap bound for which a precise analysis result needs to combine shape analysis and numeric reasoning. In this paper, we present a framework for statically finding symbolic heap bounds of CPS software. The basic idea is to separate numeric reasoning from shape analysis by first constructing an ASTG (Abstract State Transition Graph) and then extracting a pure numeric representation which can further be analyzed for the heap bounds. A quantitative shape analysis method based on symbolic execution is defined in the framework to generate the ASTG. The numeric representation is extracted based on program slicing technique and inputted into an abstract interpretation tool for computing the heap bounds. We take list manipulating programs as an example to explain how to instantiate the framework for important data structures and to exhibit its practicability. A novel list abstraction method is also presented to support the instantiation of the framework.

Keywords: CPS software, heap bounds, quantitative shape analysis, symbolic execution, program slicing.

1. Introduction

Conformance with quantitative constraints over temporal-spatial resources (such as execution time, energy, memory, etc.) is central to the correctness of CPS software. Compared with general purpose software, CPS software often suffers from very limited memory [1, 6, 8]. One of the most important quantitative properties of CPS software should be its heap bounds.

CPS software often adopts dynamic memory allocation schemes, where a program can at any time request the operating system to allocate additional memory from heap. The failure of dynamic memory allocation request may cause the failure of CPS software or even the whole CPS system. Usually

depending on environmental parameters and/or user inputs, symbolic heap bounds are extremely important for CPS software which tends to feature a tight coupling between physical and software components and runs in open environments. Besides, precise symbolic heap bounds could also be very useful for inter-procedural static analysis and hardware synthesis [2].

There are several obstacles for finding precise heap bounds of CPS software written in imperative languages like C. Firstly, loops and recursive procedures are what make heap usage exceed its bounds. However, finding loop bounds may be difficult even for numeric programs and harder when loop bounds depend on the shape of the heap. Secondly, both shape analysis and numeric computation are needed for finding heap bounds. However, a casual combination of these techniques should involve a large increase in complexity, both in terms of the verification problem and the implementation [3]. Last but not the least, programmers often adopt shared mutable data structures, such as trees and lists, to develop CPS software for the sake of effectiveness and convenience. However, none of the available heap bounds analysis techniques can handle these shared mutable data structures full automatically.

In this paper, we try to tackle these obstacles and present a novel framework for analyzing heap bounds of CPS software. The basic idea is to separate numeric reasoning from shape analysis and to make full use of existed static analysis techniques and tools for finding precise heap bounds. In detailed, the framework will first construct an ASTG via quantitative shape analysis based on symbolic execution [4]. The ASTG is employed as an intermediate representation during the analysis and the transformation. A numeric representation maintaining the heap usage properties of the original program is further extracted based on the main idea of program slicing. The abstract interpretation tool Interproc [24] is finally used to find the heap bounds.

The framework can be instantiated for various data structures manipulating programs. In order to explain how the framework should work, we take list manipulating programs as an example. A new list abstraction model which maintains both shape and quantitative properties is presented and used during instantiating the framework. The new list abstract model stores the relationship between variables and list nodes in a singly-linked list implicitly, and represents list states in a compact manner. Compared with other abstraction models for list, such as shape graph and separation logic, it enjoys lower space overhead and higher implementation efficiency.

This paper has several main technical contributions:

- We present a new framework for analyzing heap bounds of CPS software. It separates numeric reasoning from shape analysis by extracting a numeric representation which maintains the heap usage of the original program.
- We further show how the framework could be instantiated for important data structures taking list manipulating programs as an example. With proper modifications and extensions, the framework should also work for

programs containing more complex data structures such as circular lists, doubly-linked lists, etc.

- We present a novel quantitative shape analysis method based on symbolic execution. It generates an ASTG (Abstract State Transition Graph) and is more precise than classic shape analysis methods.

The paper is organized as follows. Section 2 presents the related work. Section 3 explains our main idea through a simple example. Section 4 presents the framework for analyzing heap bounds of CPS software. In Section 5 we introduce how to instantiate the framework for programs manipulating lists. Section 6 presents the experimental results. Section 7 makes a conclusion.

2. Related Work

Quantitative properties of CPS software have gained a lot of attention within the past several years, as shown by the recent publications on the subject [7-10]. But they mainly focus on the WCET problem, while we try to find heap bounds of CPS software in this paper.

Early work for heap bounds analysis and verification [11-13] mostly focuses on functional programs where data structures are basically immutable and easier to handle. These works often needn't treat shape or the shared mutable data structures.

For imperative Object Oriented programming languages such as Java, the method proposed in [14] relies on a type system and type annotations. It is therefore up to the programmer to annotate the sizes of data structures and the amount of heap memory required for each method. Hofmann et al. [15] also propose a type based heap space analysis for Java style OO programs with explicit deallocation. It uses an amortised analysis and calculates heap memory usage with an LP-solver based on function inputs during the type inference. Albert et al. introduce a Java memory-bounds tool in [16]. It uses a heap abstraction and applies heuristics based on arithmetic simplification to find a memory bound.

For assembly-level programs, Chin et al. [25] present a method to find memory resource bounds for each method in terms of the symbolic values of its parameters. However, the system does not handle shared objects.

Different from previous work [14-16, 25], we focus on the C language which is found in many critical CPS software implementations. Finding heap bounds for C programs needs both quantitative shape analysis and numeric reasoning. Previous work often omitted shape analysis; while our method uses a more precise shape abstraction, which is crucial for dealing with our examples.

He et al. [17] try to reuse a general-purpose verification system Hip/Sleek for memory usage verification, where shape, size and alias information can be readily obtained from the specifications given in separation logic. They can verify quite a number of programs that cannot be handled by previous

approaches, such as doubly linked lists, cyclic linked lists and binary trees. However, they need to supply memory specifications for the programs manually, while our framework could find heap bounds automatically.

Cook et al. [18] present a constraint-based method to find symbolic bounds for C programs combining several known methods and tools. They use the shape analysis tool THOR [20] to produce a new program without heap operations and use constraint-based techniques to find the heap bounds. Magill et al. [19] present a formal system for producing numeric abstractions of heap-manipulating programs based on the work of [18, 20, 21]. Our quantitative shape analysis procedure is based on symbolic execution techniques and that is different from THOR which is based on separation logic invariants generation. Another key difference between their method and ours should be the abstract model for list. Their work uses separation logic to model the abstract list state, while our work adopts the newly presented list abstract model. By focusing on specific data structures, our framework is able to obtain more precise results than their work while without have to ask the user to supply any annotations. Our numeric representation extraction algorithm is based on program slicing technique, which makes our result numeric CFG be smaller than theirs when applying to heap bounds analysis.

Shape graph is the most frequently used abstract model in static analysis; however, it can't express quantitative properties of heap. Some researchers [14, 17-20] used separation logic to describe the abstract state of list. Bouajjani et al. [22] use counter automata to model the abstract state of list. Our list abstract model has the equal expression ability with their counter automata. But our method enjoys lower space overhead and better scalability. Besides, the method in [22] is not implemented automatically; while we have implemented a prototype tool based on our list abstract model.

3. A Motivating Example

The example in figure 1 is taken from [2] with minor modifications, which may denote a frequently used programming pattern in CPS software. The procedure reads integers from an input signal i and returns every n inputted integers to an output signal o in inverse order. The primitive `input()` reads one integer from i , and the primitive `output()` writes one integer to o . The data structure `LIST` is used to represent singly-linked lists (with fields `data` and its `next` element). The bound of heap usage for `prio` should be $8n$ (assuming that `sizeof(LIST) = 8`).

This example is fairly simple but exhibits all the obstacles we want to overcome in this paper when finding precise symbolic heap bounds of CPS software. Using the method presented in this paper, we are able to find such a bound for this example. An intermediate representation including only numeric variables will be constructed and analyzed for the heap bounds in our framework. An equivalence program of the numeric representation written in C is given in figure 2 for understanding convenience. The numeric

representation may contain some variables from the original program (such as k and n) and some instrumentation variables such as *heap_now*, *heap_peak* (which track the heap usage) and x , y (which track the quantitative properties of shape, and in this case they represent the length of lists). Now analyzing the following numeric program, we could know the biggest value of *heap_peak* is $8n$, which is just the heap bound of the original program.

```

void prio(int n, in_signal i, out_signal o) {
LIST *head,*cur;
1: while(1){
// Build up an n-sized buffer
2:   head = (LIST*)malloc(sizeof(LIST));
3:   head->data = input(i);
4:   for(int k = 0;k<n-1;k++){
5:     cur = (LIST*)malloc(sizeof(LIST));
6:     cur->data = input(i);
7:     cur->next = head;
8:     head = cur;}
// Send the buffer to the output and deallocate it
9:   cur = head;
10:  while(cur != NULL) {
11:    output(o, cur->data);
12:    head = cur->next;
13:    free(cur);
14:    cur = head; }}}

```

Fig. 1. A motivating example

4. A Symbolic Heap Bounds Analysis Framework

In this section, we introduce a new framework for finding symbolic heap bounds statically. The framework is presented in figure 3. After getting the CFG (Control Flow Graph) of the original program, we go forward with a quantitative shape analysis which can generate shape invariants for each program point. We do not annotate the abstract states and transitions in the original CFG, but construct a new intermediate representation named as Abstract State Transition Graph (ASTG, for short). ASTG is a core internal representation in our framework which could be used to extract the numeric representation. The final numeric representation is actually a CFG which maintains the heap usage properties of the original programs and manipulates only numeric variables. A numeric reasoning tool such as Interproc [24] could be then used to find the heap bounds. We will introduce these steps in detail in the following subsections. In this paper, we take list manipulating programs as an example for explaining the main idea of the framework. When extending to programs manipulating other kinds of data structures, firstly, you need to adopt a suitable abstract model for these data

Renjian Li, Ji Wang, Liqian Chen, Wanwei Liu, Dengping Wei

structures, and then make some proper modifications when implementing the core algorithms.

```
void prio_numeric(int n, in_signal i, out_signal o)
1: {int heap_now, heap_peak, k, X, Y;
2:  heap_now = 0;
3:  heap_peak = 0;
4:  while(1){
5:    heap_now = heap_now + 8;
6:    if(heap_now > heap_peak)
7:      heap_peak = heap_now;
8:    k = 0;
9:    X = 1;
10:   while(1){
11:     if(k>=n-1)
12:       break;
13:     heap_now = heap_now + 8;
14:     if(heap_now > heap_peak)
15:       heap_peak = heap_now;
16:     k = k + 1;
17:     X = X + 1;}
18:   Y = X;
19:   while(1){
20:     if(Y==1)
21:       break;
22:     heap_now = heap_now - 8;
23:     Y = Y - 1;}
24:   heap_now = heap_now - 8; }}
```

Fig. 2. The numeric program tracking the heap bounds

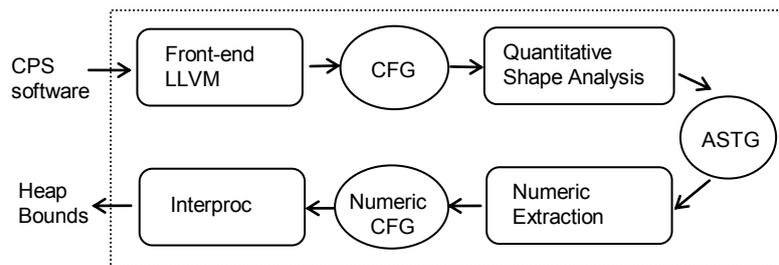


Fig. 3. Static analysis framework for symbolic heap bounds

4.1. Abstract State Transition Graph

ASTG plays an important role in our framework, so we first give its definition in this subsection.

Definition 1. An Abstract State Transition Graph (ASTG) is a 5-tuple $\langle Q, q_0, P, \rightarrow, L \rangle$, where:

- Q is a finite set of abstract states. Each $q \in Q$ is a 2-tuple $q = \langle sg, pc \rangle$ where sg is an abstract shape representation in program point pc .
- $q_0 \in Q$ is the starting state.
- $P \subseteq Q$ is the set of exit states.
- $\rightarrow \subseteq Q \times Q$ is the set of transitions.
- L is a labeling function which labels each $\tau \in \rightarrow$ with program commands.

The abstract shape representation must maintain both shape properties and quantitative properties of the current shape. Supposing sg can be further divided into shape part sg^s and quantitative part sg^q . Given an abstract shape representation sg , we record it with $sg = sg^s \oplus sg^q$. However, it's obvious that these two parts may rely on each other and are not fully independent with each other. How to express the abstract shape depends on the concrete implementation and the abstract model for shared mutable data structures.

One key difference between our method and existed methods (such as [21]) is that we classified the transitions. The transitions in \rightarrow could be classified into three disjoint subsets. \rightarrow_s stands for the kind of transitions which are labeled with statements from the original program; \rightarrow_c stands for the conditional transitions which are labeled with Boolean expressions; and \rightarrow_l stands for the kind of transitions which enter a loop structure and are labeled with a special command `MakeShapeSymbolic`. Any $\tau \in \rightarrow$ could be treated as a transfer function which maps a source abstract state to a target abstract state.

The transitions in \rightarrow_s are easily understood. Given an input state, it just generates one output state according to the semantics of the labeled program statements. It's worth noting that the definition of ASTG doesn't require the statements labeled on \rightarrow_s must be assignment statements, as you can see soon from the example ASTG in figure 4.

There are some cases that a statement could generate two output abstract states. One case is when the branch condition of a branch statement could either be true or false for an input abstract state. The other case is when some special assignments might also generate two abstract states, according to the operational semantics of the abstract shape model for the underline data structures. For these two cases, we must bring in conditional transitions which are labeled with transition conditions and add them to \rightarrow_c .

In order to handle loop structures, we bring in a special transition for each edge entering a loop structure in the CFG and add them to \rightarrow_l . The target abstract state of each transition in \rightarrow_l is a symbolic representation of the source abstract state. We label these transitions with a special command named `MakeShapeSymbolic`. It means that we should construct a new

symbolic abstract state. The quantitative part sg^q of the abstract shape representation should contain only new symbolic variables.

There are some optimizations or constraints we would like to make for the transitions in ASTG in order to reduce the abstract states set Q and to simplify the implementation of our framework. As for the transitions in \rightarrow_s , if a continuous fragment of statements can only generate one output abstract state, then they could be merged into a compound transition. The compound transition takes the source state of the first transition and the target state of the last transition and is labeled with the statements from all these transitions sequentially. There are some cases that the condition of a branch statement is definitely evaluated to *true* or to *false* for the input abstract state. We treat these branch statements as normal assignment transition in this case and label these branch statements with transitions in \rightarrow_s . The underline abstract modeling method must assure that an assignment statement should never generate more than two abstract states for the correctness of our method. Suppose the conditions labeled on the two outgoing transitions from one common source abstract states are $cond_true$ and $cond_false$, it must be assured that $cond_true = \neg(cond_false)$ and $cond_false = \neg(cond_true)$. As for the transitions in \rightarrow_i , the shape parts of the source state and the target state must be identical.

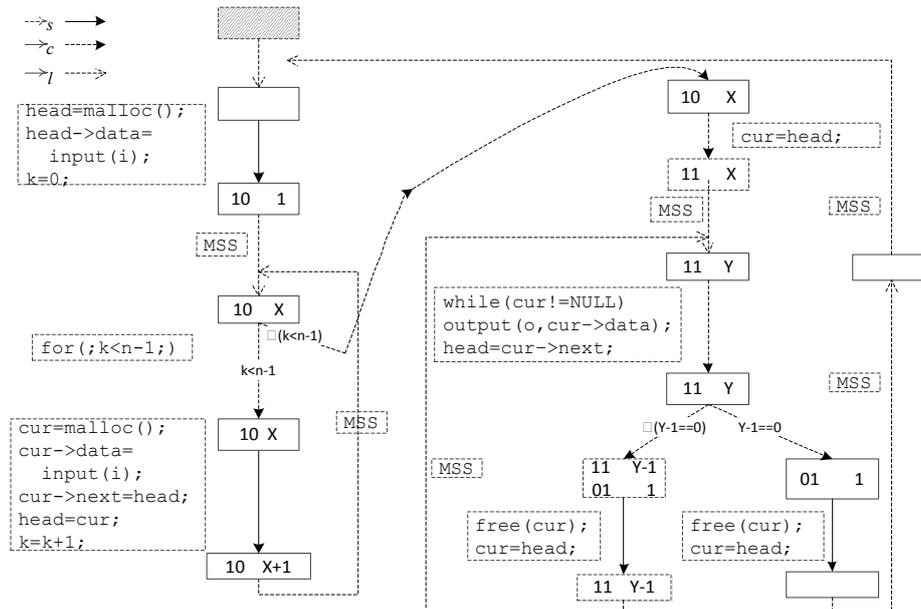


Fig. 4. The ASTG generated for the motivating example

The generated ASTG for the example by our framework in figure 1 is given in figure 4. Here each solid line box stands for an abstract shape sg and its position should exhibit the program counter pc . The abstract shape is

expressed with our list abstract model which will be explained in the next section. You don't have to doubt why an abstract shape can be expressed like that now. The starting state q is slash marked. The program will runs forever so there is no exit states. Three kinds of transitions are denoted with different arrows as shown in the figure. The dotted line boxes positioned aside present what are labeled for each transition. The operation `MakeShapeSymbolic` is represented with `MSS` for brevity. As you can see, the ASTG describes all the abstract states that may occur during the execution of the original program. A program point of the original CFG may be separated into several abstract states with different abstract shape parts. An ASTG could be treated as the result of refining the original CFG based the shape analysis result in some ways.

4.2. Quantitative Shape Analysis

In this part we introduce how we can construct an ASTG from a CFG based on the idea of symbolic execution [4]. The algorithm is presented in figure 5. Its main idea is to start symbolic execution from the initial state and record the abstract states and the transitions that can arise during symbolic execution. Semantics of all the basic shape operations must be defined at first in order to implement the algorithm. Before explain how the algorithm works, we first define the abstract subsumption relationship between two abstract states.

Definition 2. Given two abstract states s_1, s_2 , supposing $s_1 = \langle sg_1, pc_1 \rangle$ and $s_2 = \langle sg_2, pc_2 \rangle$, $sg_1 = sg_1^s \oplus sg_1^d$, $sg_2 = sg_2^s \oplus sg_2^d$. We would call s_1 is subsumed by s_2 and record with $s_1 \triangleleft s_2$ if and only if $pc_1 = pc_2 \wedge sg_1^s = sg_2^s$ and sg_2^d includes only atom symbolic variables.

The algorithm in figure 5 maintains two sets of abstract states, where *NEW* maintains the abstract states needed to be analyzed and *OLD* keeps the ones that have been analyzed. The algorithm will start symbolic execution from a selected abstract state in *NEW* and runs along the original CFG. When the set *NEW* is empty we could get the final ASTG. The method of selecting the next abstract state to analyze from *NEW* is not fixed and depends on the adopted search strategy. With a selected state from *NEW*, the algorithm will keep executing until it reaches one of the following three special cases:

- When reaching an exit abstract state, it will select another abstract state from *NEW*, and start a new symbolic execution process.
- When reaching a statement that may generate two possible abstract states, it will first construct a new \rightarrow_s transition and two new \rightarrow_c transitions. If any branched new abstract state is not subsumed by some abstract state in *OLD*, then a new abstract state has occurred and must be added to *NEW*.
- When reaching an edge which enters a loop structure in the CFG, it will check whether the state could be subsumed by some abstract state in *OLD*. If not, then a new abstract state has occurred and must be created with `MakeShapeSymbolic` command and added to *NEW*. Besides, a new \rightarrow_l

transition and a new \rightarrow_s transition may also be constructed accordingly. The `MakeShapeSymbolic` operation means making a shape representation become a more general symbolic representation. Its concrete implementation depends on the underline abstract model for shared mutable data structures.

As an example, you can refer to figure 4 which gives an ASTG generated by the algorithm for the motivating example in figure 1.

Algorithm 1: QuantitativeShapeAnalysis

INPUT: q_0 // the initial abstract state
 P // the set of exit abstract states
 cfg // the CFG of the original program

OUTPUT: $astg = \langle OLD, q_0, P, \rightarrow_s \cup \rightarrow_c \cup \rightarrow_l, L \rangle$
// the ASTG of the original CFG

begin

- 1: $OLD = \emptyset; NEW = \{q\};$
- 2: **while**($NEW \neq \emptyset$) **do**
- 3: select and remove s from NEW , add it to OLD ;
- 4: start symbolic execution from s until the following cases happen:
 // suppose the temporal abstract state before the interrupt is s'
- 5: In case of reaching a statement that may generate two different abstract states s_1, s_2 :
- 6: **if** $s \neq s'$ **then**
- 7: add $\langle s, s' \rangle$ to \rightarrow_s , label it with corresponding statements;
- 8: add s_1 to NEW if $\forall s_i \in OLD. \neg(s_1 \triangleleft s_i)$;
- 9: add s_2 to NEW if $\forall s_i \in OLD. \neg(s_2 \triangleleft s_i)$;
- 10: add $\langle s', s_1 \rangle, \langle s', s_2 \rangle$ to \rightarrow_c , label it with corresponding conditions;
- 11: **continue**;
- 12: In case of reaching an edge entering a loop structure in the CFG:
- 13: **if** $s \neq s'$ **then**
- 14: add $\langle s, s' \rangle$ to \rightarrow_s , label it with corresponding statements;
- 15: **if** $\forall s_i \in OLD. \neg(s' \triangleleft s_i)$ **then**
- 16: $s'' = \text{MakeShapeSymbolic}(s')$;
- 17: add s'' to NEW ;
- 18: add $\langle s', s'' \rangle$ to \rightarrow_l , label it with `MakeShapeSymbolic`;
- 19: **else** // suppose $\exists s_i \in OLD. s' \triangleleft s_i$
- 20: add $\langle s', s_i \rangle$ to \rightarrow_l , label it with `MakeShapeSymbolic`;
- 21: **continue**;
- 22: In case of $s' \in P$
- 23: **continue**;
- 24: **od**

end

Fig. 5. The QuantitativeShapeAnalysis algorithm

4.3. Numeric Extraction

In this subsection, we will first introduce how we can model the heap usage of the original program with two numeric variables, and then introduce the main steps for extracting a numeric CFG from the ASTG.

Heap bound is a quantitative property intending to find a peak value for heap usage. Here we bring in two instrumentation variables *heap_now* and *heap_peak* which represent the heap usage at present and the peak heap usage until now respectively. There are two cases when we need to modify these two variables.

When programs call library functions such as `malloc()` to allocate some amount of memory from heap, *heap_now* should be increased by the amount of allocated heap memory. Besides, we have to determine whether *heap_now* is greater than *heap_peak*, and update *heap_peak* if it was. For each original statement `ptr=malloc(malloc_size)`, we should instrument with the following statements:

```
heap_now = heap_now + malloc_size;
if(heap_now>heap_peak)
    heap_peak = heap_now;
```

Other library functions such as `realloc()` and `calloc()` could also be handled in this way with respect to their operation semantics. We will not list them in detail.

When programs call library function `free()` to give back some amount of memory to heap, *heap_now* should be decreased by the amount of deallocated heap memory. Suppose the size of the freed memory *free_size* has been gained by a pre-analysis task, we will instrument `free(ptr)` with the following statements:

```
heap_now = heap_now - free_size;
```

We can traverse all statements labeled on the transitions of ASTG and complete the instrumentation work based on syntax analysis. The biggest value of *heap_peak* should be the heap bounds of the original program. However, besides depending on numeric program variables, *heap_peak* may also be controlled by loops and branches which may further depend on the shape of the heap, as we can see from the example in figure 1. Existed numeric reasoning tools could not be adopted directly. We will try to overcome these obstacles by constructing a pure numeric representation of the original program. The good news is that ASTG contains plentiful information for transforming these syntax structures into corresponding numeric versions.

We can transform these loops depending on the shape of the heap as following. Because we have refined the original loop structures in the quantitative shape analysis phase, all new loop structures in ASTG enjoy the good character that the shape parts of the abstract shape representations in the loop entries are identical. So the loop body can only affect the

quantitative properties of the abstract states. We try to bring in new instrumentation variables to describe the change of the quantitative parts of the abstract representations. Each transition in \rightarrow_l is also well designed requiring that the shape parts of the source abstract state and the target abstract state must be identical. Besides, the quantitative part of the target abstract state of a \rightarrow_l transition contains only atom symbolic variables. We could take these atom symbolic variables as new numeric instrumentation variables, and assign the corresponding symbolic expressions from the quantitative part of the source abstract state of \rightarrow_l to them. These assignment statements could then reflect the effect of the loop body for the abstract state.

As for the branches depending on the shape of the heap, we can replace the shape related branch conditions with equivalent quantitative properties of the shape. It's fortunately that the generated ASTG has already transformed these branch conditions into numeric versions, as you can see in figure 4. We will explain how it is possible for us to make the transformation taking lists manipulating programs as an example in the next section.

Now we can extract the statements that affect the value of *heap_peak* and construct the numeric CFG. The extraction algorithm presented in figure 6 is based on the program slicing technique [5]. Program slicing can be used to extract program statements which are relevant to a particular computation. A program *slice* is an executable program whose behavior must be identical to a specific subset of the original program's behavior. The principle of getting this behavior subset is called *slicing criterion* and can be expressed as the value of some sets of variables at some set of statements and/or program points.

The numeric CFG is a heap bounds slice of the instrumented ASTG with the initial slicing criterion including all the statements that modify *heap_peak*. The slicing procedure then starts to find and label the statements on all these edges that lead the program reaching some slice criterion based on the main idea of classic program slicing. After getting the labeled ASTG, we could construct the numeric CFG easily.

Algorithm 2: ExtractNumericCFG

INPUT: *astg* // the intermediate representation
OUTPUT: *cfg* // the final numeric representation tracking the heap bounds of the original program

begin

- 1: traverse *astg* and instrument it with *heap_now*, *heap_peak*;
- 2: traverse *astg* and label all transitions in \rightarrow_l with corresponding assignment statements;
- 3: add all the statements that modify *heap_peak* into slicing criterion;
- 4: slice the ASTG and construct *cfg*;

end

Fig. 6. The ExtractNumericCFG algorithm

As an example, the extracted numeric CFG for the motivating example by our framework is presented in figure 7. Each box may stand for a combination of several basic blocks. We present it like this on purpose to reflect the main idea of the ExtractNumericCFG algorithm and for simplicity. Suppose the initial value for *heap_now* and *heap_peak* are all zero, then we could get the heap bounds $8n$ with the abstract interpretation tool Interproc now.

5. Instantiate the Framework

In this section, we illustrate how to instantiate the framework for various shared mutable data structures. List is one of the most frequently used data structures in CPS software. So we will take list as an example and present a novel abstract model for lists in the first subsection. In the second subsection we will explain some special issues needed to be considered when instantiating the framework.

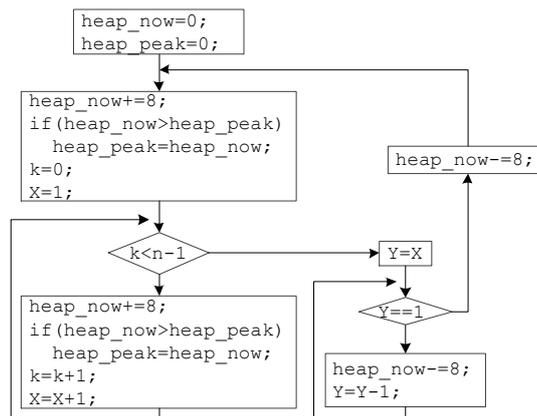


Fig. 7. The final numeric CFG for the motivating example

5.1. A New List Abstract Model

In order to express our basic idea more clearly, we focus on non-circled singly-linked lists at first. Although doubly-linked lists and circled lists are special, they can all be expressed using this abstract model with simple extensions. A singly-linked list node contains one *next* field pointing to the next list node; while all other fields can be treated as data fields. The abstract syntax considered in this paper is given in figure 8. Here *PVar* is a finite set of

pointer variables of list type and $DVar$ is a finite set of variables of primitive types (for simplicity, we only consider integer variables by now). Allowed syntax structures include assignment statements, branch statements, and loop statements. We suppose that there is at most one *next* operator in a list operation. All other cases could be transformed by bringing in temp variables. One example manipulating lists is presented in figure 1.

$$\begin{aligned}
 p, q \in PVar & \quad x \in DVar \\
 e & := x \mid p \rightarrow data \mid e_1 + e_2 \mid e_1 - e_2 \\
 AsgnStmnt & := x = e \mid p = null \mid p = q \mid p = q \rightarrow next \mid p \rightarrow next = q \mid \\
 & \quad p \rightarrow next = null \mid p = malloc() \mid free(p) \mid p \rightarrow data = e \\
 Cond & := p == q \mid p == null \mid e_1 > e_2 \mid e_1 < e_2 \mid true \mid false \\
 & \quad \neg Cond \mid Cond_1 \vee Cond_2 \mid Cond_1 \wedge Cond_2 \\
 BranchStmnt & := \mathbf{if} \textit{Cond} \mathbf{then} \{Stmnt;*\} [\mathbf{else} \{Stmnt;*\}] \mathbf{fi} \\
 WhileStmnt & := \mathbf{while} \textit{Cond} \mathbf{do} \{Stmnt;*\} \mathbf{od} \\
 Stmnt & := AsgnStmnt \mid BranchStmnt \mid WhileStmnt \\
 Program & := \{Stmnt;*\}
 \end{aligned}$$

Fig. 8. The abstract syntax for operating lists

Suppose the set of list nodes is N , and the variables in $PVar$ form a special subset of list nodes $V \subseteq N$. Another special node $NULL \in N$ is used to denote the *null* node. We define a binary relation E from $N - \{NULL\}$ to $N - V$: $\forall n_1, n_2 \in N, \langle n_1, n_2 \rangle \in E$ iff n_1 points to list node n_2 when $n_1 \in V$ and n_2 is the *next* list node of n_1 otherwise. We record the transitive, irreflexive closure of E with E^+ , and define a binary predicate $Reach(n_1, n_2)$ such that $\forall n_1, n_2 \in N, Reach\langle n_1, n_2 \rangle$ evaluates to true iff $\langle n_1, n_2 \rangle \in E^+$.

For the time being, we consider programs without recursion or concurrency constructs, and therefore all variables could be assumed to be global. We arrange all the variables in $PVar$ in order, and for each $0 \leq i \leq |V| - 1$, V_i stands for the i th variable. The binary predicate $Reach$ describes the reachability property between list nodes in N . If $Reach(V_i, n)$ evaluates to true, then V_i could access list node n via a number of *next* operators and we would say that the variable V_i can *reach* list node n . For each list node $n \in N - V$, its reachability property for all the variables could always be expressed with a Boolean vector.

Definition 3. For each list node $n \in N - V - \{NULL\}$, its Variable Reachability Vector (VRV for short) γ_n is a $|V|$ -sized Boolean vector $\gamma_n \in \{0, 1\}^{|V|}$ where $\gamma_n[i] = 1$ iff $Reach(V_i, n)$ evaluates to true.

Let's see an example of VRV. Suppose the example in figure 1 has just executed the statement in line 7 during the $(n-1)$ th loop. The current list state may be like what is presented in the left part of figure 9. We denote list nodes with boxes, while denote those special nodes with boxes with dotted lines. If we define $V_0 = head$, $V_1 = cur$, then the VRVs listed on the top of each list node could describe their reachability. And as we can see, the list node n_n has just been created and pointed to by *cur*, so the VRV for n_n should be 01. The VRV

for all the other list nodes should be 11 because they can all be reached by both *head* and *cur*.

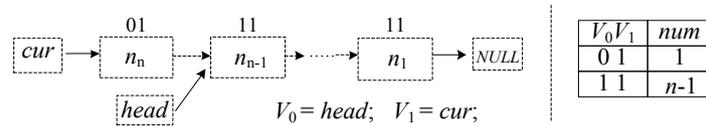


Fig. 9. An example for VRV and VRVSC

Without confusing, we would also say that the variable V_i can reach VRV γ if $\gamma[i]=1$. Given a VRV γ , we define a set of integers $R_\gamma = \{i | \gamma[i]=1\}$ which describes the variables that could reach γ . For two VRVs γ_1, γ_2 , if $R_{\gamma_1} \subset R_{\gamma_2}$, then we would say γ_1 can reach γ_2 and record with $\gamma_1 \prec \gamma_2$. Besides, for each variable V_i , we define a set of VRVs $\Gamma_i = \{\gamma | \gamma[i]=1\}$ which contains all the VRVs that the variable V_i can reach. After defining the reachability relationships between VRVs, for each variable V_i , we can find the minimal element in Γ_i and record it with $\gamma_i^0 : \nexists \gamma \in \Gamma_i, \gamma \prec \gamma_i^0$. It's obvious that γ_i^0 must be the right VRV for the list node pointed to by V_i .

Let's see the example in figure 9 again, where $\Gamma_0 = \{11\}, \gamma_0^0 = 11$, so we can know that the list node pointed to by *head* has the VRV 11. Similarly, because $\Gamma_1 = \{01, 11\}, \gamma_1^0 = 01$, we can know that the list node pointed to by *cur* has the VRV 01.

Given a list state, we can always construct a set of VRVs according to Definition 3. The relationship between these VRVs can describe the relative position of the corresponding list nodes. We can also get the VRV to which each variable points. There may exist any number of nodes with identical VRVs. Because we are only interested in the shape of heap and its quantitative properties, we can simply count the number of list nodes with identical VRV as following.

Definition 4. VRV Set with Counters (VRVSC for short) is a set of 2-tuples $VRVSC = \{\langle vrv, num \rangle\}$, where *vrv* is a VRV and *num* is the number of list nodes whose VRV equals to *vrv*.

According to the definition, all tuples in a VRVSC should be different in their *vrv*s and the *num* field for each tuple should always be greater than zero. We could always get one and only one VRVSC for each list state after defining the sequence of the variables. So VRVSC can be used as an abstract list model which maintains both the shape and quantitative properties.

For example, the VRVSC given on the right part of figure 9 could deliver the same information as the shape graph given on the left part. We could read from the VRVSC representation that there are $n-1$ node which could be accessed by both *head* and *cur* via a number of *next* operators, and there is

one list node pointed to by *head* but not pointed to by *cur*. Also we can know the *next* node of the node pointed to by *cur* should be the same node pointed to by *head*.

The operation of data fields and primitive data types are the same as normal, so we only consider the abstract semantics for list operations. For simplicity, we use some simple recording symbols. Given a set of integers S and a VRVSC tuple $\langle vrv, num \rangle$, we use $vrv/s_{\leftarrow 0}$ to represent the operation of replacing all the bits of vrv in S with 0, meanwhile, use $vrv/s_{\leftarrow j}$ to represent the operation of replacing all the bits of vrv in S with $vrv[j]$. $new(S)$ means creating a new VRV in which all the bits in S are 1 and the other bits are 0. We also use p to represent the p th variable when not confusing.

An assignment statement can be treated as a transfer function for the abstract list model. Given a VRVSC $vrpsc$, the abstract semantics given in figure 9 describe how it can be updated according to the semantics of each assignment statement. For each assignment statement, a tuple $\langle vrv, num \rangle \in vrpsc$ may be changed only when $vrv \in \Gamma_p \cup \Gamma_q$ (or Γ_p if the assignment statement doesn't manipulate q). We will use $vrpsc_{\Delta}$ to represent the unmodified part of $vrpsc$ in figure 10.

$$\begin{aligned}
\llbracket p = null \rrbracket (vrpsc) &= \{ \langle vrv', num' \rangle \mid \forall \langle vrv, num \rangle \in vrpsc \wedge vrv \in \Gamma_p. \\
&\quad vrv' = vrv / \{p\}_{\leftarrow 0}, num' = num \} \cup vrpsc_{\Delta} \\
\llbracket p = q \rrbracket (vrpsc) &= \{ \langle vrv', num' \rangle \mid \forall \langle vrv, num \rangle \in vrpsc \wedge vrv \in \Gamma_p \cup \Gamma_q. \\
&\quad vrv' = vrv / \{p\}_{\leftarrow q}, num' = num \} \cup vrpsc_{\Delta} \\
\llbracket p = q \rightarrow next \rrbracket (vrpsc) &= \{ \langle vrv', num' \rangle \mid \forall \langle vrv, num \rangle \in vrpsc \wedge vrv \in \Gamma_p \cup \Gamma_q. \\
&\quad vrv' = vrv / \{p\}_{\leftarrow q}, num' = (vrv == \gamma_q^0) ? num - 1 : num \} \\
&\quad \cup \{ \langle \gamma_q^0 / \{p\}_{\leftarrow 0}, 1 \rangle \} \cup vrpsc_{\Delta} \\
\llbracket p \rightarrow next = null \rrbracket (vrpsc) &= \{ \langle vrv', num' \rangle \mid \forall \langle vrv, num \rangle \in vrpsc \wedge vrv \in \Gamma_p. \\
&\quad vrv' = vrv / \{R_{\gamma_p^0}\}_{\leftarrow 0}, num' = (vrv == \gamma_p^0) ? num - 1 : num \} \\
&\quad \cup \{ \langle \gamma_p^0, 1 \rangle \} \cup vrpsc_{\Delta} \\
\llbracket p \rightarrow next = q \rrbracket (vrpsc) &= \{ \langle vrv', num' \rangle \mid \forall \langle vrv, num \rangle \in vrpsc \wedge vrv \in \Gamma_p \cup \Gamma_q. \\
&\quad vrv' = vrv / \{R_{\gamma_p^0}\}_{\leftarrow q}, num' = (vrv == \gamma_p^0) ? num - 1 : num \} \\
&\quad \cup \{ \langle \gamma_p^0, 1 \rangle \} \cup vrpsc_{\Delta} \\
\llbracket p = malloc() \rrbracket (vrpsc) &= \{ \langle vrv', num' \rangle \mid \forall \langle vrv, num \rangle \in vrpsc \wedge vrv \in \Gamma_p. \\
&\quad vrv' = vrv / \{i\}_{\leftarrow 0}, num' = num \} \cup \{ \langle new(\{p\}), 1 \rangle \} \cup vrpsc_{\Delta} \\
\llbracket free(p) \rrbracket (vrpsc) &= \{ \langle vrv', num' \rangle \mid \forall \langle vrv, num \rangle \in vrpsc \wedge vrv \in \Gamma_p. \\
&\quad vrv' = vrv / \{R_{\gamma_p^0}\}_{\leftarrow 0}, num' = (vrv == \gamma_p^0) ? num - 1 : num \} \cup vrpsc_{\Delta}
\end{aligned}$$

Fig. 10. Abstract semantics for list operations

When executing these list operations according to the abstract semantics, three cases in the output VRVSC may occur.

(1) There exist some tuples whose *num* fields are zero. The definition of VRVSC requires that all *num* fields must be greater than zero. If this case happens, we should delete these tuples from $vrpsc$.

(2) There exist several tuples whose *vrv* fields are identical. If this case happens, we should merge all these tuples into one tuple and take the sum of their *num* fields as the new *num* field.

(3) There exist some tuple whose *vrv* field is an all zero VRV. If the tuple's *num* field is also zero, then we simply delete the tuple from *vrvsc* according to the first case handling schema. However, if the *num* field is not zero, then it means some list nodes will never be accessed by any variables, so we will report a memory leak error.

We have defined a function $Compact(vrvsc)$ to handle the above cases. The function should be called after each list operation by default.

The abstract semantics for list operations are fairly straight forward, so we won't explain in detail here, and only list them in figure 10. The detailed explanations and examples are given in Appendix A.

5.2. Some Special Issues to Be Considered

Our framework requires that the abstract model should maintain both shape properties and quantitative properties for the shared mutable data structures. VRVSC could be used as an abstract model of list meeting the above requirements. Next we show some special issues to be considered when instantiating the framework.

(1) Algorithm for checking subsumption

The quantitative shape analysis algorithm will check the subsumption relationship between two abstract states in a high frequency. So the algorithm plays an important role for improving the efficiency and extendibility of the framework. Based on Definition 2, implementing such an algorithm for the abstract list model is fairly easy.

In this case, a *vrvsc* plays the role of the abstract shape representation sg , the set of VRVs: $\{vrv | \langle vrv, num \rangle \in vrvsc\}$ plays the role of the shape part sg^s , and the constraints on the *num* fields play the role of the quantitative part sg^q . In order to check whether $sg^s_1 = sg^s_2$, we could iterate on the two set of VRVs, and check if they are identical. We design one practicable algorithm and present it in figure 11. The comparison of two Boolean vectors (checking whether $vrv_1 = vrv_2$) could be implemented in a high efficiency way, so the checking subsumption algorithm could run efficiently.

(2) Implementation of the MakeShapeSymbolic Command

The MakeShapeSymbolic command constructs a more general symbolic representation. Based on the list abstract model, we can bring in a new symbolic variable for each *num* field of all the tuples in the VRVSC in the QuantitativeShapeAnalysis algorithm. When handling transitions in \rightarrow_l in the ExtractNumericCFG algorithm, we could compare two VRVSCs of the source and the target abstract states, find two tuples with identical *vrv*, and assign the symbolic expressions kept in the *num* field of the source state to the symbolic variable kept in the *num* field of the target state.

Algorithm 3: CheckingSubsumption

INPUT: s_1, s_2 // two abstract states, supposing:

$$s_1 = \langle \{ \langle vr_1^1, num_1^1 \rangle, \dots, \langle vr_1^i, num_1^i \rangle \}, pc_1 \rangle,$$

$$s_2 = \langle \{ \langle vr_2^1, num_2^1 \rangle, \dots, \langle vr_2^j, num_2^j \rangle \}, pc_2 \rangle$$

OUTPUT: *yes* // when $s_1 \triangleleft s_2$
no // otherwise

begin1: return *no* if $pc_1 \neq pc_2$;2: return *no* if $i \neq j$;3: For each $1 \leq k \leq i$ 4: subsumed = *no*;5: For each $1 \leq t \leq j$ 6: if $vr_1^k == vr_2^t$ and num_2^t is an atom symbolic variable7: subsumed = *yes*;

8: break;

9: if(subsumed == *no*)10: return *no*;11: return *yes*;**end**

Fig. 11. Algorithm for checking subsumption

(3) How to transform the shape dependent conditions to numeric conditions

In order to facilitate the ExtractNumericCFG algorithm, the abstract list model should be able to transform the shape dependent branch conditions into numeric versions. We focus on two kinds of branch conditions depending on the shape of the heap. They can both be transformed easily as following.

The first kind of conditions check whether a pointer variable is *null*. For example, $p == null$ means that all the list nodes should not be *reached* by the variable p . Given vr_vsc , it equals to $\forall \langle vr, num \rangle \in vr_vsc. vr[p] == 0$. Considering the num fields may contain symbolic variables, we adopt another equal expression: $\forall \langle vr, num \rangle \in vr_vsc. vr[p] \neq 0 \Rightarrow num == 0$. Another kind of conditions check whether two pointer variables point to the same list node. For example, $p == q$ means that p and q should always point to the same list node. Following the above idea, we can express it with $\forall \langle vr, num \rangle \in vr_vsc. vr[p] \neq vr[q] \Rightarrow num == 0$.

(4) How could an assignment statement become a branch statement?

A shape controlled branch may also affect the heap usage, for example we may call `malloc()` or `free()` in a shape controlled branch statement. However, when generating the ASTG, all these shape related branch conditions will be evaluated to a fixed value. So we don't have to handle these branches specially. That's because our framework has transferred the uncertainty of shape controlled branches to the uncertainty of some special assignment statements as following.

As we have pointed out in the previous sections, the *Compact()* operation must check if the *num* field of a tuple in VRVSC is zero. Because we adapt symbolic execution techniques during quantitative shape analysis, the underline SMT solver may report an answer *unknown* when verifying whether a symbolic expression equal 0. In this case, the *Compact()* operation may don't know whether to delete a tuple from the VRVSC. That should generate two abstract states and both of them should be treated with conservative care. In this case, the assignment statement acts like a branch statement and we should construct two transitions of \rightarrow_c type to deal with this problem. Corresponding conditions with only numeric symbolic expressions are also labeled on the transitions. For example, when executing the statement `head = cur->next` in line 12 of the example presented in figure 1, we must check if the symbolic expression $Y-1$ equals to 0, and the adopted SMT solver will answer with *unknown*, so we add two transitions to the ASTG as shown in figure 4.

6. Experimental Results

In order to prove the practicability of our framework, we have designed and implemented a prototype tool for analyzing symbolic heap bounds of list manipulating programs statically. The prototype tool is implemented on top of the LLVM framework [27] which offers many useful facilities for the front-end analysis and the implementation of the numeric extraction algorithm. We adapt the core framework of KLEE [23] to implement the quantitative shape analysis procedure. The final numeric representation is inputted into Inerproc [24] for computing the biggest value of *heap_peak*.

We have carried our experiments for several small programs. The example given in figure1 and `copy_and_delete` are hand written. `Hash_New_Table1` and `Hash_New_Table2` are two Hash Table construction functions taken from the `hash.c` of `heapplayer-0.1-benchmarks` [26]. The other benchmarks are taken from [28] and can be downloaded from <http://www.liafa.jussieu.fr/celia/examples.html>. Table 1 shows the statistics obtained for each analyzed program. The program size is evaluated in terms of number of lines of C code (Column 2). For each program, Column 3 represents the time for the preparation of CFG with LLVM infrastructure, Column 4 represents the time taken by quantitative shape analysis, Column 5 represents the time taken by numeric representation extraction, and Column 6 represents the time taken by Interproc to compute the biggest value for *heap_peak*. We also list the symbolic heap bounds reported by our tool and the expected results in the last two columns. Our experiments were done under Fedora 12 platform on Dual Core 1.8 GHz with 1GB main memory.

Table 1. Experimental results

Programs	Size (in lines)	Control Flow Analysis (in sec.)	Quantitative Shape Analysis (in sec.)	Numeric Extraction (in sec.)	Interproc (in sec.)	Report Result (in Bytes)	Expected Result (in Bytes)
figure 1	20	0.015	0.957	0.585	0.092	$8n$	$8n$
copy_and_delete	26	0.007	0.736	0.523	0.070	$8n$	$8n$
Hash_New_Table ₁	18	0.009	0.688	0.424	0.290	65545328	65545328
Hash_New_Table ₂	25	0.011	0.791	0.459	0.397	81931660	81931660
intlist-lib-add	16	0.009	0.090	0.045	0.009	8	8
intlist-lib-add_tail	30	0.010	0.745	0.583	0.075	8	8
intlist-lib-init	16	0.009	0.701	0.421	0.289	$8len$	$8len$
intlist-fold-copyGe5	37	0.009	0.698	0.601	0.081	$8n$	$8n$
intlist-fold-splitV	42	0.012	0.684	0.583	0.087	$8n$	$8n$
intlist-fold2-concat	59	0.016	0.959	0.601	0.094	$8(n+m)$	$8(n+m)$
intlist-fold2-merge	90	0.019	1.219	0.893	0.138	$8(n+m)$	$8(n+m)$

Our tool reports precise heap bounds for all the programs. Although the original programs and their ASTGs vary very much, the final numeric CFGs are all very simple, so the time for running Interproc is almost the same. Another interesting thing found during the experiments with Hash_New_Table() is that a first slicing before the shape analysis phase may be helpful sometimes. As our tool doesn't handle arrays of pointers now, it can't analyze Hash_New_Table() at first. The reason is that there exists an assignment statement for an array of pointers in the example. A first slicing can remove these assignment statements because they don't affect the heap usage. The initial experimental results have shown that, the framework presented in the paper is practicable and the list abstraction model is effective.

7. Conclusion and Future Work

We have presented a framework for statically analyzing symbolic heap bounds of CPS software. When input CPS software, the framework will generate a numeric representation which tracks the heap usage of the original program and can further be inputted into Interproc for the heap bounds. We have taken list as an example to explain how the framework could be instantiated for shared mutable data structures. We have also presented a novel list abstraction method which maintains precise shape properties and quantitative properties. We have built a prototype tool which could analyze the heap bounds full-automatically.

As for the future work, we will first carry experiments with some more complex examples and then try to extend the framework for handling other critical data structures that may also be frequently used in CPS software such as doubly-linked lists, tree, etc.

Acknowledgments. This research is supported by the National High Technology Research and Development Program (863 Program) of China under the Grant No. 2011AA010106, and the National Natural Science Foundation of China (NSFC) under the Grant Nos. 61120106006, 90818024 and 60725206.

References

1. Xia, F., Sun, Y., Tian, Y., Tadé, M. O., Dong, J.: Fuzzy Feedback Scheduling of Resource-Constrained Embedded Control Systems. *Int. J. of Innovative Computing, Information and Control*, vol.5, no.2, 311-321. (2009)
2. Cook, B., Gupta, A., Magill, S., Rybalchenko, A., Simsa, J., Singh, S., Vafeiadis, V.: Finding heap-bounds for hardware synthesis. *FMCAD2009*, 205-212. (2009)
3. Magill, S., Tsai, M., Lee, P., Tsay, Y.: Automatic numeric abstractions for heap-manipulating programs. *POPL 2010*, 211-222. (2010)
4. King, J. C.: Symbolic Execution and Program Testing. *Commun. ACM* 19(7), 385-394. (1976).
5. Weiser, M.: Program Slicing. *IEEE Trans. Software Eng.* 10(4), 352-357. (1984).
6. Xia, F. Sun, Y.: *Control and Scheduling Codesign: Flexible Resource Management in Real-Time Control Systems*. Springer, 2008, 272 pages, ISBN: 978-3-540-78254-4. (2008)
7. Seshia, S. A., and Rakhlin, A.: Quantitative Analysis of Systems Using Game-Theoretic Learning. *ACM Transactions on Embedded Computing Systems (TECS)*. To appear.
8. Lee, E. A., Seshia, S. A., *Introduction to Embedded Systems, A Cyber-Physical Systems Approach*, <http://LeeSeshia.org>, ISBN 978-0-557-70857-4. (2011)
9. Seshia, S. A., Kotker, J.: *GameTime: A Toolkit for Timing Analysis of Software*. *TACAS 2011*, 388-392. (2011)
10. Seshia, S. A.: *Quantitative Analysis of Software: Challenges and Recent Advances*. In *7th International Workshop on Formal Aspects of Component Software* (2010).
11. Hofmann, M., Jost, S.: Static prediction of heap space usage for first-order functional programs. *POPL 2003*, 185-197. (2003)
12. Hughes, J., Pareto, L.: Recursion and Dynamic Data-structures in Bounded Space: Towards Embedded ML Programming. *ICFP 1999*, 70-81. (1999)
13. Unnikrishnan, L., Stoller, S. D.: Parametric heap usage analysis for functional programs. *ISMM 2009*, 139-148. (2009)
14. Chin, W., Nguyen, H., Qin, S., Rinard, M. C.: Memory Usage Verification for OO Programs. *SAS 2005*, 70-86. (2005)
15. Hofmann, M., Jost, S.: Type-Based Amortised Heap-Space Analysis. *ESOP 2006*, 22-37. (2006)
16. Albert, E., Genaim, S., Gómez-Zamalloa, M.: Heap space analysis for java bytecode. *ISMM 2007*, 105-116. (2007)
17. He, G., Qin, S., Luo, C., Chin, W.: Memory Usage Verification Using Hip/Sleek. *ATVA 2009*, 166-181. (2009)

18. Cook, B., Gupta, A., Magill, S., Rybalchenko, A., Simsa, J., Singh, S., Vafeiadis, V.: Finding heap-bounds for hardware synthesis. FMCAD 2009, 205-212. (2009)
19. Magill, S., Tsai, M., Lee, P., Tsay, Y.: Automatic numeric abstractions for heap-manipulating programs. POPL 2010, 211-222. (2010)
20. Magill, S., Tsai, M., Lee, P., Tsay, Y.: THOR: A Tool for Reasoning about Shape and Arithmetic. CAV 2008, 428-432. (2008)
21. Magill, S., Berdine, J., Clarke, E. M., Cook, B.: Arithmetic Strengthening for Shape Analysis. SAS 2007, 419-436. (2007)
22. Bouajjani, A., Bozga, M., Habermehl, P., Iosif, R., Moro, P., Vojnar, T.: Programs with Lists Are Counter Automata. CAV 2006, 517-531. (2006)
23. Cadar, C., Dunbar, D., Engler, D. R.: KLEE: Unassisted and Automatic Generation of High-Coverage Tests for Complex Systems Programs. OSDI 2008, 209-224. (2008)
24. Lalire, G., Argoud, M., and Jeannet, B.: The interproc analyzer. <http://pop-art.inrialpes.fr/people/bjeannet/bjeannet-forge/interproc/index.html>.
25. Chin, W., Nguyen, H., Popeea, C., Qin, S.: Analysing memory resource bounds for low-level programs. ISMM 2008, 151-160. (2008)
26. Berger, E. D., Zorn, B. G., McKinley, K. S.: Composing High-Performance Memory Allocators. PLDI 2001, 114-124. (2001)
27. Lattner, C., Adve, V. S.: LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation. CGO 2004, 75-88. (2004)
28. Bouajjani, A., Drăgoi, C. et al. On Inter-Procedural Analysis of Programs with Lists and Data. PLDI2011. (2011)

Appendix A. The Abstract Semantics for List Operations

When explaining the abstract semantics for list operations, we will continue to use the recording symbols from section 5.1.

(1) $p = null$

After executing the assignment statement $p = null$, the variable p points to the special node $NULL$ and won't reach any list nodes, so the p th bit of the vrv should be 0. We can modify $vrvc$ like following:

$$\llbracket p = null \rrbracket (vrvc) = \{ \langle vr v', num' \rangle \mid \forall \langle vr v, num \rangle \in vrvc \wedge vr v \in \Gamma_p, \\ vr v' = vr v /_{\{p\} \leftarrow 0}, num' = num \} \cup vrvc_{\Delta}$$

(2) $p = q$

The variables q and p will point to the same list node after the execution, so p will reach and only reach the list nodes formerly reached by q . We can construct the new abstract state by copying the q th bit of $vr v$ to its p th bit. And $vrvc$ can be modified like the following:

$$\llbracket p = q \rrbracket (vrvc) = \{ \langle vr v', num' \rangle \mid \forall \langle vr v, num \rangle \in vrvc \wedge vr v \in \Gamma_p \cup \Gamma_q, \\ vr v' = vr v /_{\{p\} \leftarrow q}, num' = num \} \cup vrvc_{\Delta}$$

(3) $p = q \rightarrow next$

After the assignment statement $p = q \rightarrow next$ is executed, the reachability properties of the two variables are identical for all the list nodes except the

list node that q pointed to formerly. For each tuple $\langle vr v, num \rangle$ in $vr vsc$ with $vr v \in \Gamma_p \cup \Gamma_q$:

- 1. If $vr v \neq \gamma_q^0$, then we can replace the p th bit of $vr v$ with its q th bit;
- 2. Otherwise, the corresponding list nodes could be divided into two categories. At first, because p will not reach the list node formerly pointed to by q , we can replace the p th bit of γ_q^0 with 0 and construct a new tuple with num equaling 1 to describe the list node. Secondly, for other list nodes with VRV γ_q^0 , both p and q will reach them after the execution. We can replace the p th bit of $vr v$ with its q th bit, making both the p th bit and the q th bit equal 1. But because we have excluded one list node, so the num part of the tuple should be decreased by 1.

To sum up, when executing $p=q \rightarrow next$, we can modify $vr vsc$ as following:

$$\llbracket p = q \rightarrow next \rrbracket (vr vsc) = \{ \langle vr v', num' \rangle \mid \forall \langle vr v, num \rangle \in vr vsc \wedge vr v \in \Gamma_p \cup \Gamma_q. \\ vr v' = vr v / \{p\} \leftarrow q, num' = (vr v == \gamma_q^0) ? num - 1 : num \} \\ \cup \{ \langle \gamma_q^0 / \{p\} \leftarrow 0, 1 \rangle \} \cup vr vsc_{\Delta}$$

For an example, let's consider the execution in figure 12. Four variables may point to list nodes. The shape graphs are given on the left for convenient and the VRVSC are given on the right part. The gray cell in source VRVSC represents γ_q^0 and which stands for the two list nodes n_2, n_3 . When executing the assignment statement $p = q \rightarrow next$, the p th bit will be replaced with the q th bit for the VRVs 1100, 1110, 1111 and become 1100, 1110, 1111 respectively, the num corresponding to γ_q^0 (1100) will be decreased by 1 and become 1. The tuple listed in the last cell stands for the newly constructed tuple by assigning the p th bit of γ_q^0 with 0 and making the num field equal 1.

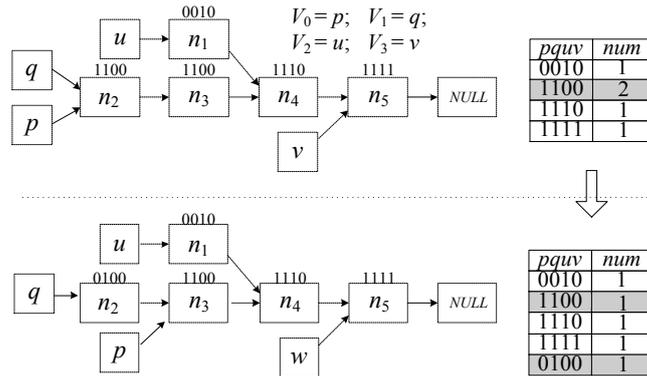


Fig.12. An example for executing $p = q \rightarrow next$

(4) $p \rightarrow next = null$

After the assignment statement $p \rightarrow next = null$ is executed, for each tuple $\langle vrv, num \rangle$ in $vrvc$ with $vrv \in \Gamma_p$:

- 1. If $vrv \neq \gamma_p^0$, then all the variables which reach γ_p^0 formerly will not reach vrv now. We can assign all the bits in $R_{\gamma_p^0}$ with 0.

- 2. If $vrv = \gamma_p^0$, then the corresponding list nodes could be divided into two categories. The VRV for the list node formerly pointed to by p should not change. In order to describe this list node, we can construct a new tuple with VRV equaling γ_p^0 and num equaling 1. As for other list nodes with VRV γ_p^0 , none variables will not reach them after the execution. We can assign all the bits in $R_{\gamma_p^0}$ with 0. Because we have excluded one list node, the corresponding num should be decreased by 1.

To sum up, we can express the abstract semantics for $p \rightarrow next = null$ as following:

$$\begin{aligned} \llbracket p \rightarrow next = null \rrbracket (vrvc) = & \{ \langle vrv', num' \rangle \mid \forall \langle vrv, num \rangle \in vrvsc \wedge vrv \in \Gamma_p. \\ & vrv' = vrv / \{ R_{\gamma_p^0} \} \leftarrow 0, num' = (vrv == \gamma_p^0) ? num - 1 : num \} \\ & \cup \{ \langle \gamma_p^0, 1 \rangle \} \cup vrvsc_{\Delta} \end{aligned}$$

(5) $p \rightarrow next = q$

After executing the assignment statement $p \rightarrow next = q$, the reachability properties of the two variables are identical for all the list nodes except the list node that p pointed to formerly. For each tuple $\langle vrv, num \rangle$ in $vrvc$ with $vrv \in \Gamma_p \cup \Gamma_q$:

- 1. If $vrv \in \Gamma_q$, then vrv can be reached by the variables which can reach γ_p^0 formerly. We can replace all the bits in $R_{\gamma_p^0}$ with the q th bit, making these bits equal 1;

- 2. If $vrv \in \Gamma_p - \Gamma_q$ but $vrv \neq \gamma_p^0$, then vrv will not be reached by the variables which formerly reach γ_p^0 . We can replace all the bits of vrv in $R_{\gamma_p^0}$ with the q th bit, making these bits equal 0;

- 3. If $vrv = \gamma_p^0$, then the corresponding list nodes could be divided into two categories. In order to describe this list node, we can construct a new tuple with VRV equaling γ_p^0 and num equaling 1. The VRV for the list node formerly pointed to by p should not change. As for other list nodes with VRV equaling γ_p^0 , none variables will not reach them after the execution. Because we only consider non-circular list now, $\gamma_p^0[q]$ must equal 0. We can replace all the bits in $R_{\gamma_p^0}$ with the q th bit, making all these bits of vrv equal 0. But because we have excluded one list node, the num should be decreased by 1.

To sum up, we can express the abstract semantics for $p \rightarrow next = q$ as following:

$$\begin{aligned} \llbracket p \rightarrow next = q \rrbracket (vrvc) = & \{ \langle vr v', num' \rangle \mid \forall \langle vr v, num \rangle \in vrvc \wedge vr v \in \Gamma_p \cup \Gamma_q. \\ & vr v' = vr v / \{ R_{\gamma_p^0} \} \leftarrow q, num' = (vr v == \gamma_p^0) ? num - 1 : num \} \\ & \cup \{ \langle \gamma_p^0, 1 \rangle \} \cup vrvc_{\Delta} \end{aligned}$$

For an example, let's consider the execution of $p \rightarrow next = u$ in figure 13. The black cell in source abstract state represents γ_p^0 which stands for the list node n_3 . When executing the assignment statement $p \rightarrow next = u$, the bits in $R_{\gamma_p^0} (\{p, q\}$ in this case) will be replaced with the u th bit for the VRVs 0010,1100,1110,1111 generating 1110,0000,1110,1111 respectively. The num corresponding to γ_p^0 (1100) will be decreased by 1 and become 0. The tuple listed in the last cell describes the VRV for n_3 . As we have pointed in section 5.1, a default called operation *Compact()* should be called to handle the output VRVSC after each assignment statement is executed. For this example, we should delete a tuple $\langle 0000, 0 \rangle$ and join the two tuples with the same VRV 1110 as you can see in figure 13.

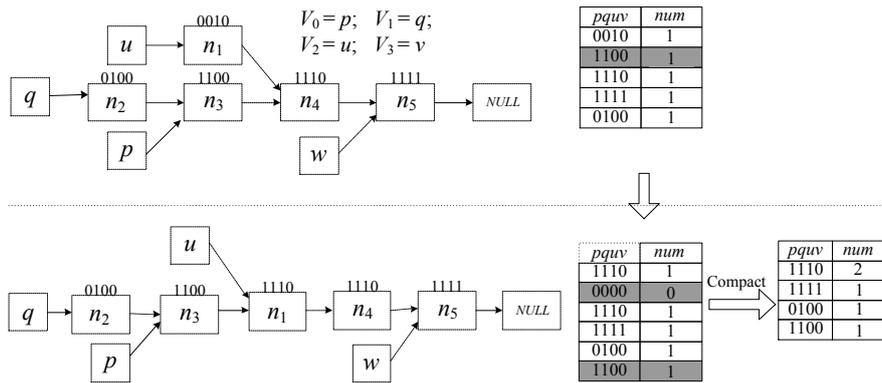


Fig.13. An example for executing $p \rightarrow next = u$

(6) $p = malloc()$

After executing the assignment statement $p = malloc()$, the variable p will point to a new created list node and won't reach all the already existed list nodes, so the p th bit of all the $vr v$ for all tuples in $vrvc$ should be 0. In order to describe the new created list node, we can create a tuple with only the p th bit of its $vr v$ equaling 1 and its num equaling 1.:

We could express the operational semantics for $p = malloc()$ as following.

$$\begin{aligned} \llbracket p = malloc() \rrbracket (vrvc) = & \{ \langle vr v', num' \rangle \mid \forall \langle vr v, num \rangle \in vrvc \wedge vr v \in \Gamma_p. \\ & vr v' = vr v / \{ i \} \leftarrow 0, num' = num \} \cup \{ \langle new(\{p\}), 1 \rangle \} \cup vrvc_{\Delta} \end{aligned}$$

(7) $free(p)$

After executing $free(p)$, all the variables which reach γ_p^0 formerly will never reach the VRVs in Γ_p . We could assign 0 to all the bits in $R_{\gamma_p^0}$ for all the VRVs

Renjian Li, Ji Wang, Liqian Chen, Wanwei Liu, Dengping Wei

in Γ_p . Because we have deallocated a list node, the num corresponding to γ_p^0 should be decreased by 1. So we can express the abstract semantics as following:

$$\llbracket free(p) \rrbracket (vrvc) = \{ \langle vrv', num' \rangle \mid \forall \langle vrv, num \rangle \in vrvsc \wedge vrv \in \Gamma_p. \\ vrv' = vrv / \{R_{\gamma_p}\} \leftarrow 0, num' = (vrv == \gamma_p^0?) num - 1 : num \} \cup vrvsc_{\Delta}$$

Renjian Li received his M.S. degree in computer science from National University of Defense Technology (NUDT) in 2006. He is currently a Ph.D. candidate in School of Computer, NUDT. His research interests are in the areas of program analysis and verification.

Ji Wang received his Ph.D. degree in computer science from National University of Defense Technology in 1995. He is currently a professor at National Laboratory for Parallel and Distributed Processing of China. His research interests include high confidence software and systems, software engineering and distributed computing. He is a senior member of China Computer Federation.

Liqian Chen is an assistant professor at the School of Computer Science, National University of Defense Technology, Changsha, China. He received his PhD degree in Computer Science from National University of Defense Technology in 2010. His research interests include software engineering, program analysis and verification.

Wanwei Liu receives his Ph.D degree in Jun. 2009 at NUDT. He is currently a lecturer in School of CS, NUDT. His research interests mainly include: automata theory, logic in CS and model checking.

Dengping Wei is an assistant professor at the School of Computer Science, National University of Defense Technology, Changsha, China. She received her PhD degree in Computer Science from National University of Defense Technology in 2011. Her research interests include Semantic Web, Web service and Cyber-Physical Systems.

Received: March 2, 2011; Accepted: April 22, 2011.

A Model-Based Software Development Method for Automotive Cyber-Physical Systems

Zhigang Gao¹, Haixia Xia², and Guojun Dai¹

¹ College of Computer Science, Hangzhou Dianzi University,
Hangzhou 310018, China
gaozhigang@zju.edu.cn, daigj@hdu.edu.cn

² College of Informatics & Electronics, Zhejiang Sci-Tech University,
Hangzhou 310018, China
lansehaij@163.com

Abstract. The development of automotive cyber-physical systems (CPS) software needs to consider not only functional requirements, but also non-functional requirements and the interaction with physical environment. In this paper, a model-based software development method for automotive CPS (MoBDAC) is presented. The main contributions of this paper are threefold. First, MoBDAC covers the whole development workflow of automotive CPS software from modeling and simulation to code generation. Automatic tools are used to improve the development efficiency. Second, MoBDAC extracts non-functional requirements and deals with them in the implementation model level and source code level, which helps to correctly manage and meet non-functional requirements. Third, MoBDAC defines three kinds of relations between uncertain physical environment events and software internal actions in automotive CPS, and uses Model Modifier to integrate the interaction with physical environment. Moreover, we illustrate the development workflow of MoBDAC by an example of a power window development.

Keywords: Automotive cyber-physical systems, non-functional requirements, physical environment, model-based methods, model transformation, code generation.

1. Introduction

From smart power grids to intelligent homes and from environmental monitoring to transportation systems [1-3], CPS are increasingly permeated into every aspect of our society. Unlike traditional computer systems which mainly focus on computing and information processing, CPS need to consider computing, communication, physical environment, and their interaction [4]. Therefore, CPS software is hard to develop because developers need to consider functional properties, non-functional properties,

such as timeliness, energy, memory, safety and reliability, and the interaction with physical environment.

In the recent two decades, both academia and industry have made efforts to explore approaches which are more applicable to the development of embedded software, such as the real-time object-oriented modeling (ROOM) [5], the UML-RT (implemented by IBM Rational Rose) [6], the Specification and Description Language (SDL), a language widely used in telecommunications domain [7], and Model-integrated Computing [8]. Among them, the model-based development of embedded software has become one of the most promising methods.

The current research on model-based development of embedded software basically focuses on high-level modeling and simulations (e.g. Ptolemy [9]), or integration methodologies of tools (e.g. MoBIES [10]) for embedded software, but seldom involves a suite of complete implementation for a specific domain and considers both the non-functional requirements and environment requirements. Although there are a few of model-based CPS development methods [11-12], these methods only stay on high-level design model or require well-defined components. In this paper, we propose MoBDAC, which supports the implementation under the OSEK/VDX (Open Systems and the Corresponding Interfaces for Automotive Electronics (in German)/Vehicle Distributed eXecutive (in French)) specification [13]. MoBDAC covers the whole development workflow of automotive CPS applications from modeling and simulation to code generation. MoBDAC helps to increase development efficiency and improve software quality, and it is easy to integrate the interaction with physical environment.

The rest of this paper is organized as follows. Section 2 summarizes the related work. Section 3 presents the architecture of MoBDAC. The implementation process is presented in section 4. In section 5 we illustrate the implementation of a power window control system as a case study and discuss the characteristics of MoBDAC. We conclude this paper in section 6.

2. Related Work

Ptolemy is one of the first research projects in the model-based development of embedded software. Ptolemy II [9], the current modeling tool of Ptolemy, is a hierarchical heterogeneous modeling environment for modeling, simulation, and design of concurrent, real-time, embedded systems. The purpose of Ptolemy II is to provide a trial platform for heterogeneous models of computation (MOC). Ptolemy II supports many kinds of MOC, such as synchronous dataflow (SDF), process networks (PN), finite-state machines (FSM), etc. Different components can be hierarchically integrated into a complex system under the government of different MOC. GME (The Generic Modeling Environment) [14] is a domain-specific meta-modeling environment, which provides different views to model the objects, relations and constraints. Because GME only provides meta-meta models, a whole

modeling process includes third phases: meta-modeling, modeling, and system-modeling. MoBIES (Model-Based Integration of Embedded Software) [10] is a tool chain for the integration of reusable embedded software. MoBIES integrates several kinds of existing commercial and academic tools to cover the modeling, model analysis, code generation, and runtime analysis in the development of embedded software, and uses standard XML file formats to exchange information among different tools. Currently, many commercial model-based development tools, such as Matlab/Simulink [15], MetaEdit+ [16], DOME [17], Rhapsody [18], etc., have been used to the development of embedded software, such as automotive electronic or avionic controlling software.

In recent years, CPS software development has been attracting more and more attention. Woo et al. [19] present a formal software development method with a suite of feedback control laws and efficient resource monitoring mechanism to deal with system failure effectively. Lin et al. [20] present an integrated simulation method in order to accurately reflect the operation and interaction between the cyber aspects and the physical aspects of CPS. Ma et al. [21] present a high-confidence cyber-physical alarm system (CPAS) and discuss its requirements, system models and implementation.

A few of research efforts have been made to develop CPS software with model-based methods. Magureanu et al. [11] present a model-based CPS development method for gas distribution. Their method mainly focuses on building the high-level models using UML. Bhatia et al. [12] present a model-based framework called SysWeaver to model, integrate, analyze, verify, and implement AUTOSAR-compliant automotive systems, which extends AUTOSAR (Automotive Open System Architecture) in order to meet the real-world requirements of automotive CPS, such as timeliness, fault tolerance, feedback, etc. In the development of automotive systems, SysWeaver requires the components are available and have well-defined parameters.

3. The Architecture of MoBDAC

The architecture of MoBDAC is shown in Figure 1. The main workflow of development includes four steps. First, extract software specifications from system specifications. Second, use modeling tools to build the models in problem domains (MPD), and then perform simulation in order to verify the correctness of models. Third, transform MPD into the models in implementation domains (MID). Finally, MID are used to generate code. Note that both non-functional requirements and the interaction with physical environment are extracted from system specifications besides software specifications, the non-functional requirements are used by analysis tools to verify whether the non-functional requirements of the software are met, and the information of the interaction with physical environment is used by Model Modifier to modify MID in order to generate correct code.

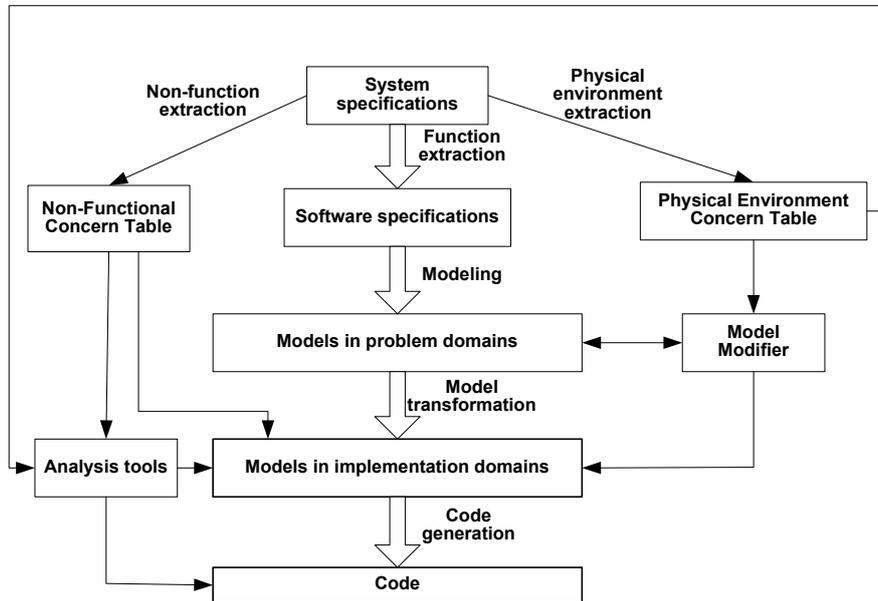


Fig. 1. Architecture of MoBDAC

3.1. Function Extraction

The purpose of function extraction is to extract software specifications from system specifications of automotive CPS. For automotive CPS, their system specifications include the following three aspects.

- Functional requirements. Functional requirements define the behavior of a system and what the system does [22]. In automotive CPS, functional requirements define what functions they include, the operating process when completing a function, and the relations among different operations, etc. Developers can specify the functional requirements of automotive CPS according to what functions their subsystems (such as the body subsystem, the safety subsystem, etc) include, and how to carry out these functions.
- Non-functional requirements. Non-functional requirements define the quality of a system and how well the system should work [22]. Non-functional requirements specify global constraints [23], such as timeliness, safety, fault-tolerance, energy, etc [24-26]. Developers can specify the non-functional requirements of automotive CPS according to the global constraints from systems specifications.
- Physical environment requirements. For automotive CPS, different subsystems may work in different physical environment, and have different interaction modes and requirements. Physical environment

requirements define the interaction modes and requirements between automotive CPS software and their physical environment. For example, for an in-vehicle air-conditioning system, it detects the temperature and humidity in the vehicle, and decides its working status; for a backup radar, it detects the distance between the vehicle and obstacles, and decides its alarm status.

When performing function extraction, it only extracts the function requirements and put them into software specifications. The non-functional requirements and the interaction with physical environment are extracted by non-function extraction and physical environment extraction, and put them into Non-Functional Concern Table and Physical Environment Concern Table respectively.

3.2. Modeling

System specifications and software specifications are all text which is mainly used to communicate among designers. During modeling, we build MPD which denote the structures and functions of software from software specifications, and verify their correctness by simulation. MPD describe the structures of software, the relations among different parts, and the transition relations among different states.

Because they do not consider the characteristics of deployment platforms, MPD belong to Platform Independent Models (PIM), MPD are more suitable for designers to concentrate themselves on high-level function design and can enhance the portability of software. Moreover, there is a Model Modifier in the right of Figure 1. Model Modifier can build the relations between external physical environment events and software internal actions according to some rules, and then modify MID in order to make the modeling and simulation in MPD independent of physical environment. In automotive CPS, some physical environment events are certain. For example, in-vehicle temperature need be detected at a specific period. For certain interaction with physical environment, it is enough to model its behavior in functional requirements. For uncertain physical environment events, MoBDAC defines three kinds of relations between physical environment events and software internal actions.

- Correlative relation. When a physical environment event occurs, a corresponding software internal action must happen. For example, it is a correlative relation between a physical event of turning on an in-vehicle light and a software action that the state of the in-vehicle light turns from off to on. Developers can model the state of the in-vehicle light in problem domains to denote the effects caused by the corresponding physical event.
- Exclusive relation. The occurrence of a physical environment event means that a software internal action will not happen. For example, a power window will not move up when an obstacle is detected. There is an exclusive relation between the power window moving up and the

obstacle event occurring. Developers can model the software action in problem domains and process the results caused by the physical event in the reverse logic.

- Complementary relation. A physical environment event maybe happens when a software internal action occurs. For example, a wheel slip maybe occurs when a brake action is executing. Developers can model the software action in problem domains and detect the physical event in the software action.

In automotive CPS, developer can use correlative relation and exclusive relation to describe the interaction between passive reaction systems (e.g., Power Window System, Supplemental Restraint System, etc.) and physical environment, and complementary relation to describe the interaction between active reaction systems (e.g., Anti-lock Brake System, Anti Slip Regulation, etc.) and physical environment. Using these three relations, developers can model and simulate software functions in MPD without considering the influence of the uncertain physical environment events, Model Modifier adds the interaction with physical environment to MID after model transformation.

3.3. Model Transformation

MPD are independent of platforms and implementation. After they are built and verified, MPD need to be transformed into the models for specific hardware and software platforms, i.e., Platform Specific Models (PSM). The model transformation is composed of two steps:

- Model Analysis. Model analysis extracts all kinds of elements in MPD, the functions of elements, and the relations among elements according to the characteristics of the modeling tools. By model analysis, we know what elements are useful for MID, what elements are useless for MID (i.e. they will be filtered during model transformation), what elements server for the same functions, what elements share the same resource, the dependence relations among elements, etc, in order to provide support for generating MID.
- MID Generation. Because tasks are widely used software models in current software implementation, we use tasks as MID. We need organize the elements in MPD into tasks during model transformation according to their functions in automotive CPS. Because there may be hierarchical relations among the elements in MPD, it needs a suitable granularity to transform the elements in MPD into the elements in MID. Moreover, we need decide the relations among tasks, such as precedence order, the message passing, etc. Note that it is an important problem to assign task properties, such as periods, deadlines, execution voltage, processor temperature, etc. Some task properties may be decided according to the results of model analysis (for example, periods), some from Non-Functional Concern Table (for example, deadlines), and others according to specific rules or algorithms (for example, priorities and execution voltage). Because

the information of deployment platform is available, we use analysis tool to verify whether MID meet the non-functional requirements.

3.4. Code Generation

After building MID, we can generate the code for OSEK-compatible OS according to the relations among tasks and the properties of tasks. Some properties of tasks highly depend on their implementation code. For example, the WCET (Worst-Case Execution Time) of tasks are usually evaluated on source code level or assemble code level. In automotive cyber-physical system development, MoBDAC supports two methods to improve the flexibility when evaluating code-dependent task properties, i.e., from the task model level or the source code level. We can use analysis tools to verify whether the software implementation meets its non-functional requirements. If the software implementation does not meet its non-functional requirements, we can modify the system design and repeat the above process until non-functional requirements are met.

After generating source code, we can use development tools which are usually available from chip manufacturers (e.g. CodeWarrior from Freescale Semiconductor [27]) to generate the machine code for special hardware platforms, e.g. DSP, MCS51, MPC555, and HCS12.

4. The Implementation of MoBDAC

Currently, we have implemented the development workflow of MoBDAC by an Automotive Electronic CPS (AECPS) tool chain which combines Ptolemy II with the development tools designed by ourselves, and the results have proved the effectiveness of MoBDAC. The following is the implementation of MoBDAC.

4.1. Function Extraction

Because system specifications and software specifications are designer-oriented documents, they mainly make designers understand the system and software requirements more easily and exactly. We perform function abstraction by hand. From system specifications, we find the functions and relations for software parts and put them into software specifications. After all specifications relevant with software have been abstracted, we get software specifications.

4.2. Modeling and Simulation in Problem Domains

In Ptolemy II, designers build MPD according to software specifications, and then verify their correctness by simulation. Modeling includes two steps. First, choose suitable MOC (Models of Computation). The choices of MOC are made mainly according to the continuity or discreteness of time, and the synchrony or asynchrony of events, etc. For example, the model of a power window control system is a hybrid model, where the states of the windows can be described by FSM, and the position of the window is a value that is suitable to be described by CT (Continuous Time) model. Second, construct MPD. Once the model is built, developers can observe the running results of the models in simulation windows. If the results of the simulation are not consistent with the requirements of software specifications, designers can check the models and debug the errors during modeling.

4.3. Model Transformation

Models built in Ptolemy II are independent of platforms and implementation. They need to be transformed into the ones under OSEK-compatible OS.

Major modeling elements in Ptolemy II [28] include:

- **Entity** is a text segment with specific functions, e.g. directors, actors (including the ports, relations, and links that belong to directors/actors).
- **Port** is an input or output interface of an entity.
- **Relation** is the route of data or messages transmitted between different entities or just inside one entity.
- **Link** is a connection between input/output interfaces of entities and relations.
- **Property** is a characteristic of an entity element, such as its position, parameter, and name, etc.

In Ptolemy II, entities are the highest-level elements. Other elements are attached to entities. In the model built by AECPSDesigner, a modeling tool designed by us for MID, there are also elements as listed above, but they do not have the same meaning. Transforming MPD to MID becomes the key problem of the design process. We present the method of stepwise refining to transform MPD to MID. The transformation process is shown in Figure 2.

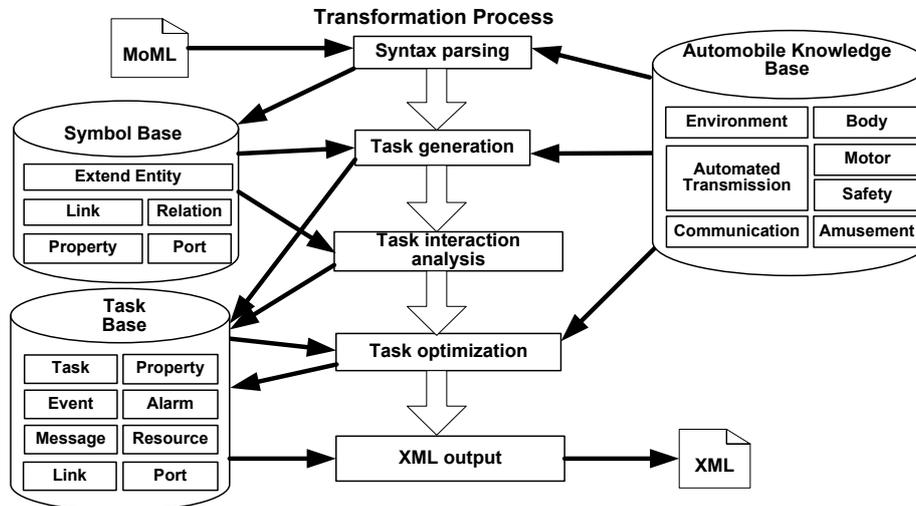


Fig. 2. The process of model transformation

In Figure 2, the transformation workflow of models is composed of five steps, i.e., syntax parsing, task generation, task interaction analysis, task optimization, and XML output. There are three databases, i.e., Symbol Base, Task Base, and Automobile Knowledge Base, and multiple tables in each database. Besides the entity name field, records of Extend Entity table in Symbols Base include the directors that entities belong to, together with the functions of entities. Records of Task table in Task Base include the functions of tasks as well as the entities it includes. Automobile Knowledge Base includes the subsystems of the automotive CPS, their functions, and the importance level of their functions according to real-time and safety-critical degree. For example, the body subsystem consists of the power window control function, the power skylight control function, the power rear-view mirror control function, and the seat adjusting function, and so on. Automobile Knowledge Base helps to partition tasks in model transformation according to their functions, and merge different tasks according to their importance levels.

1) Syntax Parsing: In order to obtain the information of the models built in Ptolemy II, we analyze the output the MoML (Modeling Markup Language) file. We first extract the entities and put them into Extend Entity table as a new record. Then we take out other parts of entities, e.g. the directors it belongs, and put them into the records of relevant entities as new fields. After that, we search Automobile Knowledge Base to find out the functions of the entities and put them into Extend Entity table as a field. The properties, relations, ports and links that we get from the MoML file should also be put into the relevant tables.

2) Task Generation: We classify the entities in Symbol Base and then create tasks, as well as find the properties and functions of each entity and output them to Task Base. We use the algorithm shown in Figure 3 to classify the entities. In Figure 3, if the MOC of F is relevant to events, Classify algorithm finds all entities depending on these events and marks the same task tag (Line 4-Line 13). Otherwise, Classify algorithm finds all entities with the same function and marks them the same task tag (Line 15-Line 22). Note that all unmarked entities and ungrouped entities are marked the same task tag (Line 12, Line 21).

```
1 Algorithm Classify(FN)
2 Begin
3   F = The collection of all entities having function FN in Extend Entity table;
4   If (The MOC of F is relevant to events) then
5     Begin
6       ai = The first entity in F;
7       While (ai <> null and the task tag of ai are unmarked)
8         Begin
9           Find out the entities depending on the same one or more events, merge them to a group,
           and mark a new task tag;
10          ai = the next entity in F;
11        End
12      Classify all the unmarked entities in F into a group, and mark a new task tag;
13    End
14  Else
15    For (each entity ai) in F
16      If (ai completes the same function with some entities which belong to task ti ) then
17        Set the task tag of ai to be ti, and merge them to a group;
18      Else
19        Allocate a new task tag ti;
20      Endif
21    Classify all the ungrouped entities in F into a group, and mark a new task tag;
22  Endif
23 End
```

Fig. 3. Entities classifying algorithm

After classifying the entities, all entities with the same task tag are grouped into one task and saved into Task table in Symbol Base. From Property table in Symbol Base, a corresponding property table can be created for the entities in Task Base by performing the following operation:

a) If an entity is only relevant with other entities in the same task, eliminate its properties.

b) If an entity is relevant with the entities of other tasks, combine all properties of the entity as the task's global properties and then put them into Property table in Task Base.

According to the functions of each entity in Automotive Knowledge Base, put its importance level property into Task table in Task Base.

3) Task Interaction Analysis: In order to find the relations between tasks, we employ the algorithm defined in Figure 4. In Figure 4, for every task in Task table, Interaction algorithm finds its links, ports, and properties and outputs them to corresponding tables in Symbol Base (Line 3-Line 18). After that, Interaction algorithm finds the corresponding messages, events, alarms,

and resources for every task in Task table and output them to corresponding tables in Symbol Base (Line 20-Line 38).

```

1 Algorithm Interaction()
2 Begin
3   ti = The first task in Task table;
4   While (ti <> null)
5     Begin
6       R = All relations in Relation table contained by entities pertaining to ti;
7       ri = The first relation in R;
8       While (ri <> null)
9         Begin
10          If (ri links with entities belonging to other tasks in Task table) then
11            Add a new link to Connection table in Task Base, and add the linking ports
            between tasks to Port table;
12          Else
13            Add a corresponding property to Property table;
14          Endif
15          ri = The next relation in R;
16        End
17        ti = The next task in Task table;
18      End
19      ti = The first task in Task table;
20      While (ti <> null)
21        Begin
22          ci = the first link associated with ti in Relation table;
23          While (ci <> null)
24            Begin
25              If (The port connected with ci transmits data) then
26                Add a message to Message table;
27              Else
28                If (The port connected with ci transmits sporadic events) then
29                  Add an event to Event table;
30                ElseIf (The port connected with ci transmits periodic events)
31                  Add an alarm to Alarm table;
32                Endif
33              Endif
34              Search other tasks having the same property. If it succeeds, convert the property to
              resource and add it to Resource table;
35              ci = The next link associated with ti in Relation table ;
36            End
37            ti = The next task in Task table;
38          End
39        End

```

Fig.4. Task Interaction analysis algorithm

4) Task Optimization: The tasks generated are analyzed to decide whether they should be merged in order to reduce task number. The following factors should be taken into account:

- a) The dependence relationship between tasks.
- b) The importance level of each task.

We employ the merging algorithm shown in Figure 5 to perform task optimization. In Figure 5, TaskOptimizing algorithm finds the tasks whose importance level of its function is no more than a threshold $P_{\text{threshold}}$ (which is set by developers) (Line 6), and merged them into a task in order to reduce the task number in the system (Line 11-Line 12).

Note that the merging operation of two tasks is allowed if it does not result in an annular dependence relation, and the merged tasks are stored in Task Base. The critical tasks will not be merged. For example, for safety-critical tasks, the real-time property is highly demanded, and they will not be merged.

```

1 Algorithm TaskOptimizing()
2 Begin
3   ti=The first task in Task table;
4   While (ti <> null)
5     Begin
6       If (ti <> null and the importance level of ti's function <= Pthreshold) then
7         tj=The first task in Task table;
8         While (tj <> null)
9           Begin
10            If (the importance level of tj's function <= Pthreshold and no cyclic dependences of
11              resources and events among all tasks after tj composited with ti) then
12              Add the properties, events, relations, ports, connections, resources and
13              messages to corresponding tables associating with ti;
14              Delete ti and its property, events, relations, ports, connections, resources and messages;
15            Endif
16            tj=The next task in Task table;
17          End
18        Endif
19      End
20    End

```

Fig. 5. Task optimizing algorithm

4.4. XML Output and Code Generation

We take out the tasks from Task Base and parse their functions, properties and relations with other tasks. As a result, An XML file is created.

Because the MID are based on OSEK-compatible OS, AECPSDesigner can show the implementation domain model by parsing the transformed XML file. In AECPSDesigner, developers can modify model properties, e.g. task names and priorities. Using AECPSDesigner, implementation code of models for OSEK-compatible OS can also be generated automatically through analyzing the relationship between tasks and other objects (such as alarms, events, and resources which is defined in OSEK-compatible OS). Developers can also modify/add implementation code by hand as widely supported in other model-based development tools such as Simulink and Rhapsody because fully model-based design is almost impossible currently. Currently, we have implemented the non-functional analysis for tasks in time and energy-savings by using the methods in [29-31]. In non-functional property analysis, measurement of the WCET of a program is a fundamental problem. There are many methods to estimate the WCET of a program [32], such as static program analysis, measurement, simulation, etc. In our current implementation, we use measurement-based method because it has been widely supported by current development tools for automotive electronics. For example, after defining the parameters of worst-case execution path, we can measure the worst-case execution time of a program by using CodeWarrior Debugger to simulate the microcontroller's running and obtain the number of the processor's clock cycles elapsed since the beginning of the simulation. After measuring the WCET of a program, we can mark the WCET of tasks in MID and analyze whether its deadline is met. Combining the characteristics of processors and OS, timing analysis and energy-saving

algorithm can be used to analyze whether the deadlines of tasks meet and improve the energy-saving effect of software.

5. Case Study and Analysis

We have applied MoBDAC to the development of automotive CPS software, and achieved good effect. In this section, we demonstrate the design process presented in section 3 through a power window control system. Figure 6 shows the network topology structure of a power window control system. In Figure 6, P1 is the processor which controls the power windows, and P2 is the processor which is responsible for information display. A passenger can press Button1 to control the *up* or *down* of the power window, and a driver can also press Button2 to send messages to P1 to control the *up* or *down* of the power window. Once it changes, the position of the power window can be sent to P2 and shown in LCD.

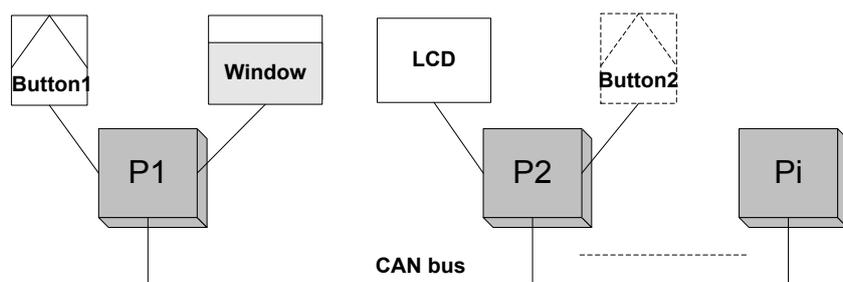


Fig. 6. The network topology structure of the power window

The power window control system is a relatively complex system with the following functions: a) manual *up*; b) manual *down*; c) automatic *up*; d) automatic *down*; and e) obstacle-detecting. Because the *up* and *down* messages from CAN bus is equivalent to these from Button1, we only consider the *up* and *down* messages regardless of their sources. For simplicity, we only consider the functions of manual *up*, manual *down*, and obstacle-detecting. Note that obstacle-detecting is a safety measure which prevents arms from being clamped when the power window is moving up. We assume the software specifications can be described as follows:

a) When a passenger pushes the *up* button once, the window move up for 4cm if the position of the power window is less than 40cm and there is no obstacle.

b) When a passenger pushes the *down* button once, the window move down for 4cm if the position of the power window is more than 0cm.

c) If an obstacle is detected during moving up, the window moves down for 4cm.

During its movement, the states of the power window are controlled by the event of *up* or *down*. We choose FSM as its MOC and classify the states of the power window into *fully_opened*, *fully_closed* and *semi_opened*.

The model of the power window built in Ptolemy II consists of three levels. The first-level model is shown in Figure 7.

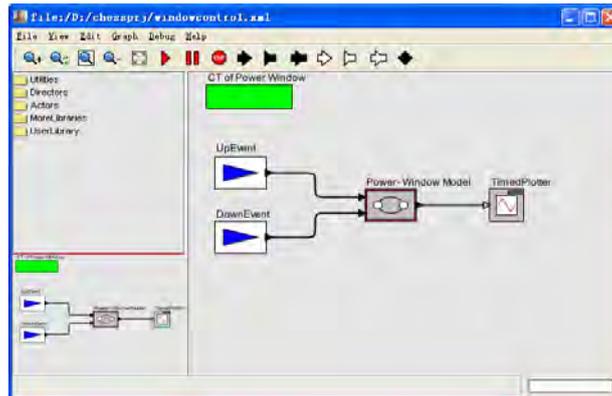


Fig. 7. The first-level model of the power window

In Figure 7, there are two discrete periodical event sources (UpEvent and DownEvent), a power window model and a Timedplotter. The two discrete periodical event sources are used to generate *up* events, and *down* events respectively. Their configuration is shown in Table 1. The offsets denote the time span from the occurrences of events to the period of the events. The position of the window is output to TimedPlotter in order to observe its value.

Table 1. The configuration of event sources

Event source	Period (second)	offsets	values
UpEvent	25	{1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0}	{1, 1,1,1, 1,1,1, 1,1,1}
DownEvent	25	{11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0}	{1, 1,1,1, 1,1,1, 1,1,1}

The second-level model is the power window model. Note that the function model we actually need is the power-window model. The first-level model is used to simulate the controlling effects.

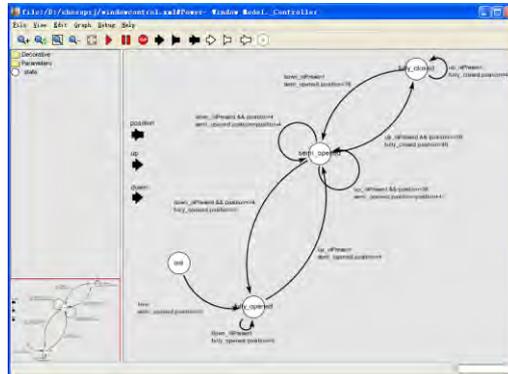


Fig. 8. The second-level model of the power window

The second-level model is a FSM denoting the state transition of the power window, as shown in Figure 8. The FSM consists of four states: *init*, *fully_opened*, *fully_closed*, and *semi_opened*. We assume the initial state of the FSM is *init*. In fact, the *init* state is an additive state for the convenience of controlling. The FSM will immediately transfers into *fully_opened* state from the *init* state when it begins to work. The end states of the FSM are *fully_opened*, *fully_closed*, or *semi_opened*. The obstacle-detecting function should be implemented when an UpEvent occurs. However, whether there is an obstacle depends on physical environment is uncertain. It is difficult to simulate this uncertainty. We know obstacles need to be detected when UpEvent events occur and obstacle-detecting has the exclusive relation with UpEvent events. The Model Modifier records the following rules:

Name: Obstacle-detecting.

Relevant Event: UpEvent.

Relationship: Exclusive.

Expression: if (it is *semi_open*) power-window: position = power-window: position -4; if (it is *fully_opened*) NO ACTION.

In the above expression, it reduces the position of the power window by 4cm when its state is *semi_open*; and takes no action when its state is *fully_opened*.

After Model Modifier records the above rules, we need not consider the obstacle-detecting function in the MPD. The relations among different states, triggering conditions (*guardExpression* in Ptolemy II) and triggering actions (*setActions* in Ptolemy II) are shown in Table 2.

In Table 2, *down_isPresent* denotes the occurrence of a *down* event; *up_isPresent* denotes the occurrence of an *up* event; *position* denotes the current position of the power window. *semi_opened*, *fully_opened*, and *fully_closed* are refined into the third-level models with the same name, i.e. the window position model. *.*position* denotes the position of a power window in the third-level models.

Table 2. State transition of the power window

State transition	guardExpression	setActions
Init->fully_opened	True	fully_opened.position=0
semi_opened	down_isPresent	semi_opened.position
-> semi_opened	&& position>4	=position-4
semi_opened	up_isPresent	semi_opened.position
-> fully_opened	&& position<36	=position+4
semi_opened	down_isPresent	fully_opened.position =0
-> fully_opened	&& position<=4	
semi_opened	up_isPresent	fully_closed.position =40
->fully closed	&& position≥36	
fully_opened	down_isPresent	fully_opened.position =0
-> fully_opened		
fully_opened	up_isPresent	semi_opened.position =4
-> semi_opened		
fully closed	up_isPresent	fully_closed.position =40
->fully closed		
fully closed	down_isPresent	semi_opened.position =36
-> semi_opened		

The third-level model is a model of CT denoting the position of the power window. There are three third-level models which are corresponding to the refined states of *semi_opened*, *fully_opened*, and *fully_closed* respectively. The three third-level models have the same structure, as shown in Figure 9.

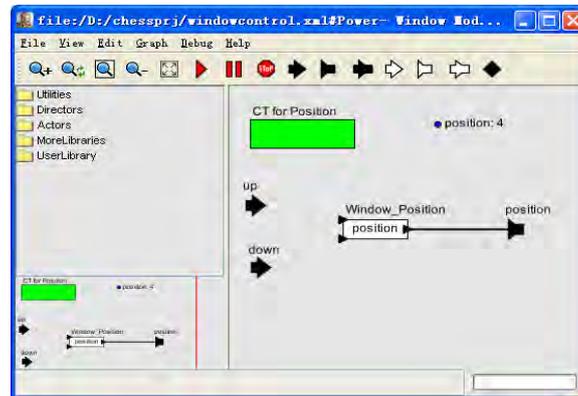


Fig. 9. The third-level model of the power window

In Figure 10, the *position* at the top right is a parameter denoting the position of the power window. The *Window_Position* is an expression actor. It uses the *position* parameter as its input and directly outputs it to the *position*

port. The *position* port is connected to TimedPlotter in order to display the position of the power window.

From the events source characteristics, we can know that the power window should move up and down in turn. The simulation result in Ptolemy II is shown in Figure 10.

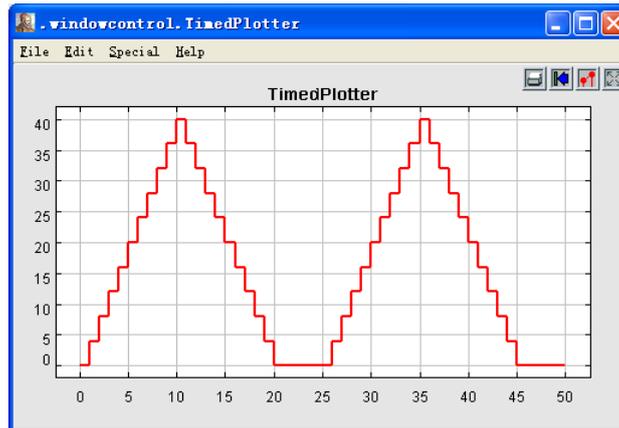


Fig. 10. The simulation result of the power window

In this paper, we don't describe all the details of the MoML file for the power window model due to space constraints, and only explain the essential parts for the model transformation.

```
1 <?xml version="1.0" standalone="no"?>
2 <!DOCTYPE entity PUBLIC "-//UC Berkeley//DTD MoML 1//EN" "http://ptolemy.eecs.berkeley.edu/xml/
  dtd/MoML_1.dtd">
3 <entity name="windowcontrol" class="ptolemy.actor.TypedCompositeActor">
4   <property name="_createdBy" class="ptolemy.kernel.attributes.VersionAttribute" value="5.0.1">
5     </property>
6   <property name="CT of Power Window" class="ptolemy.domains.ct.kernel.CTMixedSignalDirector">
7     <property name="startTime"
8       </property>
9   </property>
10  <entity name="Power-Window Model" class="ptolemy.domains.fsm.modal.ModalModel">
11    ...
12  </entity>
13  <entity name="UpEvent" class="ptolemy.domains.ct.lib.EventSource">
14    <property name="period" class="ptolemy.data.expr.Parameter" value="25">
15      </property>
16    <property name="offsets" class="ptolemy.data.expr.Parameter" value="{1.0,2.0,3.0,4.0,5.0,
17      6.0,7.0,8.0,9.0,10.0}">
18      </property>
19    <property name="values" class="ptolemy.data.expr.Parameter" value="{1, 1,1,1, 1,1,1, 1,1,1}">
20      </property>
21  </entity>
22  ...
23  <relation name="relation3" class="ptolemy.actor.TypedIORelation">
24    </relation>
25  ...
26  <link port="UpEvent.output" relation="relation3"/>
27  ...
28 </entity>
```

Fig. 11. The first-level model stored in MoML file

The MoML file of the first-level model is shown in Figure 11. Note that some unimportant details have been omitted by suspension points for easy to understand.

- Line 1-2: They define the version and Document Type Definitions (DTD) used in this MoML file.
- Line 3-5: The file name of the MoML file is *windowcontrol*; the model is a composite model (including different MOCs); the version of Ptolemy II is 5.0.1. The range of this entity is from line 3 to line 28.
- Line 6-9: The name of the first-level model is “ \mathbb{T} of Power Window”. It is a CT model.
- Line 10-12: They define the second-level model to describe the state transition of the power window. Note that the second-level model uses the FSM model and its detail is omitted in line 11.
- Line 13-22: They define the UpEvent entity with its periods, offsets, and values from line 13 to line 21. The DownEvent entity and the TimedPlotter entity are omitted in line 22.
- Line 23-25: Relations are defined in order to represent the links among UpEvent, DownEvent, Power-Window Model, and TimedPlotter. Note that only one is shown, and the others are omitted.
- Line 26-27: The links among UpEvent, DownEvent, Power-Window Model, and the TimedPlotter are defined. Note that only one is shown, and the others are omitted.

The second-level model denotes the state transition of the power window, as shown in Figure 12.

```
1 <entity name="Power-Window Model" class="ptolemy.domains.fsm.modal.ModalModel">
2   <port name="up" class="ptolemy.domains.fsm.modal.ModalPort">
3     <property name="input"/>
4   </port>
5   ...
6   <entity name="fully_opened" class="ptolemy.domains.fsm.kernel.State">
7     ...
8   </entity>
9   ...
10  <relation name="relation6" class="ptolemy.domains.fsm.kernel.Transition">
11    <property name="guardExpression" class="ptolemy.kernel.util.StringAttribute"
12      value="up_isPresent && position>=36">
13    </property>
14    <property name="setActions" class="ptolemy.domains.fsm.kernel.CommitActionsAttribute"
15      value="fully_closed.position=40">
16    </property>
17  </relation>
18  ...
19  <link port="fully_opened.incomingPort" relation="relation8"/>
20  ...
21  <entity name="fully_opened" class="ptolemy.domains.fsm.modal.Refinement">
22    ...
23  </entity>
24 </entity>
```

Fig. 12. The second-level model stored in MoML file

- Line 1: The name of the model is Power-Window Model. It is a FSM model.

- Line 2-5: They define an input port, i.e. up port, from line 2 to line 4. Other ports and their properties are omitted in Line 5.
- Line 6-9: They define an entity named `fully_opened` from line 6 to line 8 to denote a state of the FSM in the second-level model. Other entities and their properties are omitted in line 9.
- Line 10-17: They define a state transition with its triggering condition (*guardExpression*) and corresponding actions (*setActions*) from line 10 to line 16. Other state transition, triggering condition and corresponding actions are omitted in line 17.
- Line 18-19: They define all the links used in the second-level model.
- Line 20-23: they denote a third-level model refined from the `fully_opened` states from line 20 to line 22. Other third-level models from the refinement of `fully_closed` and `semi_opened` are omitted in line 23.

The third-level model denotes the position of the power window, as shown in Figure 13.

```
1 <entity name="fully_opened" class="ptolemy.domains.fsm.modal.Refinement">
2   <property name="CT for Position"
3     class="ptolemy.domains.ct.kernel.CTEmbeddedDirector">
4   ...
5 </property>
6 <port name="up" class="ptolemy.domains.fsm.modal.RefinementPort">
7   <property name="input"/>
8 ...
9 </port>
10 ...
11 <entity name="Window_Position" class="ptolemy.actor.lib.Expression">
12   <property name="expression" class="ptolemy.kernel.util.StringAttribute"
13     value="position">
14   </property>
15 ...
16 </entity>
17 <relation name="relation" class="ptolemy.actor.TypedIORelation">
18 </relation>
19 <link port="position" relation="relation"/>
20 <link port="Window_Position.output" relation="relation"/>
21 </entity>
```

Fig. 13. The third-level model stored in MoML file

- Line 1: The name of the model is `fully_opened`. It is a refinement of the FSM model.
- Line 2-4: The third-level model is a CT model.
- Line 5-9: The up port and its properties are defined from line 5 to line 8. The other two ports, the down port and the *position* port are omitted in line 9.
- Line 10-14: They define the expression entity in the third level and its properties, ports, relations and links.
- Line 15-18: They define the relations and links used in the third-level model.

After building the correct models in Ptolemy II, we transform them into the model in AECPSDesigner. We have mentioned that the first-level model is only for the purpose of simulation. We remove the first-level model and keep

the second-level model (power window model) and the third-level models (power window position model) before performing model transformation. In order to recognize the type of the input and output signals, we add *signalType* parameters to the *up* port and the *down* port with the values of “DISCRETE”, and a *signalType* parameter to the *position* port with the value of “CONTINUOUS”. The process of model transformation consists of five steps.

First, parse the MoML file of the power window. Although directors are entities, they are not viewed as common entities because they are only used to denote MOC. There are four entities in the power window model, *init*, *semi_opened*, *fully_opened*, and *fully_closed*. From the Automobile Knowledge Base, their functions are found and put into Extend Entity table, as shown in Table 3. The properties, relations, ports and links are also parsed. Some properties are not needed, such as the size, position, color, etc. They are not included in Property table.

Table 3. Functions of entities

Entity name	MOC	Function
<i>init</i>	FSM	Body_Window
<i>semi_opened</i>	FSM	Body_Window
<i>fully_opened</i>	FSM	Body_Window
<i>fully_closed</i>	FSM	Body_Window

Second, generate tasks. Because entities of *semi_opened*, *fully_opened*, and *fully_closed* have the same function (Body_Window) and depend on the *up* event and *down* event, they are grouped into a task named *task1*. The entity *init* is also included into *task1* because it has the same function with other entities in *task1*. The other three entities in the third-level model are classified into *task2*.

Third, analyze the interaction among tasks. The state transition information of *task1* is stored in Property table. The position property is stored in Property table of *task2*. The *up* event and *down* event, including the triggering condition and state transition, are stored in Event table of *task1*. The *up* port and *down* port are stored in Port table of *task2*. The *position* port is stored in Port table of *task2*. Because *task2* controls the position of the power window, a resource, *res1*, is created and put into Resource table.

Fourth, optimize tasks. In the Automobile Knowledge Base, the functions that *task1* and *task2* perform are not safety-critical, and have lower priority. *task1* and *task2* are incorporated into one task, *task3*. Their properties, ports, and events are incorporated into a new task. *task1* and *task2* are removed from Task table. The priority is an important property for a task. We assign the priority levels of tasks according to the importance level of their functions.

Fifth, generate the XML file. *task3* are taken out from Task Base. It waits for two events. *task3*'s running information can be generated according to the triggering condition and state transition. *Res1* is the resource it uses. This model is the one under OSEK-compatible OS. Then the model is output to an XML file.

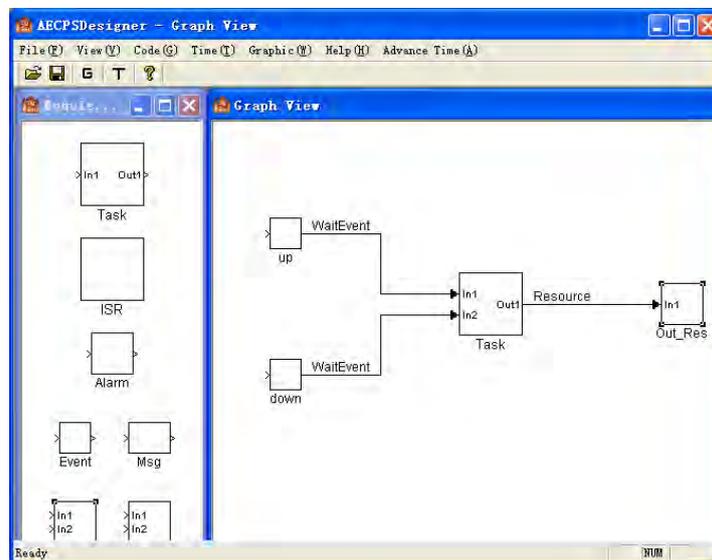


Fig. 14. Models in AECPSDesigner

The pseudo-codes of the generated codes are shown in Figure 15. In Figure 15, *task3* first initializes the position and state of the power window (Line 2-Line 3), and then waits and processes input event (Line 4-Line 28). Once receiving an input event, *task3* obtains a resource in order to access the power window (Line 6), and then changes the position and state of the power window according to the rulers in Table 2 (Line 7-Line 26). Note that the obstacle-detecting is implemented in Line 9-Line 10 and Line 20-Line 23. After that, *task3* updates the state of the power window, sends its position to P2 by CAN bus (Line 25), and release the resource (Line 27).

From the development process of the power window, we can know MoBDAC covers the whole development workflow of software. By separating function requirements, non-functional requirements, and physical environment requirements, developer can concentrate their attention on the function logic of CPS in MPD. The non-functional requirements are analyzed and verified in MID where the characteristics of deployment platform and the execution properties of software are available, and the interaction with physical environment is integrated into implementation models by analyzing relevant events. MoBDAC improves development efficiency by automatic model transformation and code generation, improves software quality by verifying function properties in MPD and non-functional properties in MID, and is easy to integrate the interaction with physical environment.

```
1 TASK(Task){
2   Set initial position of the power window to be zero;
3   Set initial state of the power window to be fully_opened;
4   While (true){
5     Wait for an event;
6     Obtain the resource of the power window;
7     if (The state is fully_opened) {
8       if (The event is an up event)
9         if (No obstacle is detected)
10        Set the position of the power window to be 4.
11    }
12    else if (The state is fully_closed) {
13      if (The event is a down event)
14        Set the position of the power window to be 4.
15    }
16    else {
17      if (The event is a down event)
18        Decrease the position of the power window by 4.
19      else if (The event is an up event)
20        if (No obstacle is detected)
21          Increase the position of the power window by 4.
22        else
23          Decrease the position of the power window by 4.
24    }
25    Send the position of the power window to CAN bus.
26    Update the state of the power window.
27    Release the resource of the power window;
28  }
29  TerminateTask();
}
```

Fig. 15. Codes generated by AECPSDesigner

6. Conclusions

Aiming at the development of automotive CPS software, we present a model-based development method under operating systems compatible with OSEK/VDX specification. This method increases development efficiency by automatic tools, and can verify the correctness of function requirements, non-functional requirements, and integrate the interaction with physical environment. The future work is to integrate more analysis methods for non-functional requirements and verify the interaction with physical environment in MIP.

Acknowledgment. This work has been supported by the Pre-research Scheme of 973 Program of China (Grant No. 2010CB334707), Zhejiang Provincial Key Science and Technology Program (No. 2009C14013), and the Scientific Research Found (No. 0804623-Y, No. KYS055610046, No. KYS055608107).

References

1. Sha, L., Gopalakrishnan, S., Liu, X., Wang, Q.: Cyber-Physical Systems: A New Frontier. In Proceedings of the 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 2008), 1–9. (2008)
2. Shih, E., Bahl, P., Sinclair, M. J.: Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices. In Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom 2002), 160–171. (2002)
3. Yoerger, D. R., Jakuba, M., Bradley, A. M., Bingham, B.: Techniques for Deep Sea Near Bottom Survey Using an Autonomous Underwater Vehicle. *Journal of Robotics Research*, Vol. 26, No. 1, 41–54. (2007)
4. Lee, E. A.: Cyber Physical Systems: Design Challenges. *International Symposium on Oriented Real-Time Distributed Computing (ISORC'08)*, 363-369. (2008)
5. Selic, B., Gullekson, G., Ward, P.: *Real-Time Object-Oriented Modeling*. John Wiley & Sons, New York, NY. (1994)
6. Selic, B., Rumbaugh, J.: Using UML for modeling complex real-time systems. (1998). [Online]. Available: <http://www.objecttime.com/uml>.
7. The SDL Forum website. (2005). [Online]. Available: <http://www.sdl-forum.org>.
8. Sztipanovits, J., Karsai, G.: Model-integrated Computing. *IEEE Computer*, Vol. 30, No. 4, 110-112. (1997)
9. Lee, E. A.: Overview of the Ptolemy Project. (2003). [Online]. Available: <http://Ptolemy.eecs.berkeley.edu/publications/papers/03/overview/>.
10. MoBIES Automotive Open Experimental Platform. (2003). [Online]. Available: <http://vehicle.me.berkeley.edu/mobies/>.
11. Magureanu, G., Gavrilescu, M., Pescaru, D., Doboli, A.: Towards UML Modeling of Cyber-Physical Systems: A Case Study for Gas Distribution, In Proceedings of the 8th International Symposium on Intelligent Systems and Informatics (SISY 2010), 471–476. (2010)
12. Bhatia, G., Lakshmanan, K., Rajkumar, R.; An End-to-End Integration Framework for Automotive Cyber-Physical Systems Using SysWeaver. In Proceedings of the 1st Analytic Virtual Integration of Cyber-Physical Systems Workshop (AVICPS 2010). (2010)
13. OSEK/VDX Binding Specification Version 1.4.1. (2003). [Online]. Available: <http://www.osek-vdx.org/mirror/oil241.pdf>.
14. Ledeczi, A., Maroti, M., Bakay, A., et al.: The Generic Modeling Environment. In Proceedings of the IEEE Workshop on Intelligent Signal Processing (WISP 2001). (2001)
15. The MathWorks, Inc. (2011). [Online]. Available: <http://www.mathworks.com/>.
16. Metacase company. (2011). [Online]. Available: <http://www.metacase.com/>.
17. DOME Users' Guide. (2011). [Online]. Available: <http://www.htc.honeywell.com/dome/support.htm#documentation>.
18. Telelogic inc. (2011). [Online]. Available: <http://modeling.telelogic.com/index.cfm>.
19. Woo, H., Yi, J., Browne, J. C., et al.: Design and Development Methodology for Resilient Cyber-Physical Systems. In Proceedings of the 28th International Conference on Distributed Computing Systems Workshops (ICDCS Workshops 2008), 525-528. (2008)
20. Lin, J., Sedigh, S., Miller, A.: Towards Integrated Simulation of Cyber-Physical Systems: A Case Study on Intelligent Water Distribution. In Proceedings of the

Zhigang Gao, Haixia Xia, and Guojun Dai

- 8th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC 2009), 690-695. (2009)
21. Ma, L., Yuan, T., Xia, F., et al.: A High-confidence Cyber-Physical Alarm System: Design and Implementation. In Proceedings of 2010 IEEE/ACM International Conference on Green Computing and Communications (GreenCom) & International Conference on Cyber, Physical and Social Computing (CPSCom), 516-520. (2010)
 22. Tsai, J. J. P., Li, B., Liu, A.: Modeling and Parallel Evaluation of Non-Functional Requirements Using FRORL Requirements Language. In Proceedings of the 8th Annual International Computer Software and Applications Conference (COMPSAC 1994), 11-16. (1994)
 23. Chung, L., Nixon, B. A., Yu, E., Mylopoulos J.: Non-functional requirements in Software Engineering. Kluwer Academic Publishers. (1999)
 24. Fidge, C.J., Lister, A. M.: The Challenges of Non-Functional Computing Requirements. In Proceedings of the 7th Australian Software Engineering Conference (ASWEC 93). (1993)
 25. Lee, E. A.: Embedded Software. *Advances in Computers*, Vol. 56, 56-97. (2002)
 26. Niz, D. de, Rajkumar, R.: Model-based Embedded Real-time Software Development. (2003). [Online]. Available: <http://www.cse.wust1.edu/~cdgill/RTASO3/published/TimeWeaverPosition.pdf>.
 27. CodeWarrior Development Tools. (2011). [Online]. Available: http://www.freescale.com/webapp/sps/site/homepage.jsp?code=CW_HOME.
 28. Brooks, C., Lee, E. A., Liu, X., Neuendorffer, S., Zhao, Y., Zheng, H.: Heterogeneous Concurrent Modeling and Design in Java (volume 1: Introduction to Ptolemy II). University of California, Technical Report UCB/ERL M05/21. (2005)
 29. Gao, Z., Wu, Z., Lin, M.: Energy-Efficient Fixed-Priority Scheduling for Periodic Real-Time Tasks with Multi-Priority Subtasks. In Proceedings of the 2007 International Conference on Embedded Software and System (ICSS 2007), LNCS, Vol. 4523, 572-583. (2007)
 30. Gao, Z., Wu, Z., Li, H.: Implementation Synthesis of Embedded Software under Operating Systems Supporting the Hybrid Scheduling Model. In Proceedings of the 2006 IFIP International Conference on Embedded and Ubiquitous Computing (EUC 2006), LNCS, Vol. 4096, 426-436. (2006)
 31. Gao, Z., Wu, Z.: Implementation Synthesis of Embedded Software under the Group-Based Scheduling Model. In Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2006), 190-196. (2006)
 32. Wilhelm, R., Engblom, J., Ermedahl, A., et.al.: The Worst-Case Execution-Time Problem—Overview of Methods and Survey of Tools. *ACM Transactions on Embedded Computing Systems (TECS)*, Vol. 7, No. 3, 36:1-36:53. (2008)

Zhigang Gao received the Ph.D. degree from the College of Computer Science, Zhejiang University, Hangzhou, China in 2008. He is a teacher in the College of Computer Science, Hangzhou Dianzi University, Hangzhou, China. His current research interests are pervasive computing, Cyber-Physical Systems, and automotive electronic systems.

Haixia Xia received the Ph.D. degree from Zhejiang University, Hangzhou, China in 2007. She is a teacher in the College of Informatics & Electronics, Zhejiang Sci-Tech University, Hangzhou, China. Her current research interests are pervasive computing and motor systems.

Guojun Dai received the Ph.D. degree from Zhejiang University, Hangzhou, China in 1998. He is a professor in the College of Computer Science, Hangzhou Dianzi University, Hangzhou, China. His current research interests are pervasive computing and intelligent embedded systems.

Received: March 3, 2011; Accepted: April 26, 2011.

Velocity Adaptation for Synchronizing a Mobile Agent Network

Lijing Li, Hui Yan, Hui Li, and Chunxi Zhang

School of Instrumentation Science and Opto-electronics Engineering,
Beihang University, 100191 Beijing, China.
{lilijingbuaa, huiy, lihui, zhangcx}@gmail.com

Abstract. This paper investigates the problem of synchronizing a mobile agent network by means of a velocity adaptation strategy, where each agent is assigned different moving velocities to establish a time-varying network topology, and the velocity of each agent develops adaptively according to the local property between itself and its neighbors. We show that our strategy is effective in enhancing the synchronizability of the mobile agent network, i.e., the region of power density for which the network can achieve synchronization is enlarged as compared to the fast-switching case. In addition, the influence of the controlling parameter on network evolution is studied by assessing the convergence time.

Keywords: mobile agents, velocity adaptation, complex coupled networks, synchronization.

1. Introduction

In the past decade, network-based approaches have attracted an increasing interest and have been proved to be prominent candidates to investigate the collective dynamics in many branches of science and engineering [1], [2], [3], [4], [5]. As a typical collective motion, synchronization in complex dynamical networks has been extensively studied. And a number of studies have pointed out that topological structure plays a significant role in the formation of network synchronization [6], [7], [8].

However, most investigations have focused on networks that do not change with the dynamics, i.e., network topological structures and weights are fixed as time evolves. As a matter of fact, a great deal of real-world complex networks in biological, social and communication systems exhibit time-varying topological structures: edges are deleted, added or rewired according to some rules. Therefore, many researchers have recently devoted their attention to the study of time-varying networks [9], [10], [11], [12], [13], [14], [15], [16], [17], particularly to the co-evolution of dynamical states and network structures. As a result, lots of models of adaptive networks have

been proposed and the corresponding synchronization issues have been intensively investigated [18], [19], [20], [21].

In this paper, we focus our discussions on the synchronization issue in a power-driven mobile agent network [22]. Since this type of agent network has a remarkable feature of switching topology, it could be used as a good representation to explore many kinds of real-world problems, e.g., synchronous motion in clock of mobile robots [23], bulk oscillations of yeast cells [24], and collective reaction of a group of animals [25]. We propose a velocity adaptive rule to guarantee its synchronous behavior based on the moving agent network with heterogeneous moving velocities. The general idea behind the adaptation strategy is that, each agent develops its moving velocity according to local synchronization property between itself and its neighbors. In the following sections, we will show that when power density is large, synchronization of the considered network can be effectively guaranteed under the velocity adaptation strategy. In addition, by assessing the convergence time, we investigate the influence of controlling parameter on network synchronization.

The rest of this paper is organized as follows: Related work is presented in Section 2. In Section 3, a moving agent network model with velocity adaptation strategy is proposed. And synchronization analysis of the considered network is given in Section 4. Further discussions including the Influence of controlling parameter are shown in Section 5. Conclusions are finally drawn in Section 6.

2. Related Work

Mobile agent network seems to be a good solution to investigate various synchronization problems in different complex systems. Generally speaking, each agent in the agent network is equipped with an identical oscillator, and switching topology is constructed via the change of neighbouring interactions. The mobile agent network, indeed, can be used to explore many problems such as clock synchronization in mobile robots, swarming animals or the appearance of synchronized bulk oscillations, consensus problem in multi-agent systems and so on, partially because of a good choice to capture jumps or switches of coupling evolutions.

Frasca et al investigated the fast-switching synchronization of a moving agent network and pointed out that the density of mobile agents determines network synchronization [13]. Shi et al developed the mobile agent network model by assigning different emission powers to each agent, which further provides an insight into the collective behaviours of coupled agent systems [19], [21], [22]. To enhance network synchronizability, Wang et al proposed an power-adaptation rule to synchronize the power-driven mobile agent network model [27]. Also, Wang et al introduced pinning control strategy (apply localized feedback control to small fraction of the network model) to regulate the mobile agent network [28].

Based on the moving agent network with heterogeneous moving velocities, the paper presents a velocity adaptive rule to guarantee its synchronous behavior. Different from the existing agent network models [19], [21], [22], [26], [27], [28], we assign each agent different moving velocities, so as to establish a time-varying coupling mechanism, which can be used to characterize the heterogeneous moving capabilities of the mobile agents. It has been found that the heterogeneous moving capabilities widely exist in individuals of real-world systems. For instance, insects or animals fly or run with different velocities [29]; treated as agents, pedestrians and automobiles apparently show different moving abilities [30] and so on.

3. A Moving Agent Network Model with Velocity Adaptation Strategy

3.1. Moving Agent Network with Heterogeneous Velocities

Consider a set of N agents, each of which is equipped with a chaotic oscillator $x_i(t) \in \mathbb{R}^n$, $i=1,2, \dots, N$. Assume that all the agents move randomly in a two-dimensional space of size L with periodic boundary conditions, and their positions and orientations are updated according to:

$$y_i(t+\Delta t) = y_i(t) + v_i(t) \Delta t \quad (1)$$

$$\theta_i(t+\Delta t) = \eta_i(t+\Delta t) .$$

where $y_i(t)$ is the position of agent i in the plane at time t , $v_i(t)$ with modulus $V_i(t)$ and direction angle $\theta_i(t)$ is the moving velocity of agent i , $\eta_i(t)$, $i=1,2, \dots, N$, are N independent random variables chosen at each time unit with uniform probability in the interval $[-\pi, \pi]$, and Δt is the time unit. It is noted that, differing from the existing mobile rules [19], [21], [22], [26], [27], [28], the velocity modulus $V_i(t)$ of agents are generally different from each other.

First of all, we recall the power-driven mechanism [22] to establish connections among the mobile agents. In detail, each agent is regarded to be a wave source with emission power P_e^i , which reads:

$$P_e^i = 4\pi d^2 s^i(d), s^i(d) \geq S_c. \quad (2)$$

where d indicates the distance from agent i , $S^i(d)$ as a function of d is the intensity of wave emitted by agent i , and S_c is a critical wave intensity, if the intensity of wave is beyond S_c , then agents can perceive it accurately. If agent i has emission power P_e^i , then there exists an influence radius, denoted by $R_i = (P_e^i / 4\pi S_c)^{1/2}$, within which $S^i(d) \geq S_c$. That is, directed couplings from

agent i to its neighboring agents will be established immediately when the neighboring ones move into its influence range. For the sake of simplicity, we here consider that each agent is of the same emission power, denoted by $P_e^1 = P_e^2 = \dots = P$.

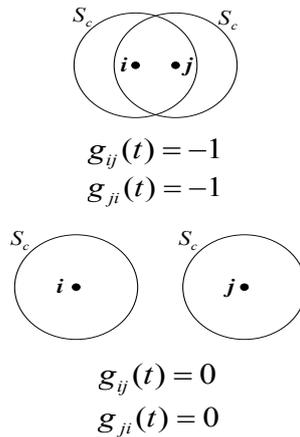


Fig. 1. Coupling relationship between agents i and j , where the circle means the influence area.

Hence, we construct a time-varying dynamical network by combining the mobile agents, chaotic oscillators and their power-driven coupling rules. Moreover, the mobile agent network can be formulated as follows:

$$\dot{x}_i = f(x_i) - \sigma \sum_{j=1}^N g_{ij}(t) h(x_j). \tag{3}$$

where $i=1,2, \dots, N$, $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ governs the local dynamics of oscillator, $h: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a vectorial output function, $\sigma > 0$ is the coupling strength, and Laplacian matrix $G(t) = [g_{ij}(t)] \in \mathbb{R}^{N \times N}$ defines the coupling relationship of agents at a given time t . For specific details, for two agents i and j , if the relative distance between i and j is larger than the influence radius $R = (P/4\pi S_c)^{1/2}$, then $g_{ij}(t) = g_{ji}(t) = -1$, otherwise $g_{ij}(t) = g_{ji}(t) = 0$. Fig. 1 shows the detailed coupling relationship between two agents i and j . The diagonal elements of $G(t)$ satisfy: $g_{ii}(t) = m_i(t)$, where $m_i(t)$ is the number of neighbors of agent i at time t . In this paper, without lack of generality, we consider the *Rössler* oscillator, and we choose $h(x_j) = Hx_j$ with $H = \text{diag}(1, 0, 1)$.

3.2. Velocity Adaptive Strategy of the Moving Agent Network

We here introduce a simple scheme of velocity adaptation according to local synchronization property to synchronize the moving agent network (3). To achieve a global synchronization of network (3), we suppose that each agent develops its moving velocity if its state has a deviation from the mean state

of its neighbors. Specifically, the velocity modulus $V_i(t)$ of agent performs according to the following law using the information of all its neighbors, i.e.,

$$\dot{V}_i(t) = \frac{\gamma \Delta_i(t)}{1 + \Delta_i(t)} \quad (4)$$

$$\Delta_i(t) = \left\| \sum_{j=1}^N g_{ij}(t) x_j(t) \right\|^2$$

where the derivative of $V_i(t)$ is to suppress its difference Δ_i from the mean activity of its neighbors, $\gamma > 0$ is a controlling parameter which is used to adjust the convergence rate.

It is worth noting that the mathematical expression of the velocity adaptive strategy seems similar to that in [18], [19], [20], [21]. However, the velocity adaptation strategy is quite different from those in [18], [19], [20], [21] in principle. The basic idea of adaptation in Ref [18] and Ref [20] is to strengthen the coupling weights of edges; the fundamental principle in Ref [19] is to establish more edges among nodes, and the idea in Ref [21] is to adjust the blinking pace so as to achieve a proper blinking mode that can facilitate synchronization of the network; while the essence of the velocity adaptation in Eq. (4) is to accelerate the rate of information exchange by increasing the moving velocity.

4. Synchronization Analysis of the Moving Agent Network under the Velocity Adaptive Strategy

We first briefly consider the case that all the agents move with sufficiently high velocities. Under such case, the topology of network (3) switches among each possible configuration with sufficiently high speed. Then, according to the results in [31], we can derive that, if the following time-average network achieves synchronization

$$\dot{x}_i = f(x_i) - \sigma \sum_{j=1}^N g'_{ij} h(x_j), \quad (5)$$

then network (3) with time-varying topologies can realize synchronization, where $g'_{ij}(t)$ is the element of the average Laplacian matrix $G' = \int G(s) / T_b ds$, and time window T_b is a constant. Apparently, by recalling the master-stability-function (MSF) method [6], we can deduce that network (5) achieves synchronization if all the non-zero eigenvalues of G' locate in the interval $[\alpha_1/\sigma, \alpha_2/\sigma]$, where α_1 and α_2 are constants determined by MSF corresponding to network (5).

By elementary transformation, we derive the N eigenvalues of G' : $\lambda_1=0$, $\lambda_2=\dots=\lambda_N=NP/(4S_cL^2)$. Thus, under the case of sufficiently high moving velocity $V_i(t)$, network (3) is synchronizable when the power density of agents

lies in the bounded region $[\rho_{e1}, \rho_{e2}]$, where $\rho_{e1}=4S_c\alpha_1/\sigma$, $\rho_{e2}=4S_c\alpha_2/\sigma$. This result indicates that, under the case of sufficiently high moving velocity $V_i(t)$, regardless of the network size, synchronization of network (3) can be achieved when the power density satisfies the above condition.

4.1. Adaptive Synchronization under the Case that Power Density Locates in the Interval $[\rho_{e1}, \rho_{e2}]$

As moving velocities $V_i(t)$ of the agents evolve according to adaptive law (4), we consider the case of power density locating in the interval $[\rho_{e1}, \rho_{e2}]$, where synchronization of network (3) can be guaranteed if all the agents move with sufficiently high moving velocities. Suppose each agent starts from a random small moving velocity. In this case, synchronization of network (3) can be always guaranteed by adaptive law (4). We can use the method of proof by contradiction to explain. If network (3) evolves sufficiently long time without synchronization, then, according to adaptive law (4), the moving velocity $V_i(t)$ of each agent develops gradually and finally achieves sufficiently high. Thus the topology switching will be sufficiently fast under sufficiently high moving velocities, which results in convergence of network (3) as power density locates in the interval $[\rho_{e1}, \rho_{e2}]$.

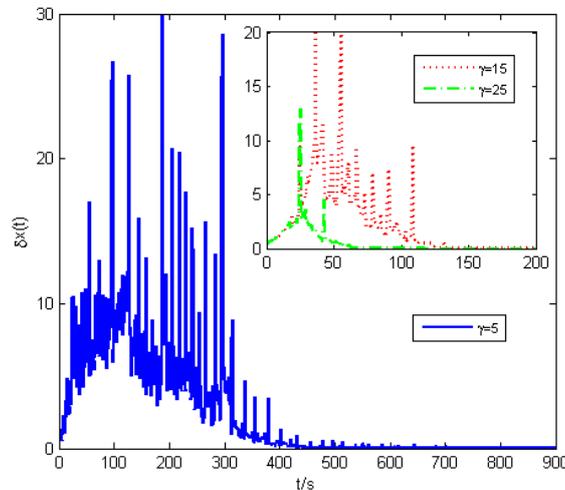


Fig. 2. Evolution of synchronization error $\delta x(t)$ for the case that power density locates in the interval $[\rho_{e1}, \rho_{e2}]$

Fig. 2 gives the corresponding simulation of network (3) under adaptive law (4), where the synchronization error is defined as $\delta x(t) = (\sum_i |x_i - x_1|) / N$, network size $N=100$, power density $\rho_{e1} < \rho_e = 0.2 < \rho_{e2}$, the other parameters are set as: $\sigma=10$, $P=3.14$, $S_c=0.25$, and $\Delta t=10^{-3}s$. It is easy to see that network (3)

under adaptive law (4) can achieve synchronization for different controlling parameter γ and the main trends of synchronization error $\delta x(t)$ perform almost the same though differences exist in γ : $\delta x(t)$ increases with t at the beginning since the initial moving velocity of each agent is small and the whole network can be regarded as an unconnected one, thus the network has no propensity to synchronization; as time evolves, moving velocities $V_i(t)$ develop gradually under adaptive law (4) and switching among all possible topological configurations becomes more and more fast; furthermore, $\delta x(t)$ is prone to decrease, and finally converges to zero, which means a complete synchronization of network (3). Also notice in Fig. 2 that enhancing γ will speed up the convergence rate.

4.2. Adaptive Synchronization under the case that Power Density is Larger than ρ_{e2}

In the following, we consider the case that power density of network (3) is larger than the critical value ρ_{e2} . According to the results derived under the fast-switching condition, network (3) cannot achieve synchronization in such a case if the moving velocity of each agent is sufficiently high. While numerical simulations show that synchronization of network (3) can still be guaranteed by velocity adaptive law (4).

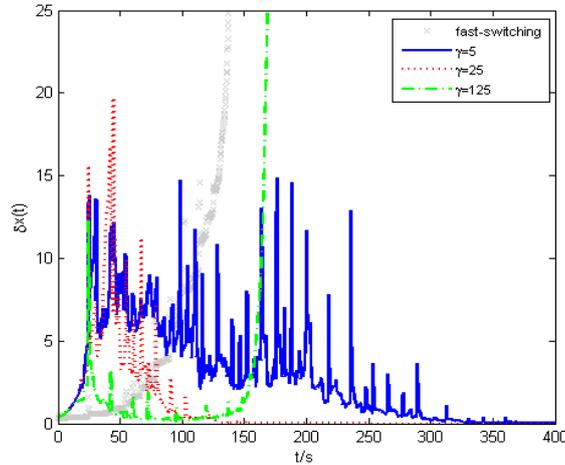


Fig. 3. Evolution of synchronization error $\delta x(t)$ for the case that power density is larger than ρ_{e2}

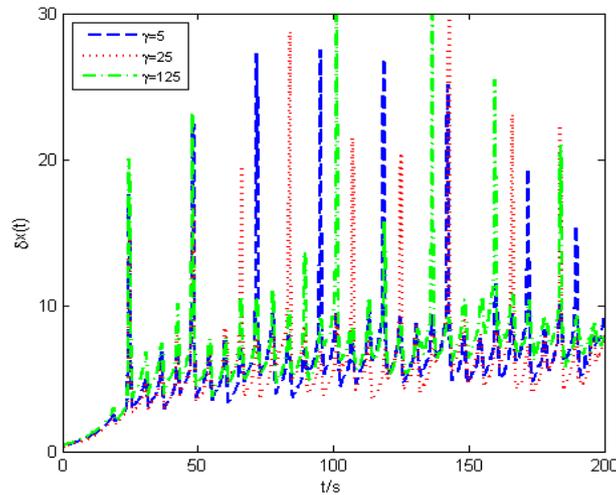


Fig. 4. Evolution of synchronization error $\delta x(t)$ for the case that power density is smaller than ρ_{e1} .

Fig. 3 gives the result of synchronization error $\delta x(t)$ evolving with time t under fast-switching case and velocity adaptive law (4) respectively, where network size $N=100$, power density $\rho_e=0.53 > \rho_{e2}$, and the other parameters are set as: $\sigma=10$, $P=3.14$, $S_c=0.25$, and $\Delta t=10^{-3}$ s. It can be seen from Fig. 3 that synchronization error $\delta x(t)$ increases with time t when all the agents in network (3) move with sufficiently high speed, i.e., fast-switching condition is satisfied, which means network (3) cannot achieve synchronization. When network (3) evolves according to velocity adaptive law (4), synchronization error $\delta x(t)$ will converge to zero if the controlling parameter γ is relatively small ($\gamma=5$ and $\gamma=25$), which means network (3) achieves complete synchronization. While, if the controlling parameter γ is relatively large ($\gamma=125$), synchronization error $\delta x(t)$ cannot realize convergence, i.e., network (3) cannot achieve complete synchronization. This phenomenon can be explained as follows: if controlling parameter γ is sufficiently large, then moving velocity $V_i(t)$ of each agent will develop sufficiently fast. Thus, moving velocities $V_i(t)$ become sufficiently high before complete synchronization of network (3) is achieved; furthermore, fast-switching condition is satisfied. While network (3) cannot realize synchronization under fast-switching condition if power density is larger than ρ_{e2} . In conclusion, network (3) cannot realize synchronization by velocity adaptive law (4) if controlling parameter γ is relatively large. Therefore, we can choose a relatively small γ to guarantee synchronization of network (3).

4.3. Adaptive Synchronization under the Case that Power Density is Smaller than ρ_{e1}

For the case that power density is smaller than ρ_{e1} , we conduct simulations to see how network (3) evolves under velocity adaptive law (4). Fig. 4 give the result of synchronization error $\delta x(t)$ evolving with time t under different controlling parameter γ , where network size $N=100$, power density $\rho_e=0.01 < \rho_{e1}$, and the other parameters are set as: $\sigma=10$, $P=3.14$, $S_c=0.25$, and $\Delta t=10^{-3}$ s. From Fig. 4, we can see that no matter what value of controlling parameter γ is, synchronization error $\delta x(t)$ cannot converge to zero as time evolves, i.e., network (3) cannot achieve synchronization under velocity adaptive law (4) when power density is smaller than ρ_{e1} . That is to say, velocity adaptive law (4) works little in enhancing synchronizability of network (3) when power density is smaller than ρ_{e1} .

5. The Influence of Controlling Parameter γ on Adaptive Synchronization of the Agent Network

The discussions above show that controlling parameter γ plays a significant role in making network (3) realize synchronization. Here, we conduct simulations and give the results of synchronization index $\langle \delta x \rangle = \langle \delta x(t) \rangle$ evolving with power density ρ_e under velocity adaptive law (4) with different controlling parameter γ in Fig. 5, where synchronization index $\langle \delta x \rangle$ is the average of $\delta x(t)$ during the period $[T, T+\Delta T]$. The other parameters in simulations are set as: $T=200$ s, $\Delta T=50$ s, $\sigma=10$, $P=3.14$, $S_c=0.25$, and $\Delta t=10^{-3}$ s. It can be seen from Fig. 5 that, network (3) under velocity adaptive law (4) possesses a relatively large power density region as compared to the fast-switching case, in which synchronization of network (3) can be realized. Velocity adaptive law (4) broadens the upper bound of the power density region, while it works little in broadening the lower bound. It can be noticed that, as controlling parameter γ decreases, the upper bound of the power density region increases, which indicates that lowering γ is in favor of enhancing the synchronizability of network (3).

However, lowering γ also results in some poor performance indexes. Fig. 6 gives the result of convergence time T_c evolving with controlling parameter γ , where the convergence time T_c is defined to be the total time from the beginning to the moment that full synchronization of network (3) is achieved [32] (in simulations, when $\delta x(t) < 10^{-4}$, network (3) is regarded to realize synchronization). The other parameters in simulations are set as: $N=100$, $\sigma=10$, $P=3.14$, $S_c=0.25$, and $\Delta t=10^{-3}$ s. From Fig. 6, convergence time T_c is a finite value for a particular γ , which means network (3) can realize synchronization. It is noted that, when γ is small, convergence time T_c is considerably long, which means poor performance index. Therefore, we

should choose a proper controlling parameter γ to meet system requirement, not being too large or too small.

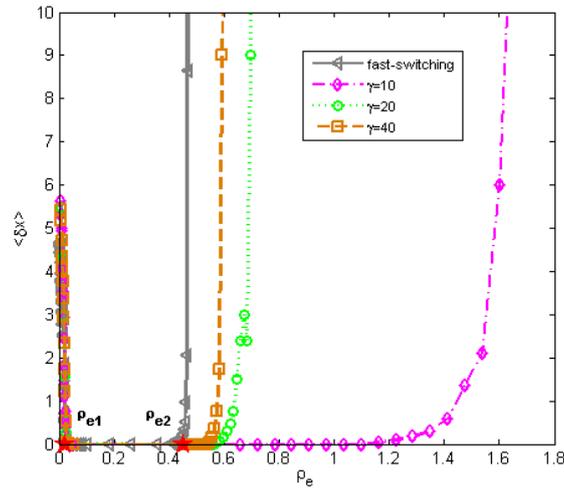


Fig. 5. Evolution of synchronization index $\langle \delta x \rangle$ with power density ρ_e for different controlling parameter γ

6. Conclusion

In conclusion, different moving velocities are introduced for the mobile agent network to characterize the heterogeneous moving capability of each agent. For such an agent network with heterogeneous moving velocities, we suggest a velocity adaptive strategy in order to guarantee the corresponding synchronization motion. Theoretical analysis and numerical simulations have shown that the proposed velocity adaptive tactic is quite efficient in enhancing the synchronizability of the considered network. In addition, we also discussed the impact of controlling parameter on network synchronization by assessing the convergence time. All these investigations may provide some insights for the future research work on synchronization enhancement in coupled oscillator networks and also may open up new possibility to design potential engineering applications, e.g., mobile sensor networks, moving robotics systems, as well as unmanned aerial vehicles.

References

1. Watts, D.J., Strogatz, S.H.: Collective dynamics of “small world” networks. *Nature*, Vol. 393, 440-442. (1998)

2. Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. *Science*, Vol. 286, 509-512. (1999)
3. Boccaletti, S., Kurths, J., Osipov, G., Valladares, D.L., Zhou, C.S.: The synchronization of chaotic systems. *Physics Reports*, Vol. 366, No. (1-2), 1-101. (2002)
4. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.-U.: Complex networks: structure and dynamics. *Physics Reports*, Vol. 424, No. 4-5, 175-308. (2006)
5. Barabási, A.-L.: Scale-free networks: a decade and beyond. *Science*, Vol. 325, No. 1, 412-413. (2009)
6. Pecora, L.M., Carroll, T.L.: Master stability functions for synchronized coupled systems. *Physical Review Letter*, Vol. 80, No. 1, 2109-2112. (1998)
7. Wen, G., Wang, Q.G., Lin, C., Li, G.Y., Han, X.: Chaos synchronization via multivariable pid control. *Physics Reports*, Vol. 17, No. 5, 1753-1758. (2007)
8. Arenas, A., Diaz-Guilera, A., Kurths, J., Moreno, Y., Zhou, C.S.: Synchronization in complex networks. *Physics Reports*, Vol. 469, No. 3, 93-153. (2008)
9. Belykh, I.V., Belykh, V.N., Hasler, M.: Blinking model and synchronization in small-world networks with a time-varying coupling. *Physica D*, Vol. 195, No. 1-2, 188-206. (2004)
10. Belykh, V.N., Belykh, I.V., Hasler, M.: Connection graph stability method for synchronized coupled chaotic systems. *Physica D*, Vol. 195, No. 1-2, 159-187. (2004)
11. Skufca, J.D., Bollt, E.M.: Communication and synchronization in disconnected networks with dynamic topology: moving neighborhood networks. *Mathematical Biosciences and Engineering*, Vol. 1, No. 2, 347-359. (2004)
12. Lu, J.H., Chen, G.R.: A time-varying complex dynamical network model and its controlled synchronization criteria. *IEEE Transactions on Automatic Control*, Vol. 50, No. 6, 841-846. (2005)
13. Frasca, M., Buscarino, A., Rizzo, A., Fortuna, L., Boccaletti, S.: Dynamical network model of infective mobile agents. *Physical Review E*, Vol. 74, No. 3, 036110. (2006)
14. Chen, M.Y.: Synchronization in time-varying networks: a matrix measure approach. *Physical Review E*, Vol. 76, No. 1, 016104. (2007)
15. Wu, X.Q.: Synchronization-based topology identification of weighted general complex dynamical networks with time-varying coupling delay. *Physica A*, Vol. 387, No. 4, 997-1008. (2008)
16. Qin, S.M., Zhang, G.Y., Chen, Y.: Coevolution of game and network structure with adjustable linking. *Physica A*, Vol. 388, No. 23, 4893-4900. (2009)
17. Zhang, X.S., Chen, T., Zheng, J., Li, H.: Proactive worm propagation modeling and analysis in unstructured peer-to-peer networks. *Journal of Zhejiang University SCIENCE C*, Vol. 11, No. 2, 119-129. (2010)
18. Zhou, C.S., Kurths, J.: Dynamical weights and enhanced synchronizability in adaptive complex networks. *Physical Review Letter*, Vol. 96, No. 16, 164102. (2006)
19. Wang, L., Shi, H., Sun, Y.X.: Power adaptation for a mobile agent network. *Europhysics Letters*, Vol. 90, No. 1, 10001. (2010)
20. Zhu, J.F., Zhao, M., Yu, W.W., Zhou, C.S., Wang, B.H.: Better synchronizability in generalized adaptive networks. *Physical Review E*, Vol. 81, No. 2, 206201. (2010)

Lijing Li, Hui Yan, Hui Li, and Chunxi Zhang

21. Shi, H., Dai, H.P., Sun, Y.X.: Blinking Adaptation for Synchronizing a Mobile Agent Network. *Journal of Zhejiang University SCIENCE C*, in press, DOI: 10.1631/jzus.C1000338
22. Shi, H., Wang, L., Dai, H.P., Sun, Y.X.: Synchronization in a power-driven moving agent network. *Physica A*, Vol. 389, No. 16, 3094-3100. (2010)
23. Buscarino, A., Fortuna, L., M., Frasca, Rizzo, A.: Dynamical network interactions in distributed control of robots. *Chaos*, Vol. 16, No. 1, 015116. (2006)
24. Dano, S., Sorensen P.G. Hynne F.: Sustained oscillations in living cells. *Science*, Vol. 402, 320-322. (1999)
25. Peng, L.Q., Zhao, Y., Tian, B.M., Zhang, J., Wang, B.H., Zhang, H.T., Zhou, T.: Consensus of self-driven agents with avoidance of collisions. *Physical Review E*, Vol. 79, No. 2, 026113. (2009)
26. Frasca, M., Buscarino, A., Rizzo, A., Fortuna, L., Boccaletti, S., 2008. synchronization of moving chaotic agents. *Physical Review Letter*, Vol. 100, No. 4, 044102. (2008)
27. Wang, L., Shi, H., Sun, Y.X.: Induced synchronization of a mobile agent network by phase locking. *Physical Review E*, Vol. 82, No. 4, 046222. (2010)
28. Wang, L., Sun, Y.X.: Pinning synchronization of a mobile agent network. *Journal of Statistical Mechanics: Theory and Experiment*, Vol. 2009, No. P11005. (2009)
29. Sara, M.L., Christopher, K.C.: Flash signal evolution, mate choice, and predation in fireflies. *Annual Review of Entomology*, Vol. 53, 293-321. (2008)
30. Brockmann, D., L. Hufnagel, L., Geisel, T.: The scaling laws of human travel, *Nature*, Vol. 439, 462. (2006)
31. Stilwell, D.J., Bollt, E.M., Roberson, D.G.: Sufficient conditions for fast switching synchronization in time-varying network topologies. *SIAM Journal on Applied Dynamical Systems*, Vol. 5, No. 1, 140-156. (2006)
32. Sundararaman, B., Buy, U., Kshemkalyani, A.D.: Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, Vol. 3, No. 3, 281-323. (2005)

Lijing Li received the Ph.D degree in precision instrument and opto-electronics engineering from Tianjin University, Tianjin, China, in 2002. He is currently an Associate Professor of Optical Engineering, Beihang University. His research interests include complex networks, optical fiber sensing technique, measurement information processing.

Hui Yang was born in Hunan, China. She received the B.S degree in Electronic and Information Engineering from Northwest A&F University, Xi'an, China, in 2010. She is currently a graduate student of Optical Engineering in Beihang University, Beijing, China. Her research interests include complex networks, optical fiber sensing technique.

Hui Li received the Ph.D degree in Precision Instrument and Machinery from Beihang University, Beijing, China, in 2009. She is currently an Assistant Professor of Optical Engineering, Beihang University. Her research interests include cooperative control, switching control and robust control.

Chunxi Zhang received the M.S. degree from Ordnance Engineering College, Shijiazhuang, China, in 1989, and the Ph.D degree in fiber optic gyroscope technology and application from Zhejiang University, Hangzhou, China, in 1996. He is currently a Professor of Instrumentation Science and Opto-electronics Engineering, Beihang University. His main research interests include optoelectronic devices, optoelectronic measurement and optical fiber sensing technique.

Received: March 15, 2011; Accepted: May 4, 2011.

A Case Study on REST-Style Architecture for Cyber-Physical Systems: Restful Smart Gateway

Qiang Li^{1,2}, Weijun Qin¹, Bing Han³, Ruicong Wang³, and Limin Sun¹

¹ Institute of Software, the Chinese Academy of Sciences,
Beijing 10090, China
{liqiang, qinweijun, sunlimin}@is.iscas.ac.cn

² Graduate University of Chinese Academy of Sciences,
Beijing 10090, China
{liqiang}@is.iscas.ac.cn

³ School of Software and Microelectronics, Beijing,
102600 Beijing, China
{hanbing, wangruicong}@is.iscas.ac.cn

Abstract. Due to several key factors, Cyber-physical systems (CPS) pose great challenges in software system design, which are dynamic composition, heterogeneous, adaptation and uncertain in environmental factors. In this paper we present our research on the development of REST-style architecture for CPS. We propose a path towards solving requirements of CPS architecture through Restful principles. By using this architectural style, we have built a prototyping system called the restful smart gateway, which seamlessly integrates conceptual and physical resources into the Web and scale better. Some experiments on the smart gateway are given to illustrate its performance.

Keywords: cyber-physical systems, REST, architecture, smart gateway, wireless sensor network.

1. Introduction

Cyber-physical systems (CPS) [2] are physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core. CPS bridges the virtual world of computing and communications and the real world, which lead to enormous societal impact and economic benefit. CPS pose great challenges [1] in software system design, due to several key factors: (1) physical devices expose their functionalities to the outside with networking capability, leading to the complexity of the system increases exponentially (2) the cyber and physical worlds are tightly integrated, which requires systems should tolerate the failures of components and uncertainty or the noise in the physical environment, and (3) components are inherently heterogeneous. Given

above-mentioned challenges designing CPS, one system design solution will not meet the requirements. Rather, what we need is a basic system software architecture [3] upon which services can easily be designed, deployed and composed on demand by individual applications, in a manner that satisfies specific safety, security, reliability, efficiency and predictability requirements, while still remaining within the bounds of given hardware capabilities.

In this paper we propose to use REST-Style architecture [4] as design principles, to guide developing of CPS. The Representational State Transfer (REST) style provides a set of architectural constraints that emphasizes scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems. As demonstrated by the success of the Web (using REST-style architecture guide), loosely coupled approaches possess high scalability and robustness - which are fundamental properties for building a worldwide network of devices. Furthermore, the real value of such applications comes from the sharing and integration of data among heterogeneous devices. Based on these considerations, our proposal is about CPS designing and developing under the guiding of REST-Style architecture, which seems to be the primary choice of system design is to ensure reliable, safe, efficient and predictable behavior of the applications.

The contribution of this paper is given as follows. On one hand it proposes to use REST-Style architecture as the basis for systems developing that meet the requirements for CPS architecture. On the other hand it proposes a case study of the smart gateway, which is lightweight, scalability and extensible software components that enable Web-based interactions with all kinds of embedded devices. For CPS, the role of smart gateways simplifies greatly the process to export data and functionality of the physical devices to be provided to end users on the web. Using our prototype system to illustrate that REST-Style architecture is suitable for CPS.

In this paper, we present REST-Style architecture as an approach to design and developing CPS. In section 2, we present a survey of related works. In section 3, we extracted from the existing work the requirements for CPS and described the main characteristic of REST-Style architecture. In section 4, we present the implementation of prototype system, while Section 5 deals with a performance evaluation of the whole infrastructure. Finally, Section 6 discusses future work and concludes the paper.

2. Related Work

Lee [1] examines the challenges in designing CPS by considering whether current computing and networking technologies can support CPS design. The CPS Steering Group [2] present a complete description of issues related to CPS. In the work [3], Raj et al. further elaborate a multitude of technical challenges about the design, construction and verification of CPS, which

must be addressed by a cross-disciplinary community of researchers and educators. Most of the works on CPS raise design challenges or design issues such as dynamic composition, and high confidence software design [12-15]. Ying Tan [6, 7] has employed CPS architecture. These works define control components which determine and provides necessary functionality, and propose spatio-temporal event model. Abdelwahed and his colleagues [8] present an approach to designing high confidence software for CPS. They present four design methods; model-driven system execution modeling, model-based diagnosis, online control, fault-adaptive control, and an architecture for CPS. These are the motivations for our research, which try to gain a general architecture for CPS.

Differs from all of these architectures about CPS, our work profoundly is influenced by the concept of the REST principles, which guide designing and developing of CPS. The notion of REST has been conceptualized in Roy Fielding's PhD thesis [4]. Erik Wilde [5] proposes to put things to REST, which combine today's Web content and the integration of physical things. This is one of the first projects that envision the notion of the Web of Things. Following this direction, Stirbu [20] also enables heterogeneous sensor devices in the Web, but it focuses mainly on the discovery of these devices. TinyREST [19] offers a Restful gateway that has some similarities to our implementation but it violates REST principles by introducing the extra verb SUBSCRIBES. In addition, no evaluation of the system is provided. The work [9] is also about the REST-Style architecture, but only to remote control of sensors over a Wireless Sensor Network and the Internet. The work [10, 11] is similar to ours; develop an open web-based architecture for controlling the behavior of systems composed of static sensors, mobile sensors and unmanned vehicles.

Unlike these previous works, we implement a prototype system about the restful smart gateway, which handles HTTP requests and thus abandons the idea of having an HTTP server directly on the device. The smart gateway is lightweight, scalability and extensible software components that enable Web-based interactions with all kinds of embedded devices.

3. REST-Style architecture for CPS

3.1. Requirements for CPS Architecture

Most of the works on CPS raise design challenges or design issues such as dynamic composition, and high confidence software design [12-15]. In particular, we envision four distinct challenges in design systems that cover the most important use cases that should be supported by a suitable architecture:

- Composition. Cyber-physical systems are inherently heterogeneous not

only in terms of their components but also in terms of essential design requirements. This heterogeneity will lead to that systems don't behave well outside of a small operational envelope and that are hard to maintain.

- Systems integration. Since physical processes are coupling with computing and communication processes, system components designing becomes complex. Simply using current technology, it is difficult in integrating complex components into complex systems.
- Reliability. Since physical devices interface with the uncertainty and the noise in the physical environment, systems must tolerate or contain the failures of components in both the cyber and physical domains.
- Security and Privacy. Cyber-physical systems open up new threats: physical systems can now be attacked through cyberspace and cyberspace can be attacked through physical devices.

3.2. REST-Style Architecture

Based on existing solutions for CPS architectures (Section 2), along with the requirements for Cyber-physical systems (Section 3.1), we suggest using REST-style architecture, which seems to be the primary choice of system design to ensure reliable, safe, efficient and predictable behavior of the applications.

The Representational State Transfer (REST) [4] style is an abstraction of the architectural elements, providing a set of architectural constraints that emphasizes scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems. The REST community has been working on refining the notions to create Resource Oriented Architectures (ROA), in order to provide and connect together services on the Web. Because of the underlying simplicity, scalability of this architecture, we believe that they could be adapted in order to interconnect the physical world. In particular, we envision three major points of REST:

- Data Elements. A resource is a key abstract concept in REST that is assigned a URI and then can be used on the systems. Any information that can be a resource: a document or image, even physical device. A resource representation is a sequence of bytes, plus representation metadata to describe those bytes. REST components perform actions on a resource by using a representation to capture the current or intended state of that resource and transferring that representation between components.
- Connectors. REST connectors provide a generic interface for accessing and manipulating the value set of a resource, regardless of how the membership function is defined or the type of software that is handling the request. REST components communicate by transferring a representation of a resource in a format matching one of an evolving set

of standard data types, selected dynamically based on the capabilities or desires of the recipient and the nature of the resource. Whether the representation is in the same format as the raw source, or is derived from the source, remains hidden behind the interface. The connectors present an abstract interface for component communication, enhancing simplicity by providing a clean separation of concerns and hiding the underlying implementation of resources and communication mechanisms.

- **Components.** The vision is to design a system as a collection of application-specific services and abstractions, which are automatically structured and deployed on a target platform according to constraints in terms of: (a) hardware capabilities, and (b) application requirements.

In conclusion, the central feature of REST architectural style is its emphasis on a uniform interface between components. By applying the engineering principle of generality to the component interface, the overall system architecture is simplified and the visibility of interactions is improved. Implementations are decoupled from the services they provide, which encourages independent evolution. However, the trade-off is that a general interface degrades efficiency, since information is transferred in a standardized form rather than one which is specific to an application's requirements.

4. A Case Study: Restful Smart Gateway

Here we describe how we implemented our smart gateway to illustrate that REST-style architecture is suitable for CPS. Due to the fact that most of the embedded devices that are used today to represent physical things do not offer TCP/IP networking by default but dedicated communication protocols and technologies, we develop a smart gateway, from the Web to the physical devices and vice-versa. We present in detail our implementation efforts in the following subsections by utilizing our framework.

4.1. Smart Gateway Architecture

The conceptual overview of the smart gateway architecture is shown in Fig. 1. Presentation module generates dynamically a representation of the available devices and their corresponding services to the Web, enabling the uniform interaction with them over a Restful interface. Device management is responsible for the management and control of embedded devices, including discovery devices, maintain device adding and delete devices. Data management incorporate data and command parse modules, which convert and parse data among different dedicated communication protocols.

Presentation module represents the access point to the smart gateway from the Web. A Web Server allows clients to interact with any available

devices and their corresponding services using any Web browser or the client. Physical devices and services can be accessible by clicking URL links. We implement the interaction with any resource through a Restful API, which is provided by a REST Engine [16]. Restlet [16] was used to implement the REST Engine, as it has very stable performance in Java environments. For resource presentations, we implement two kinds' types: XML and JSON.

Device management is responsible for the management and control of embedded devices by maintaining a device list. Device management maintains device by broadcasting periodically a message. If an embedded device appears in the scale of the smart gateway, it will attempt to connect to our smart gateway by sending a "HELLO" message. As soon as this message is received by the smart gateway, it is immediately "ACK" message. At that time, the smart gateway will add the newly-found device in the list of devices and sends a second "ACK" message to the device. If the smart gateway receives a response from the device, it will generate a single URL and the corresponding resource presentation. The device management broadcasting periodically a message, if the embedded device that depletion of energy or broken down cannot give a response to the smart gateway. If the smart gateway device management does not receive a response more than five minutes, device management will delete the item in list of devices and destroy the URL.

Data management incorporate data and command parse modules, which convert and parse data among different dedicated communication protocols. Presentation module will send this URL clicked by clients to the command parse module. The command parses module will analysis and parses the URL, and send a command to the sink. Sensors transmit data through IEEE802.15.4 to the sink. Sink receive data and transmit it to the data management by the serial port. The data parse module complete data parse and store it in the database or cache.

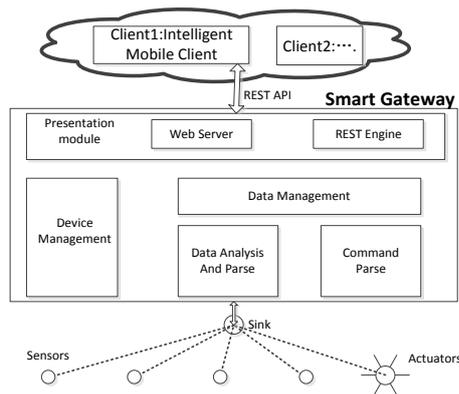


Fig. 1. Restful Smart Gateway Architecture

4.2. Restful Embedded Devices

We decided to select sensor nodes to represent embedded devices in our implementation efforts, since they are very easy to program and they offer some basic sensing capabilities. Our version of restful smart gateway that handles HTTP requests and thus abandons the idea of having an HTTP server directly on the device. The reason is that improving the scalability of restful smart gateway with respect to concurrent requests on the same device, enhanced support for asynchronous communication for the values returned by the devices. To exchange sensor and actuator data between the device and the gateway, we use the device management and data management to guarantee the smart gateway return the currently stored representation of the device.

These sensor nodes are equipped with a 250kbps, 2.4GHz, IEEE802.15.4-compliant Chip CC2420 Radio. IEEE802.15.4 is an IEEE standard that defines a MAC and PHY layer targeted to Wireless Sensor Networks (WSN). Fig. 2 shows the node of Telosb that we have used in the experiment.

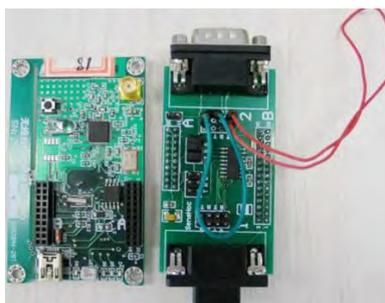


Fig. 2. The node of Telosb

We use two kinds of representations: XML and JSON.

- XML[18] is the default standard for structured information on the Web. Application scenarios can define their own schemas for XML, but in many cases standards or standards exist and can be reused. Depending on the type of resource, the representation must make the relevant aspects of the resource through XML
- For Web 2.0 applications, data is read from the server in JavaScript and then used to drive a dynamically updated Web page. The JavaScript Object Notation (JSON) [17] provides a better solution for JavaScript environments. This representation may be more limited than XML, but can be a good way to make resource data available for Web 2.0 applications.

In Table 1, we can see a general description of the Restful Web Services we created. SenCurrentRes predicates the root of current resources; SenActiveRes maintain the synchronization with the device list, predicating available resources; SenHistoryRes back given time history data;

SenCmdRes alter the state of physical devices. The first three resources have the same REST verbs, which GET is used to retrieve a representation of a resource. The fourth actuator has POST, which is a method to alter the state of a resource. All the resources have two kinds' types. Once the smart gateway receives a Restful HTTP request from a Web client, it makes the necessary validity checks and encapsulates the request as XML or JSON. It sends them to the sink, which is acting as a base station and directly connected to the smart gateway by serial port.

Table 1. A list of the RESTful devices

Resource URI	REST Verb	MIME Type	URL (the fixed prefix are http://localhost:port/wsn/)
SenCurrentRes	GET	XML/JSON	sensors/{sensorid}/{dataType}.{mediaType}
SenHistoryRes	GET	XML/JSON	sensors/history/{sensorid}/{dataType}/{fromTime}/{toTime}.{mediaType}
SenActiveRes	GET	XML/JSON	sensors/activeNode/all.{dataType}.{mediaType}
SenCmdRes	POST	XML/JSON	sensors/{sensorid}/command/{dataType}/interval



Fig. 3. The user interface of the client is showing the emperature of the surround environment which the device measured.

We have created the mobile phone application using the services provided by the restful gateway. The application combines physical and Web resources in the Restful API. The user interface of the application running the smart phone is shown as Fig. 3. The phone view is used to obtain current data from the wireless sensor network. The smart phone updates the current

environment information every ten seconds through sending GET URL requirement to the restful gateway. The gateway will give the response as the feedback to the phone. As shown in Fig. 3, we have chosen the temperature sensor whose device number is one to get current laboratory indoor environmental temperature. The real-time data view displays the current temperature information in the form of the compass.

5. Evaluation

Using the uniform interface would inevitably sacrifice efficiency specially compared with these gateway just transmit sensor data without using URL. Therefore, we have evaluated the performance of the prototype system through testing the amount of delay time from a request to the back response. As a compared target, we have also gain another delay time which through a normal gateway just translating sensor data to the Internet.

We have set the restful gateway and the service provider into the same personal computer. Around it, we deployed fixed scale of ten sensors. The sink node was plugged in one of the USB ports of the personal computer to serve as a base station, to forward IEEE802.15.4 wireless packets from/to the personal computer through the serial-over-USB port. Since the internet delay is dynamic, the client and the smart gateway using the same local IP address. We focus on gain the delay time from the smart gateway serial port reading date to the client getting the response. We performed ten times test and gain their average time in the experiment.

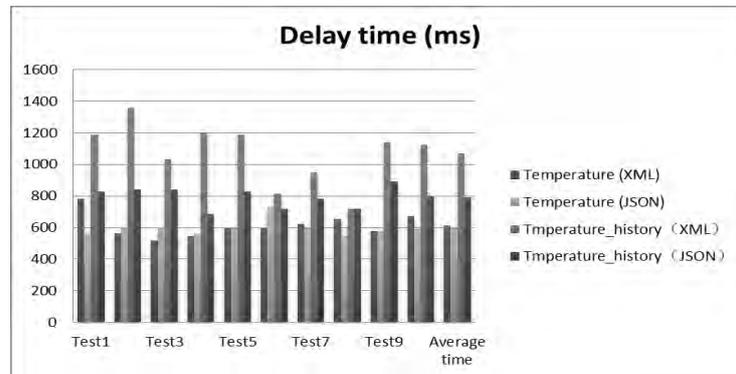


Fig. 4. Time is the duration from the creation of a request in the client to the arrival of the response back by smart gateway.

Fig. 4 presents the results of the experiment executions delay time. There are four test experiments: temperature real-time date with XML presentation, temperature real-time date with JSON presentation, temperature history date with XML presentation, temperature history date with JSON presentation. Since the temperature history back response has more than 1000 items of

data compared with the current data back response, it has longer delay time in the experiment. When the amount of data is small (real-time data response is small amount), JSON and XML presentations have more or less time delay. If the data amount is large, using JSON as resource presentation has less time delay than XML.

Further, we have divided the delay time consumed in the restful gateway into three stages, which includes the time reading from the serial-over-USB port and transformation data from Zigbee format to Ethernet format, the time stored in database, the time encapsulated data to the URL's presentation. As a compared target, the normal gateway didn't comply with REST principles, which just contains read date from serial port and data transformation. T_{total} is the total time consuming in the restful gateway, T_1 is the time that read data by the serial port and data interpret; T_2 is the time storing the data into database or cache, T_3 is the time that encapsulated data to resource presentation. We performed ten times test and gain their average time in the experiment. Fig. 5 presents the results of the experiment executions. Since the restful gateway have additional database operation, which cost most time, it had longer delay time cost than the normal gateway. After all, the performance of the gateway is acceptable.

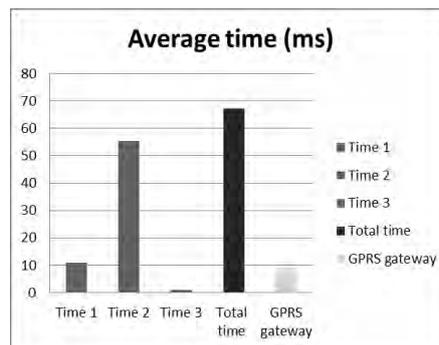


Fig. 5. Testing at each stage consuming time of the smart gateway and gateway

6. Conclusion and Future Work

In this paper we have proposed to use REST-Style architecture as a standard, to guide designing and developing of CPS. We developed a smart gateway, based on REST architectural style, which offers a uniform, efficient, and standardized way of interacting with physical devices. Flexible applications on top of heterogeneous devices can be built with little effort and acceptable performance, with any programming language that supports HTTP while Web-based solutions for traditional challenges of ad hoc environments. Through this work, our project constitutes a contribution towards the REST-style architecture for CPS.

Our future research on the REST-style architecture for CPS, besides a further evaluation about our smart gateway, is focus on some more research in device management on the system. Some advantage works in restful smart gateway for CPS are:

- i. Our smart gateway abandoned the idea of having an HTTP server directly on the device. This requires a synchronization-based mechanism. Our work just simply used device management to deal with it.
- ii. The REST-style architecture is most prominently the pull-based interaction model that is not suitable for modeling the communication with smart devices with respect to monitoring. Our smart gateway just periodic queries the messages to make up this drawback.

Acknowledgements. This work has been supported in part by State Key Development Program of Basic Research of China (Gant No.: 2011CB302902), Special Program for Key Basic Research of the Ministry of Science and Technology, China (Gant No.: 2011ZX03005-004-01) and National Natural Science Foundation of China (Gant No. : 61073180). We would like to thank the reviewers for their helpful comments on this paper.

References

1. Lee, EA.: Cyber physical systems: Design challenges. Isorc 2008: 11th IEEE Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing - Proceedings. 2008: 363-9
2. Prepared by the CPS Steering Group, USA. Cyber-physical Systems Executive Summary, March 6, 2008. <http://varma.ece.cmu.edu/summit/CPS-Executive-Summary.pdf>.
3. Rajkumar, R., Lee, I., Sha, L., Stankovic, J.: Cyber-physical systems: the next computing revolution. Proceedings of the 47th Design Automation Conference. Anaheim, California: ACM; 2010. p. 731-6.
4. Fielding, R.: Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, University of California, Irvine, California, 2000.
5. Wilde, E.: Putting Things to REST. Uc Berkeley, 2007.
6. Ying, T., Vuran, M.C., Goddard, S.: Spatio-Temporal Event Model for Cyber-Physical Systems. Distributed Computing Systems Workshops, 2009 ICDCS Workshops '09 29th IEEE International Conference on; 2009 22-26 June 2009; 2009. p. 44-50.
7. Tan, Y., Goddard, S.: A prototype architecture for cyber-physical systems. SIGBED Rev. 2008; 5(1): 1-2.
8. Abdelwahed, S., Kandasamy, N., Gokhale, A.: High confidence software for cyber-physical systems. Proceedings of the 2007 workshop on Automating service quality: Held at the International Conference on Automated Software Engineering (ASE). Atlanta, Georgia: ACM; 2007. p. 1-3.
9. Chang, Y.J., Huang, W.T.: A Novel Design of Data-Driven Architecture for Remote Monitoring and Remote Control of Sensors over a Wireless Sensor Network and the Internet. J Internet Technol. 2011; 12(1): 129-37.
10. Pinto, J., Martins, R., Sousa, J.B.: Towards a REST-style architecture for networked vehicles and sensors. Pervasive Computing and Communications

Qiang Li, Weijun Qin, Bing Han, Ruicong Wang, and Limin Sun

- Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on; 2010 March 29 2010-April 2 2010; 2010. p. 745-50.
11. Romero, D., Hermosillo, G., Taherkordi, A., Nzekwa, R., Rouvoy, R., Eliassen, F.: RESTful Integration of Heterogeneous Devices in Pervasive Environments. Distributed Applications and Interoperable Systems, Proceedings. 2010; 6115: 1-14 243.
 12. Dumitrache, I.: The next generation of Cyber-Physical Systems. Control Eng Appl Inf. 2010; 12(2): 3-4.
 13. Stankovic, J.A.: Cyber Physical Systems - Aspects as a Basis for Robustness and Openness. Aosd'09: 8th International Conference on Aspect-Oriented Software Development. 2009: 123-266.
 14. Sztipanovits, J.: Composition of cyber-physical systems. ECBS 2007: 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, Proceedings. 2007: 3-4 614.
 15. Wolf, W.: Cyber-physical Systems. Computer. 2009; 42(3): 88-9.
 16. Restlet - RESTful web framework for Java, 2007. <http://www.restlet.org/>.
 17. Crockford, D.: The application/json Media Type for JavaScript Object Notation (JSON). Internet informational RFC 4627, July 2006.
 18. DeRose, S., Maler, E., and Orchard, D.: XML Linking Language (XLink) Version 1.0. World Wide Web Consortium, Recommendation REC-xlink-20010627, June 2001.
 19. Luckenbach, T., Gober, P., Arbanowski, S., Kotsopoulos, A., and Kim, K.: TinyREST-a protocol for integrating sensor networks into the internet. in Proc. of REALWSN, 2005.
 20. Stirbu, V.: Towards a restful plug and play experience in the web of things. Semantic Computing, 2008 IEEE International Conference on, pages 512-517, Aug. 2008.

Qiang Li received his B.S. degree in University of Nankai, Tianjin, China, in 2009. Now he is a PHD student in Institute of Software, the Chinese Academy of Sciences, Beijing, China. His interests include Cyber Physical System, wireless local network and Technologies for Next Generation of Internet.

Weijun Qin is an assistive research professor at the Institute of Software, Chinese Academy of Sciences. He completed the master and PhD in 2010 at Tsinghua University in the area of pervasive computing, and received the bachelor's degree in computer science from Tsinghua University in 2003. His research interests include wireless sensor networks, internet of things, pervasive computing and context-aware computing.

Bing Han received the M.S. degree from Beijing University, in China, in 2011. He was a visiting student at the Institute of Software, Chinese Academy of Sciences in 2010. His interests include Cyber Physical System and wireless sensor network.

Ruicong Wang received the M.S. degree from Beijing University in China, in 2011. She was He was a visiting student at the Institute of Software, Chinese Academy of Sciences in 2010. Her interests include Cyber Physical System and wireless sensor network.

Limin Sun received his B.S., M.S., and D.Sc. degree in College of Computer, National University of Defense Technology, in China, in 1988, 1995, and 1998, respectively. He is currently a Professor in Institute of Software at Chinese Academy of Sciences (ISCAS). He is also a member of IEEE and a senior member of China Computer Federation (CCF). He is an editor of Journal of Computer Science and an editor of Journal computer Applications (Chinese journals). He is also a guest editor of special issues in EURASIP, Journal of Wireless Communications and Networking, and Journal of Networks, respectively. His research interests include Mobile Vehicle Networks, Delay-Tolerant Networks, Wireless Sensor Networks, Mobile IP and Technologies for Next Generation of Internet.

Received: March 10, 2011; Accepted: May 3, 2011.

CIP – Каталогизacija y publikaciji
Народна библиотека Србије, Београд

004

COMPUTER Science and Information
Systems : the International journal /
Editor-in-Chief **Mirjana Ivanović**. – Vol. 8,
No 4 (2011) - . – Novi Sad (**Trg D. Obradovića**
3): ComSIS Consortium, 2011 - (Belgrade
: Sgra star). –30 cm

Polugodišnje. – Tekst na engleskom jeziku

ISSN 1820-0214 = Computer Science and
Information Systems
COBISS.SR-ID 112261644

Cover design: **V. Štavljanin**
Printed by: Sgra star, Belgrade

ComSIS Vol. 8, No. 4, Special Issue, October 2011



Contents

Editorial

Papers

- 953 An Improved Node Localization Algorithm Based on DV-Hop for Wireless Sensor Networks
Qingji Qian, Xuanjing Shen, Haipeng Chen
- 973 Energy-Efficient Opportunistic Localization with Indoor Wireless Sensor Networks
Feng Xia, Xue Yang, Haifeng Liu, Da Zhang, Wenhong Zhao
- 991 An Energy Efficient and Load Balancing Routing Algorithm for Wireless Sensor Networks
Jin Wang, Tinghuai Ma, Jinsung Cho, Sungoung Lee
- 1009 A Redistribution Method to Conserve Data in Isolated Energy-harvesting Sensor Networks
Peng Liu, Song Zhang, Jian Qiu, Xingfa Shen, Jianhui Zhang
- 1027 A Packet Buffer Evaluation Method Exploiting Queueing Theory for Wireless Sensor Networks
Tie Qiu, Lin Feng, Feng Xia, Guowei Wu, Yu Zhou
- 1051 On the Efficiency of Cluster-based Approaches for Motion Detection using Body Sensor Networks
Kun-chan Lan, Chien-Ming Chou, Tzu-nung Wang, Mei-Wen Li
- 1073 Optimization of Multiple Gateway Deployment for Underwater Acoustic Sensor Networks
Jugen Nie, Deshi Li, Yanyan Han
- 1097 A Distributed Power Management Design Based on MOST Networks
Yushan Jin, Ruikai Liu, Xingran He, Yongping Huang
- 1117 An Energy-Efficient Localization Strategy for Smartphones
Haifeng Liu, Feng Xia, Zhuo Yang, Yang Cao
- 1129 Modeling Disease Spreading on Complex Networks
Xiangjie Kong, Yu Qi, Xiumiao Song, Guojiang Shen
- 1143 An Improved Spectral Clustering Algorithm Based on Local Neighbors in Kernel Space
Xinyue Liu, Xing Yong, Hongfei Lin
- 1159 A Novel Capacity and Trust Based Service Selection Mechanism for Collaborative Decision Making in CPS
Bo Zhang, Yang Xiang, Peng Wang, Zhenhua Huang
- 1185 Privacy Preserving in Ubiquitous Computing: Classification & Hierarchy
Tinghuai Ma, Sen Yan, Jin Wang, Sungyoung Lee
- 1207 TRM-IoT: A Trust Management Model Based on Fuzzy Reputation for Internet of Things
Dong Chen, Guiran Chang, Dawei Sun, Jiajia Li, Jie Jia, Xingwei Wang
- 1229 A Reusable Agent Design Pattern with Flexibility and Extensibility
Haolan Zhang, Wenhua Zeng, Christian Van der Velden
- 1251 Quantitative Analysis for Symbolic Heap Bounds of CPS Software
Renjian Li, Ji Wang, Liqian Chen, Wanwei Liu, Dengping Wei
- 1277 A Model-Based Software Development Method for Automotive Cyber-Physical Systems
Zhigang Gao, Haixia Xia, Guojun Dai
- 1303 Velocity Adaptation for Synchronizing a Mobile Agent Network
Lijing Li, Hui Yan, Hui Li, Chunxi Zhang
- 1317 A Case Study on REST-Style Architecture for Cyber-Physical Systems: Restful Smart Gateway
Qiang Li, Weijun Qin, Bing Han, Ruicong Wang, Limin Sun
-