SENSORCOMM 2011

The Fifth International Conference on Sensor Technologies and Applications

WSNSCM 2011

The First International Workshop on Sensor Networks for Supply Chain Management

ISBN: 978-1-61208-144-1

August 21-27, 2011

Nice/Saint Laurent du Var, France

SENSORCOMM 2011 Editors

Jens Martin Hovem, Norwegian University of Science and Technology, Norway

Joshua Ellul, Imperial College, London, UK

Laurent Gomez, SAP Labs France, SAS Mougins - France

Yenumula Reddy, Grambling State University, USA

SENSORCOMM 2011

Foreword

The Fifth International Conference on Sensor Technologies and Applications (SENSORCOMM 2011), held between August 21-27, 2011 in Nice/Saint Laurent du Var, France, was a multi-track event covering related topics on theory and practice on wired and wireless sensors and sensor networks.

Sensors and sensor networks have become a highly active research area because of their potential of providing diverse services to a broad range of applications, not only on science and engineering, but equally importantly on issues related to critical infrastructure protection and security, health care, the environment, energy, food safety, and the potential impact on the quality of all areas of life.

Sensor networks and sensor-based systems support many applications on the ground today. Underwater operations and applications are quite limited by comparison. Most applications refer to remotely controlled submersibles and wide-area data collection systems at a coarse granularity.

Underwater sensor networks have many potential applications such a seismic imaging of undersea oilfields as a representative application. Oceanographic research is also based on the advances in underwater data collection systems.

There are specific technical aspects to realize underwater applications which cannot be borrowed from the ground-based sensors net research. Radio is not suitable for underwater systems because of extremely limited propagation. Acoustic telemetry could be used in underwater communication; however off-the-shelf acoustic modems are not recommended for underwater sensor networks with hundreds of nodes because they were designed for long-range and expensive. As the speed of light (radio) is five orders of magnitude higher than the speed of sound, there are fundamental implications of time synchronization and propagation delays for localization. Additionally, existing communication protocols are not designed to deal with long sleep times and they can't shut down and quickly restart.

In wireless sensor and micro-sensor networks, energy consumption is a key factor for the sensor lifetime and accuracy of information. Protocols and mechanisms have been proposed for energy optimization considering various communication factors and types of applications. Conserving energy and optimizing energy consumption are challenges in wireless sensor networks, requiring energy-adaptive protocols, self-organization, and balanced forwarding mechanisms.

SENSORCOMM 2011 also included:

- WSNSCM 2011, The First International Workshop on Sensor Networks for Supply Chain Management

We take here the opportunity to warmly thank all the members of the SENSORCOMM 2011 Technical Program Committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to SENSORCOMM 2010. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the SENSORCOMM 2011 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.
We hope that SENSORCOMM 2011 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the area of sensor and sensor networks.

We hope Côte d'Azur provided a pleasant environment during the conference and everyone saved some time for exploring the Mediterranean Coast.

SENSORCOMM 2011 Chairs

Advisory Chairs
Jean Philippe Vasseur, Cisco Systems, Inc., France
Petre Dini, Concordia University, Canada / China Space Agency Center, China
Jaime Lloret Mauri, Polytechnic University of Valencia, Spain
Jens Martin Hovem, Norwegian University of Science and Technology, Norway
Pascal Lorenz, University of Haute Alsace, France
Sergey Yurish, IFSA, Spain

Industry Liaison Chairs
Sarfraz Khokhar, Cisco Systems, Inc., USA
Harkirat Singh, Samsung Electronics Co., Korea
Javier Del Ser Lorente, TECNALIA-Telecom - Zamudio (Bizkaia), Spain
Michael Niedermayer, Fraunhofer IZM, Germany

Research/Industry Chairs
Hristo Djidjev, Los Alamos National Laboratory, USA Teng Rui, National Institute of Information and Communication Technology, Japan
S. Biju Kumar, Philips Research - Eindhoven, The Netherlands

Special Area Chairs

Embedded systems
Joshua Ellul, Imperial College, London, UK

Security
Yenumula Reddy, Grambling State University, USA

Body networks
Alessandro Pozzebo, Università degli Studi di Siena, Italy

Underwater systems
Mylène Toulgoat, Communications Research Centre - Ottawa, Canada

Applications
Elena Gaura, Coventry University, UK

Performance
Canfeng Chen, Nokia Research Center - Beijing, China

WSNSCM Chair
Laurent Gomez, SAP Labs France, SAS Mougins - France

SENSORCOMM 2011

Committee

SENSORCOMM Advisory Chairs

Jean Philippe Vasseur, Cisco Systems, Inc., France
Petre Dini, Concordia University, Canada / China Space Agency Center, China
Jaime Lloret Mauri, Polytechnic University of Valencia, Spain
Jens Martin Hovem, Norwegian University of Science and Technology, Norway
Pascal Lorenz, University of Haute Alsace, France
Sergey Yurish, IFSA, Spain

SENSORCOMM 2011 Industry Liaison Chairs

Sarfraz Khokhar, Cisco Systems, Inc., USA
Harkirat Singh, Samsung Electronics Co., Korea
Javier Del Ser Lorente, TECNALIA-Telecom - Zamudio (Bizkaia), Spain
Michael Niedermayer, Fraunhofer IZM, Germany

SENSORCOMM 2011 Research/Industry Chairs

Hristo Djidjev, Los Alamos National Laboratory, USA Teng Rui, National Institute of Information and Communication Technology, Japan
S. Biju Kumar, Philips Research - Eindhoven, The Netherlands

SENSORCOMM 2011 Special Area Chairs

Embedded systems
Joshua Ellul, Imperial College, London, UK

Security
Yenumula Reddy, Grambling State University, USA

Body networks
Alessandro Pozzebo, Università degli Studi di Siena, Italy

Underwater systems
Mylène Toulgoat, Communications Research Centre - Ottawa, Canada

Applications
Elena Gaura, Coventry University, UK

Performance
Canfeng Chen, Nokia Research Center - Beijing, China

SENSORCOMM 2011 Technical Program Committee

Majid Bayani Abbasy, Universidad Nacional de Costa Rica, Costa Rica
Saied Abedi, Fujitsu Laboratories of Europe LTD. - Middlesex, UK
Abdullah A. Al-Shehri, Saudi Aramco, Saudi Arabia
Sarfraz Alam, University Graduate Center - Kjeller, Norway
Andrei Alexandru Enescu, Polytechnic University of Bucharest, Romania
Mothanna Alkubeily, Université de Technologie de Compiègne, France
Boushra Alkubily, Université de Technologie de Compiègne, France
Al-Khateeb Anwar, Politecnico di Torino, Italy
Miquel Ardid Ramirez, Universitat Politècnica de València, Spain
Nauman Aslam, Dalhousie University - Halifax, Canada
Isabelle Augé-Blum, INSA Lyon - Laboratoire CITI -Villeurbanne, France
Reza Azarderakhsh, The University of Western Ontario, Canada
Sebastian Bader, Mid Sweden University, Sweden
Faruk Bagci, German University Cairo, Egypt
Valentina Baljak, National Institute of Informatics & University of Tokyo, Japan
Dominique Barthel, Orange Labs Division R&D - Meylan, France
Novella Bartolini, "Sapienza" University of Rome, Italy
Rezaul K. Begg, Victoria University - Melbourne, Australia
Paolo Bellavista, University of Bologna, Italy
Alessandro Bogliolo, University of Urbino, Italy
Lina Brito, University of Madeira, Portugal
Tiziana Calamoneri, “La Sapienza” Università di Roma, Italy
Maria-Dolores Cano Baños, Technical University of Cartagena, Spain
Juan Vicente Capella Hernández, Universidad Politécnica de Valencia, Spain
Berta Carballido Villaverde, Cork Institute of Technology, Ireland
Mari Carmen Domingo, Barcelona Tech University, Spain
Angel Catala Monzo, Universidad Politecnica de Valencia, Spain
Chao-Tsun Chang, Hsiuping Institute of Technology, Taiwan
Chih-Yung Chang (張志勇), Tamkang University, Taiwan
Canfeng Chen, Nokia Research Center - Beijing, China
Shu-Ching Chen, Florida International University - Miami, USA
Hugo Coll Ferri, Polytechnic University of Valencia, Spain
Daniel Curiac, "Politehnica" University of Timisoara, Romania
Javier Del Ser Lorente, TECNALIA-Telecom - Zamudio (Bizkaia), Spain
Jerker Delsing, Lulea University of Technology, Sweden
Behnam Dezfouli, University Technology Malaysia (UTM), Malaysia
Constantin Doru, University of Pitesti, Romania
Wan Du, University of Lyon (Ecole Centrale de Lyon), France
Juan C. Dueñas, Professor, Universidad Politecnica de Madrid, Spain
Imad H. Elhajj, American University of Beirut, Lebanon
Joshua D. Ellul, Imperial College London, UK
Nader Faisal Jaafar Mohamed, UAEU, UAE
Xiang Fei, Coventry University, UK
Paulo Felisberto, Institut for Systems and Robotics-Lisbon / Universidade do Algarve, Portugal
Armando Ferro Vázquez, Universidad del País Vasco - Bilbao, Spain
Paul J. Fortier, University of Massachusetts Dartmouth, USA

Abdallah Makhoul, Université de Besancon - Belfort, France
Gianfranco Manes, Univerisità degli studi di Firenze, Italy
Alireza Masoum, Twente University, The Netherlands
Natarajan Meghanathan, Jackson State University, USA
Nathalie Mitton, INRIA-Lille Nord, France
Deok Hee Nam, Wilberforce University, USA
Enrico Natalizio, INRIA-Lille, France
Mahmuda Naznin, Bangladesh University of Engineering and Technology - Dhaka, Bangladesh
Arye Nehorai, Washington University in St Louis, USA
Sarmistha Neogy, Jadavpur University, India
Michael Niedermayer, Fraunhofer Institute for Reliability and Microintegration, Germany
Frank Oldewurtel, RWTH Aachen University, Germany
Carlos Enrique Palau Salvador, Universidad Politecnica de Valencia, Spain
Nikolaos A. Pantazis, Technological Educational Institution (TEI) of Athens, Greece
Miodrag Potkonjak, University of California - Los Angeles, USA
Alessandro Pozzebo , Università degli Studi di Siena, Italy
Shrisha Rao, International Institute of Information Technology - Bangalore, India
Yenumula Reddy, Grambling State University, USA
Càndid Reig, University of Valencia, Spain
Joel Rodrigues, University of Beira Interior, Portugal
Nirmalya Roy, Institute for Infocomm Research (I2R), Singapore
Lorenzo Rubio-Arjona, Universidad Politécnica de Valencia, Spain
Teng Rui, National Institute of Information and Communication Technology, Japan
Jorge Sá Silva, University of Coimbra, Portugal
Husnain Saeed, Trinity College Dublin, Ireland
Addisson Salazar, Polytechnic University of Valencia, Spain
Mikko Sallinen, University of Oulu, Finland
Ioakeim Samaras, Aristotle University of Thessaloniki, Greece
Francisco Javier Sánchez Bolumar, ADIF, Spain
Leo Selavo, University of Latvia, Latvia
Sandra Sendra Compte, Universidad Politécnica de Valencia, Spain
Kuei-Ping Shih, Tamkang University - Taipei, Taiwan
Salman Ijaz Siddiqui, Universidade do Algarve, Portugal
Simone Silvestri, Sapienza University of Rome, Italy
Radosveta Sokullu, Ege University - Izmir, Turkey
Peter Soreanu, Ort Braude College, Israel
Arvind K. Srivastava, NanoSonix Inc. - Skokie, USA
Grigore Stamatescu, University Politehnica of Bucharest, Romania
Tsenka Stoyanova, University of Patras, Greece
Junzhao Sun, University of Oulu, Finland
Zahra Taghikhaki, University of Twente, The Netherlands
Muhammad Tariq, Waseda University - Tokyo, Japan
Lothar Thiele, ETH Zurich, Switzerland
Rolf Thomasius, Fraunhofer Institut für Zuverlässigkeit und Mikrointegration - Berlin, Germany
Mylène Toulgoat, Communications Research Centre - Ottawa, Canada
Bernard Tourancheau, INRIA, France
Neeta Trivedi, Aeronautical Development Establishment- Bangalore, India
Wilfried Uhring, University of Strasbourg, France

Maurizio Valle, University of Genoa, Italy
Fabrice Valois, INRIA SWING / CITI, INSA-Lyon, France
Athanasios Vasilakos, University of Western Macedonia, Greece
Jean Philippe Vasseur, Cisco Systems, Inc., France
Roberto Verdone, Università di Bologna & IEIIT-BO, CNR & CNIT, Italy
Manuela Vieira, Instituto Superior de Engenharia de Lisboa, Portugal
Natalija Vlajic, York University - Toronto, Canada
Mike Walsh, Tyndall National Institute, Ireland
You-Chiun Wang, National Chiao-Tung University, Taiwan
Chih-Yu Wen, National Chung Hsing University - Taichung, Taiwan
Fang-Jing Wu, National Chiao Tung University, Taiwan
Xiaochun Xu, Amazon.com Inc., USA
Dongfang Yang, National Research Council Canada - London, Canada
Qiang Ye, University of Prince Edward Island, Canada
Norihiko Yoshida, Saitama University, Japan
Sergey Y. Yurish, IFSA - Barcelona, Spain
Tanveer Zia, Charles Sturt University, Australia

WSNSCM Chair

Laurent Gomez, SAP Labs France, SAS Mougins - France

WSNSCM 2011 Technical Program Committee

Omar Gaci, Institut Supérieur d'Etudes Logistiques, France
Dominique Guinard, MIT LMP Auto-ID Labs / ETH Zurich, Switzerland
Rasmus Krigslund, Aalborg University, Denmark
Annett Laube, Bern University of Applied Science, Switzerland
Maryline Laurent, Telecom Sud Paris, France
Herve Mathieu, Institut Supérieur d'Etudes Logistiques, France
Joachim Possega, Universtät Passau, Germany
Alessandro Sorniotti, IBM Research Zürich, Switzerland
Cedric Ulmer, SAP Research Sophia Antipolis, France

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Intelligent Road Infrastructure - A Concept Study

Ralf Wunderlich*, Sebastian Strache*, Christian Busen†, Bernhard Steinauer†, and Stefan Heinen*
*Chair of Integrated Analog Circuits
†The Institute of Road and Traffic Engineering Aachen
RWTH Aachen University, D-52062 Aachen, Germany
Email: ias@rwth-aachen.de

***Abstract*—A promising approach for future road construction is based on a surface which is manufactured in a production plant and can be unrolled on a conventional base. This paper deals with the idea to make the road ”intelligent” by integrating a net of sensor nodes. Equipped with acceleration sensors the sensor net can detect the traffic situation. With the knowledge of the exact position and velocity of each car a driver assistance system is able to find the fastest route or to give an accident warning on a very reliable level. This concept study describes the main idea of a low cost, energy harvesting sensor node containing an MEMS accelerometer with frontend, a processing unit, a photovoltaic energy harvesting power supply and a wireless communication link.**

***Keywords*—road construction, intelligent road, MEMS, energy harvesting, sensor node**

I. Introduction

The next step in road building is to introduce a high quality, centimeter thin top cover manufactured in production plants probably on the base of a synthetic material. A better noise reduction and a lower rolling drag can be achieved since a production in a plant is done under best conditions. The feasibility of a ”rollable” road has been shown in [1] and [2].

A conventional street needs a large amount of (oil based) bitumen, which is running out and therefore more expensive. In order to damp down the cost increase the amount of bitumen needed has to be reduced by decreasing the thickness of the bitumen based top layer of the road or to find a low priced suitable alternative material.

The road networks of developed countries are the backbones of their economies. To increase their capacity without building new roads intelligent traffic management systems are needed. These systems need exact information about the traffic flow to avoid traffic jams and minimize the individual traveling times. The quality of traffic management heavily depends on good input data.

Further it is getting more and more challenging and expensive to enhance the safety with vehicle based systems. In near future driving assistance systems will be upgraded to highly automated, nearly autonomous systems which need more reliable data to proceed. Vehicle based sensor systems are not able to provide these systems with data on the required quality level in every situation as shown by [3]. In contrast to this the here proposed infrastructure can be used for autonomous driving in the long run. Furthermore it can also provide information about the road situation ahead of the individual vehicle like humidity, temperature and slickness which are unknown for vehicle based systems.

II. System Concept

Fig. 1 shows a cross section of the ”intelligent” road. The base layer can be build conventional, the top layer is called car-pad (analogy to carpet). The centimeter thin, high quality, industrialy produced car-pad is unrolled and mounted to the conventional base. Therefore the time required for construction works is short.

The industrial fabrication of the car-pad enables the integration of sensors inside the top layer of the road (colored dots in Fig. 1). Integrated wireless sensor nodes allow to measure passing vehicles and different ambient road conditions like temperature or moisture. The cost of the sensor net is almost insignificant compared to present-day’s overall road costs.

Each passing vehicle generates structure-borne sound which propagate through the top layer of the road and can be detected by an accelerometer (see Fig. 2). Accelerometers have very small dimensions, are cheap and can easily be integrated inside a sensor node.

After preprocessing, the data are transmitted to small base stations mounted in beacons at the side of the road. These base stations combine the information of the different sensor nodes to a more global view of the traffic flow at this part of the road. The base stations transmit this information to a central traffic management center and to the passing cars.

Due to the very dense sensor network not only the traffic flow but also local events like accidents or other dangerous situations like cars in blind spots can be detected and road users can be warned. Since the information is determined by the infrastructure and not by a car based system, the system will even work if most users are *not* equipped with a vehicle on-board unit. To participate from warning signal no special hardware is needed. The on-board unit can be integrated into the navigation system.

Subsequently the energy harvesting sensor node is discussed. Besides the utilization in the road, this sensor node can also be used for structural health monitoring (e. g. in bridges or buildings).

III. Initial Measurements

To the knowledge of the authors this is the first project which wants to use the structure-borne sound of vehicles for their detection. Since there were no data available for the

ISBN: 978-1-61208-144-1

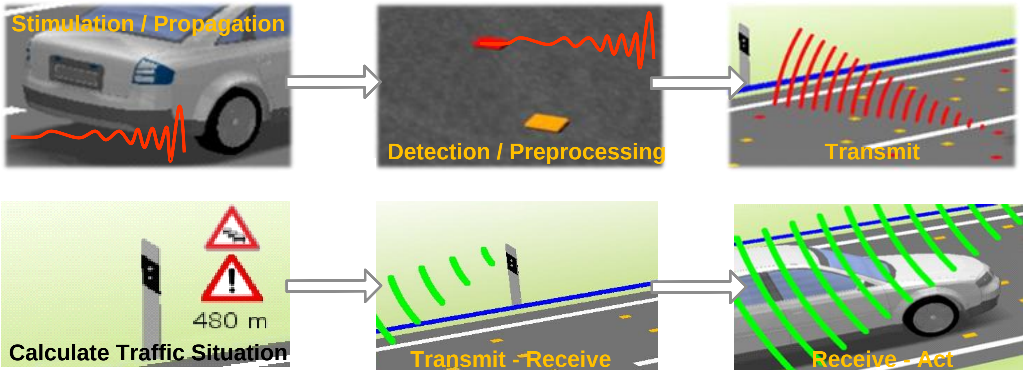

Stimulation / Propagation
Detection / Preprocessing
Transmit
Calculate Traffic Situation
480 m
Transmit - Receive
Receive - Act

Fig. 2. Working principal.

damping of the structure-borne sound nor the spectrum and the amplitude of that sound a measurement series was performed.

A commercial accelerometer with a sensitivity of 2270 mVs^2/m was used. Fig. 3 shows the accelerometers output for a passing car with 50 km/h in a distance of 30 cm. The peaks in the acceleration show the points when the car's axes passes the sensor.

In order to reduce the amount of transmitted data, preprocessing is needed inside the sensor node. This can be realized using a straight forward filter as shown in Fig. 4 which can recognize the axes of passing vehicles.

The filter output is drawn in Fig. 5. Its period was experimentally determined to 10 ms, the sample time to 1 ms.

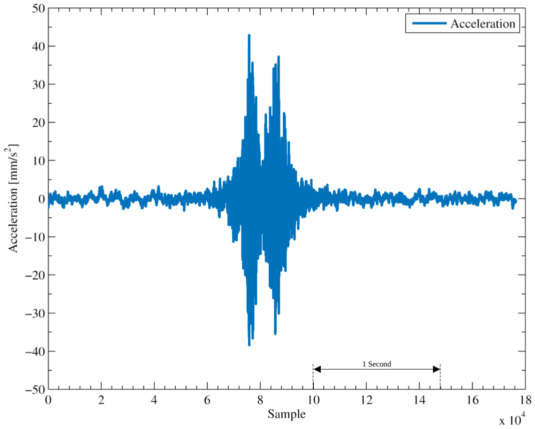


Fig. 3. Passing car with 50 km/h in 30 cm distance.

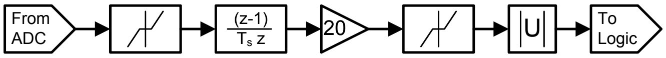


Fig. 4. Digital Filter Algorithm.

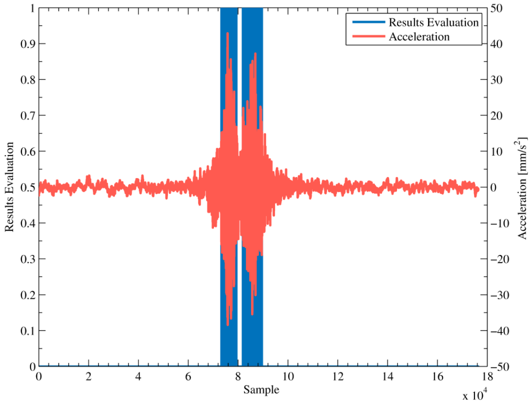


Fig. 5. Recognition with the filter of a passing car with 50 km/h in 30 cm distance.

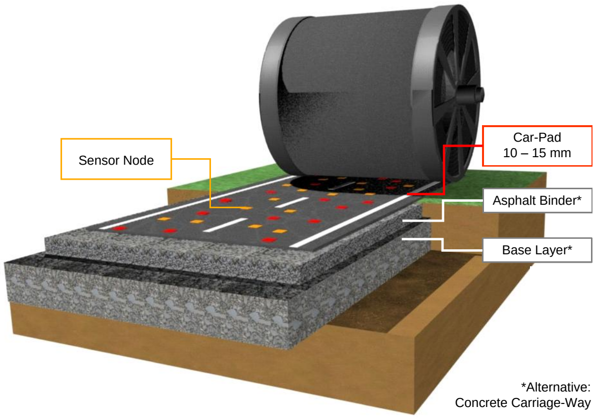


Fig. 1. Structure ”intelligent” road.

IV. Sensor Node

The structure of the sensor node to be integrated in the top layer of the road is shown in Fig 6. The power supply of the sensor node consists of series connected solar cells, power regulators, an energy storage and a power management unit. A MEMS (Microelectromechanical System) accelerometer is the primary sensor of the node to detect the structure born sound of passing vehicles. Further sensors for temperature and humidity can also be attached to the ASIC. Only the solar cells, the supercapacitor and the accelerometer sensing element are

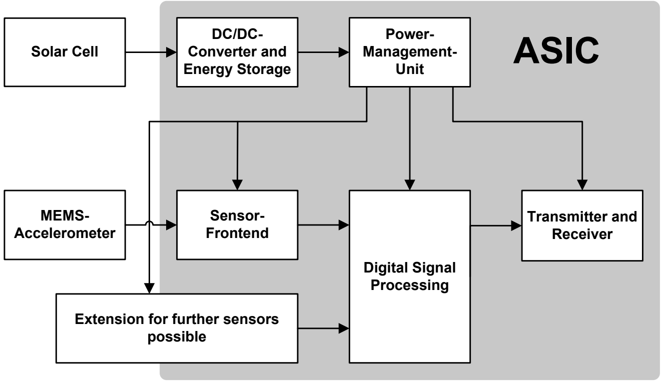

Solar Cell
DC/DC-Converter and Energy Storage
Power-Management-Unit
ASIC
MEMS-Accelerometer
Sensor-Frontend
Digital Signal Processing
Transmitter and Receiver
Extension for further sensors possible

Fig. 6. Block diagram sensor node.

external components. All other functions are fully integrated into the ASIC. By this the sensor node dimensions, its power consumption and its price are drastically reduced.

In the following section the power supply and its components are described. Afterwards the different building blocks of the sensor node, the transmitter, the MEMS accelerometer and the accelerometer frontend are presented.

A. Power Supply

The power supply for the sensor node inside the top layer of the road is a critical issue. A lifetime of 10 years is targeted for the top layer of the road. Therefore it is necessary that the senor node can operate for an even longer period of time. Batteries are not suitable for this application, because it is not possible to operate the sensor nodes for the required lifetime with affordable and small enough batteries inside the top layer of the road. Especially the high temperature differences during a year shorten the lifetimes of batteries and make their use in this application impossible [4].

Since primary batteries are not an option energy harvesting has to be used for the power supply. In [5] different energy sources like solar cells, vibration or temperature energy harvesting have been compared to find the optimal energy source for this application. Solar cells have been selected, because they offer the highest energy density for this application. Furthermore they are well studied and available for low cost. Their only drawback is the need for a transparent package with low reflection to maximize the irradiated power for the solar cell. To keep the sensor node operating during night or times when the solar cell is covered with e. g. snow an energy storage is required. In addition a power management unit is needed to control the electronics to maximize the harvested energy and the operation time of the sensor node. This section describes the structure of the power supply and its components.

1) Photovoltaic Cell: Today different types of solar cells based on different materials like monocrystalline silicon or organic polymers are available. In comparison to the other types silicon thin film solar cells offer medium efficiency of more than 10 %, low price and long term stability [6]. Furthermore the efficiency of amorphous silicon is quite constant even for weak irradiated powers [7]. To consider the life time degradation an efficiency of only 7.8 % is assumed for the power supply.

2) Energy Storage: The output power of the photovoltaic cells depends on the irradiated power which varies strongly. Therefore an energy storage is needed to supply the sensor node with enough energy if the solar cell cannot power it alone. For this application the specific energy and the number of charge-discharge cycles are the most important parameters for the energy storage. The specific power is not important because the sensor node needs less than 1 mW. Li-ion rechargeable batteries and supercapacitors are the most promising energy storage technologies for this demands [8]. Due to the harsh temperatures between -40 and 80°C in this applications batteries are not suitable because their capacity would drastically decrease within the first years of operation [4]. That is why supercapacitors have been selected as energy storage technology. They do not limit the lifetime of the sensor node nor lose much capacity due to the temperature range.

3) Power Management System: Besides the energy source and storage DC/DC-converters are needed to improve the available power for the sensor node. A direct connection of the solar cell to the ASIC with the supercapacitor in parallel would not work, because the supply voltage would change between 0 V and the open circuit voltage of the solar cells. Therefore the power supply concept depicted in Fig. 7 was developed. Instead of one large solar cell, 6 smaller ones with the same area are connected in series to increase the output voltage. A multiplexer can connect the supercapacitor or the solar cells to a charge pump. Instead of directly connecting the solar cells to the load, a charge pump is used to keep the cells in their maximum power point. This can increase the harvested energy by more than 25 % despite the losses in the charge pump [9]. The output energy from the charge pump can be used to charge the supercapacitor or directly power the LDO. Using the Multiplexer (MUX) and Demultiplexer (DEMUX) lowers the efficiency for the direct connection due resistive losses. Since the blocks only switch infrequent the transistors can be quite large minimizing the resistive losses. These multiplexers are used to enable four different operation modes. The first one is used to charge the supercapacitor with the energy out of the solar cells. In this mode the charge pump performs the maximum power point tracking and can increase the output voltage of the solar cells to utilize the full voltage range (up to 5 V) for the supercapcitor and maximize the stored charge. The ASIC is powered in the second operation mode from the supercapcitor. A LDO is used to stabilize the power supply voltage. Since the losses in the LDO are proportional to the voltage drop across it, the charge pump is used to minimize it. The ASIC can also be direct powered via the charge pump and the LDO if the output power of the solar cells is sufficient. Furthermore a mixed supply from the solar cells and the supercapacitor for the ASIC is possible, too.

The power supply of the sensor node has been designed for an operation in Germany but by adapting the solar cell area and supercapacitor size it can by used all over the world.

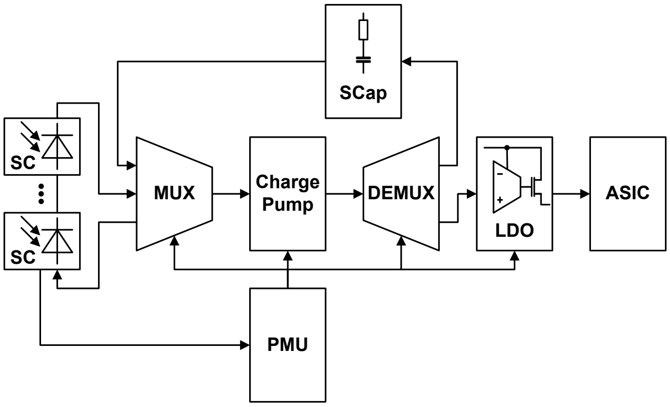


Fig. 7. Power Supply Sensor Node.

The ASIC has to be active every $20\,\mathrm{ms}$ for $2\,\mathrm{ms}$ to be able to detect passing vehicles with a speed up to $180\ \mathrm{km/h}$. During its active phase the ASIC needs less than $3.3\ \mathrm{mW}$. Therefore it consumes maximal $248\ \mathrm{mW}$ per month. Using the method described in [10] for the area around Aachen in Germany an average irradiated power of $0.69\,\mathrm{kWh/m^2}$ per day can be calculated for an availability of $95\,\%$. For an efficiency of $7.88\,\%$ an area of $3\ \mathrm{cm}^2$ of the solar cells is enough for the operation of the sensor node. The efficiency of the charge pump is assumed to be $90\,\%$ with a quiescent current of $2\ \mu\mathrm{A}$. Taking the production variations and degradation over the lifetime into account the supercapacitor size has been set to $106\ \mathrm{mF}$ which is $40\,\%$ larger than needed. The size of the energy storage enables the sensor node to operate in total darkness for 31 days which should ensure operation even during winter with a lot of snow. A more detailed design based on exact data will lead to a drastic size reduction for the solar cell and the energy storage, which will lower the price per sensor node.

B. Transmitter

The transmission of the senor data from the sensor node to the base station has to be wireless since a wired transmission would be too complex and expensive. For an approximation of the required energy for the data transmission the data packets and the attenuation have to be estimated. One data packet needs to consists of an ID, the sensor data and a checksum. It can be dimensioned to be less than 256 bit. The frequency band for the data transmission should be selected below $1\ \mathrm{GHz}$ due to the better propagation properties. Therefore the frequency ranges $869.40 - 869.65\ \mathrm{MHz}$ or $869.70 - 870.00\ \mathrm{MHz}$ have been selected for the data transmission. The attenuation for the radio wave could be calculated to be less than $61\ \mathrm{dB}$ for a German highway with three lanes per direction. This calculation assumes the absorption coefficient inside the top layer of the road to be 1.67. Since the base station will have more power available a transmitted power of $0.5\ \mathrm{mW}$ is sufficient for a stable data transmission. For this application the data transmission can be achieved with less than $2.4\ \mathrm{J}$ per data packet and fits well in the calculated power budget of $3.3\ \mathrm{mW}$ for $1\ \mathrm{ms}$ [11].

C. MEMS Accelerometer

Acceleration can be detected with different techniques like capacitive, piezoelectric or piezoresistive sensing. For this application capacitive MEMS accelerometers are most suitable, because they offer a small outline, low cost, low power consumption and constant quality. Due to their small size they have a low seismic mass which causes a high mechanical noise floor limiting the achievable resolution. The initial measurements on the test track have shown that the required resolution for the sensor nodes is $1\ \mathrm{mg}$, which can be achieved with MEMS accelerometers [12]. The challenge for this application is to maintain the required resolution with as little power consumption as possible and for a competitive price. Therefore the MEMS accelerometer have to be optimized for the structure born sound detection and for lowering the requirements for the readout circuits and its power consumption.

D. Accelerometer Readout Circuit

One main challenge of sensor node design for the ”intelligent” road infrastructure is the power efficient readout of the capacitive MEMS accelerometer. For a small parasitic capacitance at the readout node of the sensing element continuous time voltage sensing (CTV) is superior to other readout techniques like switched capacitor [13]. Furthermore it shows a better performance for low power readout circuits, because the resolution is not limited due to noise folding. In [5], [14] a special readout circuit was designed to meet the requirements for this application.

The accelerometer readout circuit is based on [12], [15] and uses dual-chopper stabilization for noise and offset reduction as well as power saving. As depicted in Fig. 8 the frontend uses energy efficient square wave stimulation with the frequency PHH ± PHL on port 1 and 2 of the sensing element. The acceleration dependent charge variations are evaluated on the middle port (M) of the sensing element. The first operation amplifier converts the charge into a voltage and adds flicker noise and offset to the signal. Its gain can be selected with the *gain0* and *gain1* switches. A passive mixer is used to mix the signal down from PHH ± PHL to PHL. Since the flicker noise and offset is around DC it is mixed up to PHL and apart from the signal. The second operational amplifier further boosts the signal and is more energy efficient since it operates at lower frequencies. After the second mixer the signal is mixed to DC and the noise is at PHL and PHH ± PHL, respectively. Therefore the noise can be reduced with a lowpass filter. The signal is digitized with an analog to digital converter.

To increase the resolution different calibration cycles are intended in the readout circuit. The sensor offset, which can drive the sensor frontend into saturation, is coarse reduced with an capacitor array at the middle port of the sensing element. A fine reduction is achieved with an DAC in the input stage of the second operation amplifier (not depicted in Fig 8). Furthermore the gain and offset for other building blocks are calibrated. To achieve the needed resolution of $1\ \mathrm{mg}$ a gain of more than

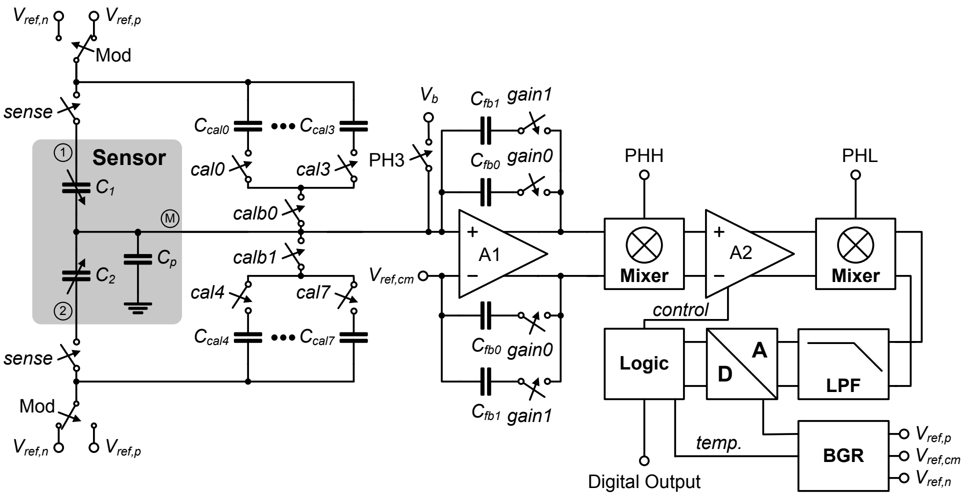


Fig. 8. Accelerometer Frontend.

60 dB is required which limits the input acceleration range to less than 1 g. The gains of both operational amplifiers are controlled by the logic to keep the frontend from saturating and therefore increase the acceleration input range.

V. Conclusion

In this paper a promising concept study of an ”intelligent” road has been shown. With this road the traffic situation can be determined without having special equipped cars. Due to the absolute and accurate position information of each car the exact traffic situation can be derived. Traditional navigation systems extended with a receiver can be used as vehicle on-board units. The system can not only be used for optimal routing but also to enhance security by giving drivers assistance and warnings.

It could be shown in a measurement series that structure born sound can be used to detect vehicles and their speed. The proposed sensor net can easily and at low cost be integrated into an industrial fabricated ”rollable” road. We have shown the sensor node on block level and listed the specifications. These specifications have been verified by reported circuitries listed in literature or by own transistor level simulations.

Acknowledgment

The authors would like to thank Dr. Dirk Droste et al. from the Robert Bosch GmbH for help and fruitful discussions concerning MEMS sensors and sensor frontends. The authors would also like to thank Dr. Christian Lank, who has been formerly with RWTH ISAC, for a lot of work in the project proposal.

References

[1] L. J. M., J. van der Kooij, R. W. M. Naus, and P. D. Bhairo, “Apt testing of modular pavement structure rollpave and comparison with conventional asphalt motorway structures,” *Accelerated Pavement Testing*, pp. 1–24, 2004.

[2] J. Groenendijk and G. Westera, “Proefvak rollpave a35. final report.” *Innovatieprogramma geluid voor weg- & spoorverkeer (iPG)*.

[3] S. Deutschle, G. Kessler, C. Lank, G. Hoffmann, M. Hakenberg, and M. Brummer, “Einsatz elektronisch gekoppelter LKW-Konvois auf Autobahnen,” *ATZ - Automobiltechnische Zeitschrift*, 2010.

[4] M. Perrin, P. Malbranche, E. Lemaire-Potteau, B. Willer, M. Soria, A. Jossen, M. Dahlen, A. Ruddell, I. Cyphelly, G. Semrau, D. Sauer, and G. Sarre, “Temperature behaviour: Comparison for nine storage technologies: Results from the investire network,” *Journal of Power Sources*, vol. 154, no. 2, pp. 545–549, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/B6TH1-4HRMV24-3/2/b0f09324ad0941ab3d304fa05ef7de14

[5] S. Strache, “System approach for an embedded accelerometer read-out circuit,” Master’s thesis, RWTH-Aachen University, 2009.

[6] M. A. Green, K. Emery, Y. Hishikawa, and W. Warta, “Solar cell efficiency tables (version 36),” *Prog. Photovolt: Res. Appl.*, vol. 18, no. 5, pp. 346–352, 2010. [Online]. Available: http://dx.doi.org/10.1002/pip.1021

[7] N. Reich, W. van Sark, E. Alsema, S. Kan, S. Silvester, A. van der Heide, R. Lof, and R. Schropp, “Weak light performance and spectral response of different solar cell types,” in *20th European Photovoltaic Solar Energy Conference*, 2005.

[8] P. J. Hall and E. J. Bain, “Energy-storage technologies and electricity generation,” *Energy Policy*, vol. 36, no. 12, pp. 4352 – 4355, 2008, foresight Sustainable Energy Management and the Built Environment Project. [Online]. Available: http://www.sciencedirect.com/science/article/B6V2W-4TPND6D-1/2/ba98bed46553588697697d0b58c0a80f

[9] F. Simjee and P. Chou, “Efficient charging of supercapacitors for extended lifetime of wireless sensor nodes,” *Power Electronics, IEEE Transactions on*, vol. 23, no. 3, pp. 1526–1536, May 2008.

[10] P. Wagner, *Photovoltaik Engineering: Handbuch fr Planung, Entwicklung und Anwendung (VDI)*. Springer, Berlin, August 2008, vol. 2.

[11] D. Daly and A. Chandrakasan, “An energy-efficient ook transceiver for wireless sensor networks,” *Solid-State Circuits, IEEE Journal of*, vol. 42, no. 5, pp. 1003–1011, May 2007.

[12] D. Fang, H. Qu, and H. Xie, “A 1 mw dual-chopper amplifier for a 50-$\mu g\sqrt{Hz}$ monolithic cmos-mems capacitive accelerometer,” in *VLSI Circuits, 2006. Digest of Technical Papers. 2006 Symposium on*, 0-0 2006, pp. 59–60.

[13] N. Yazdi, H. Kulah, and K. Najafi, “Precision readout circuits for capacitive microaccelerometers,” in *Sensors, 2004. Proceedings of IEEE*, Oct. 2004, pp. 28–31 vol.1.

[14] S. Strache, R. Wunderlich, D. Droste, and S. Heinen, “Matlab/simulink model of a mems accelerometer read-out circuit,” in *Proceedings of the EUROSENSORS XXIV September 5-8, 2010, Linz, Austria.* Elsevier BV, September 2010, pp. 1–4, iSSN 1877 7058.

[15] J. Wu, G. Fedder, and L. Carley, “A low-noise low-offset capacitive sensing amplifier for a 50-$\mu\text{g}/\sqrt{\text{Hz}}$ monolithic CMOS MEMS accelerometer,” *Solid-State Circuits, IEEE Journal of*, vol. 39, no. 5, pp. 722–730, May 2004.

Describing Non-selective Gas Sensors Behaviour via Logical Rules

Vincenzo Di Lecce, Marco Calabrese
DIASS
Politecnico di Bari
Taranto, Italy
v.dilecce@aeflab.net, m.calabrese@aeflab.net

***Abstract* - This work presents an ongoing research aimed at interpreting the responses of non-selective gas sensors (such as metal oxide resistive ones) in terms of simple IF THEN rules. In particular, it is shown how a logical combination of the output of three extremely low-cost sensors, namely MQ131, MQ136 and TGS2602, can be arranged to produce IF THEN inferences able to discriminate among CO, SO_2 and NH_3 emissions. The outcome is quantitatively similar to that obtained with high-selective and costly chemical sensors. The experimental results, albeit grounding on an empirical base, seem to support the idea that smart compositions of low-cost sensors are able to manifest surprising discrimination abilities.**

Keywords - gas discrimination, low-cost sensors, IF THEN rules.

I. Introduction

Electronic noses are progressively disseminating throughout both the business-to-business and business-to-consumer market for the number of application domains they spawn. Alcohol testers, air quality monitoring stations [1] food ripeness detectors [2] and even lung cancer sniffers [3] are all examples of this progression. The total cost of such devices is made of two chief components: signal acquisition, processing and transmission unit on the one hand and sensing unit on the other. The latter may significantly affect the final price when high-selective responses, precise and accurate performances are required. Of course, choosing cheaper solutions entails a conundrum.

Low-cost oxide-based resistive sensors are well known to be sensitive to a wide spectrum of gases and air contaminants (this, in principle, is not a drawback); however, their selectivity is generally low [4] thus providing an ambiguous response in terms of individual components of the gas mixtures. Consequently, if high selectivity is needed by the application, the only commercially available option is to buy extremely expensive sensors.

This work presents some empirical results aimed at validating the following hypothesis: it is possible to discriminate among different gas emissions by logically combining the output of low-cost sensors appropriately. The outcome should be as much close as possible to that obtained by employing high-selective and costly chemical sensors.

Therefore, wide-spectrum sensitivity is turned into an advantage because it allows one to provide a multi-detection device at extremely low-cost. Commercial prototyping of this kind of device is currently involving the set of three sensors MQ131, MQ136 and TGS2602 to discriminate among CO, SO_2 and NH_3 emissions.

Paper layout is organized as it follows. Section II overviews the principal approaches published in the literature to tackle with gas discrimination problems. Section III presents two different approaches to gas discrimination, i.e., classification and disambiguation; the latter is used in this paper and analyzed from the perspective of logics. Then, the experimental setting and the proposed approach are described in Section IV; finally, conclusions are drawn in Section V.

II. Related Work

Several techniques have been proposed in the literature to address the problem of low selectivity in low-cost sensors. With reference to tin oxide chemical sensors, two typical measurement strategies are employed [5]: multi-sensor arrays or dynamic measurements based on a single sensor. The latter mainly includes modulation in operating temperature by pulsed or oscillated heating.

A lot of works can be found on temperature modulation in semiconductor gas sensing [6-8]. The modulation is well known to provide more information than static measurement with a mode of a constant operating temperature. Sears et al. [9] suggest several advantages that can arise from the application of a heating voltage pattern. Among all, due to the different reaction rates of various analyte gases at different temperatures, a cyclic temperature variation can be used to characterize unique signatures for each gas.

Unfortunately, temperature modulation requires an intense (hence expensive) pre-calibration phase because of different kinds and patterns of temperature modulation that have to be tried in order to minimize cross-sensitivity effects. For this reason, some sort of signal post-processing is generally applied.

Signal processing attempts to extract information hidden in raw data attenuating the effect of the different sources of noise that can occur during measurements such as cluttered or dynamic background [10]. In [11], a systematic approach for automatic signal processing, evaluation and optimization of gas sensors with temperature cycles is proposed.

In very recent times, signal processing has been coupled with multi-array sensor setups. Initial promising results for example have been obtained in real-time breath monitoring applications with micro-sensor arrays [12]. In that case, the signal processing technique is based on standard statistical dimensionality reduction and classification algorithms.

Other approaches employ gray-box models for predicting metal oxide sensor response as in [13]; in this case selectivity enhancement is reached thanks to parametric models. This, however, requires a specific knowledge of the underlying electrochemical and thermodynamic aspects.

III. Different Approaches to Gas Discrimination

The problem of correctly attributing low-cost sensor response to one gas from the list of the putative ones can be handled from multiple perspectives. Two of these are considered in the following: classification and signal disambiguation.

A. Classification-based approaches

In principle, the task of assigning a class label to a pattern can be viewed as a classification problem [14], hence the expected output is a classifier that minimizes inter-class and intra-class distance. However, several riddles hinder the correct classification of gases when low-cost sensors are employed. For example: provided timesheets often show an underestimation of the number and the range of sensed gases; furthermore, sensor drift determines non-steady behavior even when similar emission stimuli are supplied thus invalidating the calibration phase.

A high desirable condition is that function boundaries allow for proper discrimination among output classes. In other words, linear separability should be always pursued to avoid misclassification.

Starting from the work of Rumelhart and McClelland [15] it is well-known that 3-layer feed-forward networks are capable of forming any possible complex decision boundary (the so-called property of 'Universal Approximation'). For this valuable property, a number of reference works in the literature applying neural networks as a computational tool for gas discrimination [16][17][18][19][20][21].

Notwithstanding, linear separation of data in case of gas mixtures can be hardly obtained. In fact, to achieve perfect classification, all the possible concentration combinations should be exploited, which is impractical and in contrast with the objective of an affordable sensor price, at least from the manufacturer's point of view. For this reason, metal oxide sensors seem unable to produce a true quantitative information of gaseous concentrations, especially in normal operating conditions.

B. Disambiguation-based approaches

In a recent paper [22], the problem of cross-sensitivity has been accounted from a different viewpoint. It has been considered as a disambiguation process driven by algorithmic rules that come from the observation of the sensor datasheets and simple hypotheses on sensor behavior. The basic idea grounds on the hypothesis that, if the same gas is actually measured by two or more sensors, then their estimated concentrations will be similar, with an accuracy related to the number of concordant sensors. The same consideration can be drawn for every possible gas detected by the sensors so that a simple ranking strategy is applied. If the level of agreement among sensors about a supposed measured gas is above a certain confidence threshold, then the gas is considered to be a good candidate for the disambiguation process.

C. Blending classification and disambiguation approaches

The current paper tries to take benefit from the two above-mentioned approaches of gas discrimination as either a classification or disambiguation problem. The proposed one deals in fact with both separability (hence classification) and semantics (hence disambiguation). To achieve this goal, logical combinations of the output of three extremely low-cost sensors are employed. They represent a simple means to characterize sensor behaviors in intuitive linguistic terms, as it happens with fuzzy logic [23] descriptions. In the following, the proposed approach is presented in more detail.

IV. Proposed Logic-Based Approach

It is fair to assume that any empirically-driven scientific methodology is based on the following steps:

1. observations and pre-processing (i.e., of measurable signals);
2. theoretical hypothesis formulation from observations;
3. consequent hypothesis validation;
4. theory formulation (hypotheses become verified rules);

If applied with care (possibly after several iterations) the steps above should lead to some verifiable (inductive) inference (i.e., a theory) about the phenomenon under scope. Hence, the result is a *description* of the analyzed phenomenon after some kind of classification and validation phase. It is noteworthy that none of the traditional Computational Intelligence [24] techniques such as neural networks or fuzzy logic seem to provide a complete coverage of all the previous points at the same time.

In [25], a heuristic for extracting IF THEN rules form signal measurements has been presented with reference to temperature time-series analysis. In this paper, the same heuristic is used for gas discrimination.

IF THEN rules are widely used in expert systems [26] for representing knowledge in a structured and logical way; their application to the field of sensor measurements is indeed quite a novel engagement in the literature.

The obtained rules are considered as "self-descriptors" of the observed signals since they manifest knowledge in the form of logical implications built upon numerical hypotheses on the input data without any external knowledge source available. Although quite at an early stage, this simple methodology has a relevant aspect: the output (i.e., rules) is built over the same alphabet of the input space (signals) by means of numerical hypotheses. Alternatively speaking, input-output mapping is performed through measurable hypotheses. At the end of the validation process, verified hypotheses become IF THEN rules, hence they provide a (logical) description of the observed input.

Since such methodology is general, it can be employed for gas discrimination, which is the aim of this paper. Consequently, numerical hypotheses applied over low-cost gas sensor output signals can be used to infer on the kind of gas actually being sensed. In other words, given *N* sensors as input, the expected output should be a set of IF THEN rules of the type:

IF *hyp(sensor$_1$)* **AND** *hyp(sensor$_2$)* … **AND** *hyp(sensor$_N$)*

THEN *gas$_x$* **OR** *gas$_y$* $_{…}$ **OR** *gas$_z$ are being measured*

where hyp() is a predicative function defined as follows:

$$hyp\left(sensor_x\right)=\begin{cases}1 & \text{if hyp is verified for sensor } x\\ 0 & \text{otherwise}\end{cases}$$

A pictorial representation of the proposed approach is drafted in Figure 1.

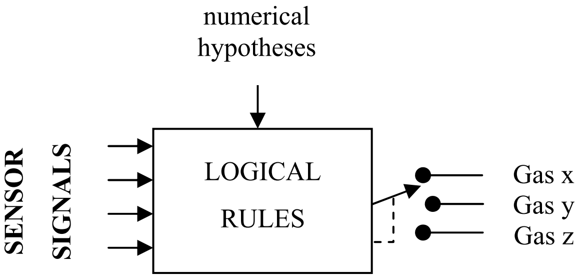


Figure 1. Conceptual schema of the proposed approach.

A. *Experimental setting*

The experimental setting has been built around three extremely low-cost sensors (see Figure 2 for more details), namely: Hanwei MQ131 and MQ136, Figaro TGS2602.

In all these sensors, the sensing material is a metal oxide semiconductor (generally tin oxide). When the sensing layer is heated at a certain temperature in the air, oxygen is adsorbed on the crystal surface with a negative charge. As quoted in [6] by withdrawing electron density from the semiconductor surface, adsorbed oxygen gives rise to Schottky potential barriers at grain boundaries, and thus increases the resistance of the sensor surface. Reducing gases decrease the surface oxygen concentration and thus decrease the sensor resistance. The overall process causes a decrease in the resistance R_s of the sensing layer that can be measured against a standard value R_0 gathered at optimal test condition. Sensor datasheets are given as R_s/R_0 values against part-per-million (ppm) concentrations.

Experiments have been carried by directly exposing the device acquisition unit to small quantities of different gaseous contaminants. Emissions have been produced in sequence to stimulate subgroups of the chosen sensor triplet. Emissions have been the following: first, carbon monoxide (CO); second, sulfure dioxide (SO_2); third, a mix of the first two ($CO+SO_2$); fourth, ammonia (NH_3).

Actually, we purposely did not use any test chamber for the experiment since our objective was mainly to discriminate among classes of emissive phenomena rather than exactly computing the ppm values of the induced gases. All emission events have been tagged with a timestamp. This assured a correct synchronization between the emitted gases and the observed sensor responses.

B. *Pre-processing*

A brief sequence of pre-processing steps has been applied over raw data, namely: 1) normalization; 2)extraction of the first derivative; 3) extraction of the absolute value.

In the normalization step, the dataset is transformed so that each signal is brought at mean zero with unary standard deviation. This allows for comparing the dynamics of the signals on the same scale. The consequence is the loss of absolute values, which is however outside the scope of the paper and is left to a future work on the subject.

In the second processing step, sensor dynamics is emphasized by considering the first derivatives as relevant features in the signal characterization process.

In the last processing step, absolute values obtained from the previous step are taken, so that oscillatory behaviors of the first derivatives are eliminated.

The effect of the three processing steps on input data is displayed in Figure 3.

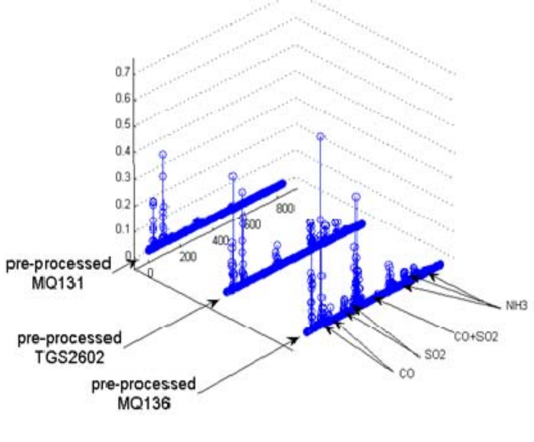


Figure 3. Synopsis of the three signals after all the pre-processing steps.

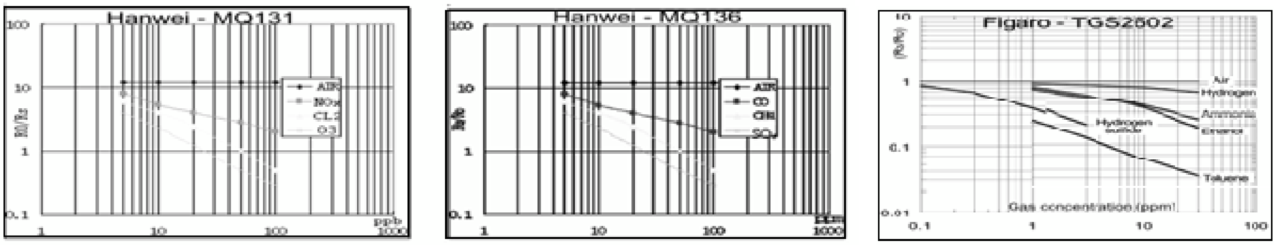


Figure 2. Datasheet of the employed sensors.

C. Defining numerical hypotheses over the dataset

After the pre-processing step, data are given in input to the logical block for gas selection.

Logical rules are defined by means of numerical hypotheses on data, as discussed above. The chosen hypothesis is a very simple one: it is defined by the following function:

$$hyp\big(datum(k)\big)=\begin{cases}1 & \text{if } \mathrm{datum}(k)\geq\vartheta\\ 0 & \text{otherwise}\end{cases}$$

where datum(k) is an input sample at time k and θ is a parametric threshold.

It is interesting to note that, depending on the value of θ, hypotheses can be true or false, i.e., below or above certain levels the hypothesis function may switch from one logical state to another. This means that an IF THEN rule of the type defined above can be verified only in certain subsets of the dataset.

Furthermore, it is important to stress that the logical state (true or false) is obtained only by means of *measures*, without explicitly knowing the analytical form of the underlying datum function. This allows for analyzing, in principle, any given measurable signal.

Parameter θ provides one degree of freedom to the definition of the logical block. Of course, more than one parameter can be considered; however, for the sake of simplicity, only one parameter is taken in this paper.

D. Finding the best parametric configuration

The choice for optimal θ has been led by the estimation of another parameter, referred to as the *coverage index* (CI).

The coverage index of an IF THEN rule is simply defined as the number of samples where the rule is verified divided by the dataset cardinality. In formulae:

$$\mathrm{CI}(Rule_i)=\frac{\#\text{samples where } Rule_i \text{ is verified}}{|\,\text{Dataset}\,|}$$

Similarly, the *total coverage index* (TCI) represents the fraction of samples in the dataset covered by at least one of the IF THEN rules representing the knowledge base.

TCI varies from 0 (no rule) to 1 (when rules completely partition the dataset). TCI can be reckoned by means of the following formula:

$$\mathrm{TCI}(KB)=\sum_{i=1}^{|KB|}\bigcup_{\text{Dataset}}\mathrm{CI}(Rule_i)$$

There can be samples in the dataset firing more than one rule; this means that there can be rule activation patterns that partially or totally overlap over the dataset.

Our attempt was to empirically find the minimum number of rules (depending on parameter θ) with the total coverage index closest to 1. For this to be achieved, an iterative procedure has been run varying the value of θ. Good results have been found for θ = 0.3 with TCI=1 obtained by means of the four rules expressed in Table 1, while an image showing all four rules activation patterns is framed in Figure 4.

TABLE I. LOGICAL RELATIONSHIPS FOUND AMONG SENSORS WITH THRESHOLD PARAMETER = 0.3. N.A. STAYS FOR NOT AVAILABLE. A MINUS SIGN IN APEX INDICATES LOW CREDIBILITY

Rule ID	**KNOWLEDGE BASE (for *θ=0.3*)**							
	IF STATEMENT			***THEN STATEMENT***				***CI %***
	MQ 131	***TGS 2602***	***MQ 136***	***SO_2***	***NH_3***	***CO***	***other***	
1	F	F	T	T	T^-	T^-	N.A.	4.9
2	F	T	F	N.A.	T	N.A.	N.A.	1.4
3	T	T	T	N.A.	N.A.	T	N.A.	2.7
4	F	F	N.A.	T	N.A.	N.A.	T	93.7

Each row from Table I represents a true IF THEN rule. Antecedents (i.e., arguments of the IF statement) are hypotheses on samples coming from sensors; consequents (i.e., arguments of the THEN statement) are True/False values over the hypothesis that a certain gas is actually being sensed. These last hypotheses have been verified in a supervised manner by windowing the emission events around the tagged timestamp of the experiment phase. For any given rule, if the number of firing events in the window of a gas emission was too low (less than 5% of the window length) then the predicate variable was considered insensitive to that gas. If the number of firing event was comprised between 5% and 50% of the window length, then the predicate value was assigned a low credibility.

To be more clear, the first row of Table I represents the following statement:

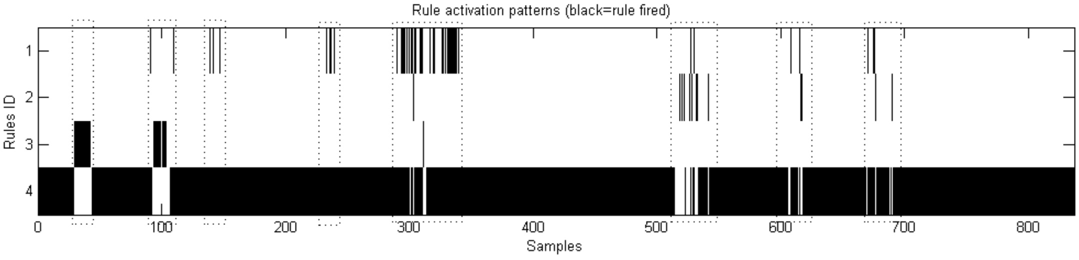


Figure 4. Rule activation patterns (rules correspond to logical relationship reported in Table I).

Rule 1:

IF MQ131< 0.3 **AND** TGS2602< 0.3 **AND** MQ136 >= 0.3

THEN SO_2 **OR** NH_3 **OR** (with low credibility) CO (with low credibility)

This means that every time the (preprocessed) values of MQ131 and TGS2602 are below 0.3 and those of MQ136 are above or equal to 0.3 then all the three gases are possibly being sensed: SO_2, NH_3 or CO, having the latter two cases a low credibility.

Rules may also provide very poor pieces of information like in **Rule 4:**

IF MQ131< 0.3 **AND** TGS2602< 0.3

THEN SO_2 **OR** something unknown

where the interpretation is that if both MQ131 and TGS2602 values are below 0.3 then *either* we are in presence of SO_2, *or* we are not capable of ascribing sensor behavior to one of the considered gases.

Of course, having multiple gases in output is not a desirable condition since it represents an ambiguous response. Notwithstanding, disambiguation can be partially dealt with by means of simple considerations.

E. Disambiguating sensor response

Rule 2 and 3 account respectively for NH_3 and CO. This means that they do not need further disambiguation. If we want our system to be able to detect also SO_2 emissions, some kind of rule post-processing has to be carried out. In particular, Rule 1 can be simplified assuming to disregard NH_3 and CO response due to their low credibility.

Of course, this is only an empirical approach aimed at showing the problem of disambiguation at a coarse scale. More formal approaches go beyond the scope of the paper; for example, the reader may refer to a previous work [22].

F. Testing results with high-cost chemical sensors

For testing purposes, the discrimination abilities of the logic block has been compared with the output of high-cost chemical sensors. In particular, two SensoriC sensors for CO and SO_2 detection respectively have been used for this aim. These two sensors are approximately between one and two orders of magnitude more costly than the low-cost ones. Their response to the events of CO and SO_2 is depicted in Figure 5.

G. Dataset heteroschedasticity

In order to have a further estimate of the inner correlation among sensors, dataset heteroschedasticity (i.e., the property of measured samples to represent a population with equal variance) has been assessed through the Barlett's test [27]. The test, computed on raw data coming from the triplet, has shown that (with reference to the proposed experiment) the number of dimensions necessary to explain the non-random variations in data is 3. The same result has been obtained by means of the principal component analysis [28].

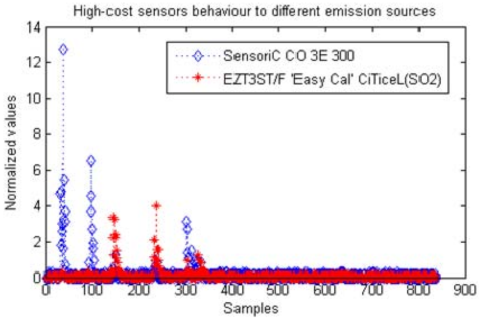


Figure 5. Rule activation patterns
(rules correspond to the logical relationships reported in Table I).

This assessment defines the putative minimum number of IF THEN rules needed to best explain our data. As shown before, four IF THEN rules were empirically found.

H. Theoretical aspects and future developments

Since IF THEN rules are logical statements, they guarantee logical coherence in their respective domain of validity. For example, assuming that "IF A AND B THEN C" is true in a certain interval, this implies that, in the same interval, "IF A AND B THEN NOT C" cannot be verified. In other words, the proposed logical approach to signal interpretation, provides a coherent framework for gas discrimination.

Another key point, is the type of numerical hypothesis to apply over incoming data. For the sake of simplicity, only one parameter has been used for tuning purposes. However, it is fair to assume that, the more parameters used, the more discriminatory the type of rules found.

Starting from these observations, it is interesting to note that the proposed approach stays amid a wide number of fields, such as measurements, computational intelligence, logics. Measurements become valuators of hypotheses over data, thus allowing for a jump from the numerical world to the logic-symbolic one.

V. Conclusion

In this work, the novel hypothesis of logically combining low-cost metal oxide sensor responses by means of IF THEN inference rules has been presented for gas discrimination purposes. Observational experiments have been made to support this claim. In particular, it has been shown how a triplet of low-cost sensors (namely, MQ131, MQ136 and TGS2602) is sufficient for discriminating three different classes of emissions (CO, SO_2 and NH_3).

This paper is built upon the theoretical and practical expertise gained from previous works in the mixed fields of measurement and computational intelligence. As far as the electrochemical and thermodynamic aspects are concerned, a temperature-humidity calibration phase was performed digitally on raw data basing on the available MQ131, MQ136 and TGS2602 datasheet information and using high-sensitivity temperature and humidity sensor output as ground-truth reference. As for the information processing aspects, the computationally-lightweight rule extraction mechanism presented in [25] allowed for producing a coherent knowledge base with a very small number of valid rules (four in our case). Since the obtained rules are *logical*, when they account for conflicting behaviors, this means that they certainly do not occur in the same sampling time.

To cap it all, the noteworthy principle behind our proposal is that gas discrimination abilities gained with low-cost sensors can be surprisingly similar to that obtained with high-cost counterparts. This idea, although grounding by now only on an empirical base, seems to open a promising perspective in the research field of both measurements and intelligent information systems.

References

[1] V. Di Lecce, A. Amato, R. Dario and C. Martines, “Air quality control for health care centres. The application of an Intelligent Distributed System”. 2009 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems. EESMS 2009 Crema (Italy). pp. 27-30.

[2] Concina et al., “Early detection of microbial contamination in processed tomatoes by electronic nose”, Food Control, Vol. 20, Issue 10, pp. 873-880, October 2009.

[3] R. F. Machado et al. “Detection of Lung Cancer by Sensor Array Analyses of Exhaled Breath”, American Journal of Respiratory and Critical Care Medicine Vol 171. pp. 1286-1291, 2005.

[4] P. Moseley and B. Tofield, Solid-State Gas Sensors. Bristol, U.K. Adam Hilger, 1987.

[5] Xingjiu Huang, Fanli Meng, Zongxin Pi, Weihong Xu and Jinhuai Liu “Gas sensing behavior of a single tin dioxide sensor under dynamic temperature modulation”, Sensors and Actuators B: Chemical, Volume 99, Issues 2-3, 1 May 2004, pp. 444-450.

[6] A. P. Lee and B. J. Reedy, “Temperature modulation in semiconductor gas sensing”, Sensors and Actuators B: Chemical, Volume 60, Issue 1, 2 November 1999, pp. 35-42.

[7] A. Far, B. Guo, F. Flitti and A. Bermak, “Temperature modulation for tin-oxide gas sensors”, Proc. of the 4th IEEE International Symposium on Electronic Design, Test and Applications, 2008, pp.378-381.

[8] M. Roth, R. Hartinger, R. Faul and H.-E. Endres , “Drift reduction of organic coated gas-sensors by temperature modulation”, Sensors and Actuators B: Chemical, Volume 36, Issues 1-3, October 1996, pp.358-362.

[9] W.M. Sears, K. Colbow and F. Consadori, “Algorithms to improve the selectivity of thermally cycled tin oxide gas sensors”, Sensors and Actuators, (19) 1989. pp.333–349.

[10] E. Llobet et al., “Multicomponent gas mixture analysis using a single tin oxide sensor and dynamic pattern recognition”, IEEE Sensors Journal, 2001, pp. 207 – 213.

[11] T. Fricke, P. Reimann, S. Horras, E. Leonhardt, P. Sahm and A. Schutze, “A systematic approach for the automatic signal processing, evaluation and optimization of T-cycled gas sensors”, Proc. of the IEEE Instrumentation and Measurement Technology Conference, 2008. IMTC 2008., pp. 1940 – 1945.

[12] K.D. Benkstein, B. Raman, C.B. Montgomery, C.J. Martinez and S. Semancik, “Microsensors in dynamic backgrounds: toward real-time breath monitoring”, DOI 10.1109/JSEN.2009.2035738, 2009, pp. 137 – 144.

[13] A. Fort, M.B. Serrano-Santos, R. Spinicci, N. Ulivieri and V. Vignoli, “Electronic noses based on metal oxide gas sensors: the problem of selectivity enhancement”, Proc. of the 21st IEEE Instrumentation and Measurement Technology Conference, 2004, pp. 599 – 604.

[14] V. Vapnik (1979), Estimation of Dependencies Based on Empirical Data. Nauka, Moscow, (In Russian). English translation. New York. Springer Verlag, 1982.

[15] David Rumelhart and J. L. McClelland, Parallel Distributed Processing: exploration in the microstructure of cognition, MIT Press, MA, 1986.

[16] Far, A.; F. Flitti, B. Guo and A. Bermak, Gas identification system based on temperature modulation tin-oxide sensors and bio-inspired processing, 15th IEEE International Conference on Electronics, Circuits and Systems, 2008, pp. 1010 – 1013.

[17] S. Marco, A. Pardo, A. Ortega and J. Samitier, Gas identification with tin oxide sensor array and self organizing maps: adaptive correction of sensor drifts, IEEE proceedings of the Instrumentation and Measurement Technology Conference, 1997. IMTC/97 'Sensing, Processing, Networking', pp. 904 – 907.

[18] B.W. Jervis, J. Desfieux, J. Jimenez and D. Martinez, Quantification of gas concentrations in mixtures of known gases using an array of different tin-oxide sensors, IEE Proceedings of Science, Measurement and Technology, Vol. 150 (3), pp. 97 – 106.

[19] M. Ambard, Guo Bin, D. Martinez and A. Bermak, A Spiking Neural Network for Gas Discrimination Using a Tin Oxide Sensor Array, 4th IEEE International Symposium on Electronic Design, Test and Applications, 2008, pp. 394 – 397.

[20] R. Kumar, R.R. Das, V.N. Mishra and R. Dwivedi, A Radial Basis Function Neural Network Classifier for the Discrimination of Individual Odor Using Responses of Thick-Film Tin-Oxide Sensors, IEEE Sensors Journal, 2009, 9(10): 1254 – 1261.

[21] Jung Hwan Cho, Chang Hyun Shim, In Soo Lee, Gi Joon Jeon, On-line monitoring of indoor environmental gases using ART2 neural networks and multi-sensor fusion, Proceedings of the Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004, pp. 125 – 129.

[22] V. Di Lecce, M. Calabrese and R. Dario, Computational-based Volatile Organic Compounds discrimination: an experimental low-cost setup, Proc. of the 2010 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA2010), 2010, pp. 54-59.

[23] Zadeh L., “Knowledge representation in fuzzy logic”, IEEE Trans. on Knowledge and Data Engineering, Vol. 1, pp. 89-100, 1989.

[24] N.J. van Eck, L. Waltman, J. van den Berg and U. Kaymak (2006), Visualizing the computational intelligence field, IEEE Computational Intelligence Magazine, 1(4):6-10.

[25] M. Calabrese, Self-Descriptive IF THEN Rules from Signal Measurements, A holonic-based computational technique, 2010 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA 2010), 2010, pp. 102-106.

[26] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 2nd ed., Prentice Hall, 2003.

[27] G. W. Snedecor and W. G. Cochran, Statistical Methods, Eighth Edition, Iowa State University Press, 1989.

[28] J. Shlens, A Tutorial on Principal Component Analysis. Institute for Nonlinear Science, UCSD, 2005.

TinyMQ: A Content-based Publish/Subscribe Middleware for Wireless Sensor Networks

Ke Shi and Zhancheng Deng
School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, China
keshi@mail.hust.edu.cn

Xuan Qin
Research and development Office
Huazhong University of Science and Technology
Wuhan, China
kjcqx@mail.hust.edu.cn

Abstract—**The emergence of Wireless Sensor Networks (WSN) technologies gave rise to the need for abstraction mechanisms that can simplify the data communication tasks. Under this respect, the publish/subscribe paradigms play an important role since they can be used to abstract the WSN in terms that are closer to the needs of applications designed mainly to detect and notify upon events of interests. In this paper, we propose TinyMQ, a content-based publish/subscribe middleware for wireless sensor network. In TinyMQ, an overlay network is constructed on top of the underlying WSN, in which sensor nodes can be logically connected independent of their geographical position. Over this overlay network, message mapping and routing schemes are presented to implement rendezvous-based publish/subscribe. The simulation study demonstrates TinyMQ is efficient in terms of both costs and time.**

Keywords-publish/subscribe; content-based; wireless sensor network

I. INTRODUCTION

In WSNs (Wireless Sensor Networks), the sensor nodes cooperatively sense the environment and send the sampled data to the base station. Since many nodes may transmit a large amount of data synchronously, how to collect, transport, and process the data efficiently is the main research domain of the WSN applications [1]. Event-driven publish/subscribe is considered as an efficient method for data collecting and transmitting tasks in WSN. It is because data is sensed and transmitted only when certain events occur. Compared with the request/response model in which information exchange occurs by issuing requests and waiting for corresponding responses, publish/subscribe model can reduce the network overload significantly [2].

Implementing an efficient publish/subscribe mechanism in the WSNs is challengeable [3]. WSN is an ad-hoc, self-organized and totally distributed network. The subscriber nodes, which are interested in some events, do not know whether, when and where the events may occur. And the publisher nodes do not know which nodes are interested in these events either. Publish/subscribe mechanism in WSN must be distributed. Moreover, the WSN is resource-limited (in terms of computation ability, storage ability, and communication ability), event mapping and routing policies should be carefully designed to achieve publish/subscribe functionality and prolong the life of the WSN as much as possible.

Rendezvous-based model is a frequently used model for distributed publish/subscribe. In this model, the queries and events are sent to a set of selected nodes called rendezvous nodes that act as the brokers where the interests of the subscribers and publishers match. The mapping and routing policy should guarantee that there exists a rendezvous node that receives both the query and its matched event. When using predetermined physical node IDs, a query or event may be mapped to an ID that does not exist because the node that initiates the query or event has only local network information. Location-based method maps the query or event to a set of geographic locations and publishes it at the nodes nearest these locations. However, most sensor nodes are deeply embedded. It is impossible from them to integrate GPS-like positioning devices. And most existing position algorithms for WSN consume a lot of computing and energy resource [4].

Gossip-based approach does not require location information. In this model, each query or event is spread throughout the network by probabilistic broadcasting [5]. The queries and events are sent to almost all the nodes, which guarantee the high probability that the interests of subscribers and publishers match. However, since each query or event must be sent to almost all the nodes, network overload may be very high and serious congestion may occur frequently.

A more efficient mechanism that does not require location information is needed to address this problem. Gossip-based approach does not concern the content of certain query/event and sent a query/event message randomly. To guarantee that an event meets every query, excessive transmission becomes necessary. It causes unnecessary query and event forwarding in an unstructured network environment. We think, with a carefully designed overlay network (structured network environment), a query/event message can be forwarded more efficiently based on content-related mapping and routing. We propose TinyMQ, a content-based publish/subscribe middleware for wireless sensor network. In TinyMQ, an overlay network is constructed on top of the underlying WSN, in which sensor nodes can be logically connected independent of their geographical position. The unique keys represented by the binary strings chosen from *{0, 1}* is used as the logical addresses of in the proposed overlay network, which enables hash based content-related message mapping and routing. This mapping mechanism guarantees the events meet the queries in the certain rendezvous nodes.

The rest of this paper is organized as follows. In Section 2, we review the related researches. Section 3 describes the details of TinyMQ including overlay maintaining, message (event and query) mapping and routing. In Section 4, the performance of the proposed scheme is evaluated. Finally, in Section 5, we present the conclusion and future works.

II. RELATED WORK

Several publish/subscribe approaches in WSN have been proposed in the literature. This section introduces these existing approaches briefly.

The Mires [6] middleware addresses the implementation of publish/subscribe communication for wireless sensor network applications. It adopts topic-based publish/subscribe mechanism that can reduce the number of transmissions and energy consumption because only the messages referring to the subscribed topic are sent to a sink node. In Mires, the publisher nodes must know which nodes the events are sent to and which paths the events are followed. The implementation detail and performance evaluation are not given either.

TinyDDS [7] is a lightweight implementation of the OMG DDS Specification Standard in WSNs. TinyDDS can adaptively perform event publication according to dynamic network conditions and autonomously balances its performance among conflicting objectives. It leverages an evolutionary multi-objective optimization mechanism to seek the optimal tradeoffs among objectives and adjust parameters in its event routing protocol. However, topic-based DDS is designed for the traditional distributed system that has a lot of difference with WSN. The main goal of TinyDDS is to provide the interoperability between WSNs and access networks. The goal of TinyMQ is to provide the interoperability among the nodes in a WSN.

PUB-2-SUB [8], a publish/subscribe framework for P2P networks, is based on two key design components: the virtualization component and the indexing component. The virtualization component maintains a naming structure by assigning to each node a unique virtual address. A tree-like overlay network is constructed and maintained. The indexing component determines the corresponding subscription and notification paths for given queries and publications, in which routing is based on the virtual addresses of the nodes. PUB-2-SUB^{+} is an extension of PUB-2-SUB for WSNs [9]. It is location-free and content guide. However, load balance is not considered in PUB-2-SUB^{+}, which can cause high throughput and even congestion in some region where many events and queries are directed, such as the region near the root nodes. The nodes in this region may exhaust their energy and crash down quickly.

Compared with PUB-2-SUB+, TinyMQ partitions the whole network into multiple trees. A query/event is mapped to multiple nodes in the different trees. It facilitates load-balancing and high-availability. TinyMQ also allows cross-tree routing, which relieves the burden of root nodes.

III. TINYMQ

TinyMQ is a content-based publish/subscribe middleware for WSNs, which employs a general message mapping and routing mechanism on top of an overlay network to provide scalable and adaptable publish/subscribe. This section will explain how TinyMQ solves the existing problems.

A. Architecture

In TinyMQ, a sensor node can act both as a producer and a consumer of information, playing the role of publisher and subscriber, respectively. Publishers and subscribers exchange messages in form of events and queries.

TinyMQ consists of two layers: overlay layer and pub/sub layer. The overlay layer maintains a naming structure by assigning to each node a unique virtual address. The sensor network is organized in a logical topology despite node failures and network churn. Publish/subscribe layer provides message mapping and routing functionality for event subscription, event publication and event notification. The message mapping and routing mechanism determines the corresponding subscription and notification paths for given queries and events, in which routing is based on the virtual addresses of the nodes.

For events and queries, the initiated nodes select the rendezvous nodes through message mapping and sent corresponding messages to these selected rendezvous nodes through message routing. The rendezvous nodes check the events against the queries in order to determine whether dispatching the events to the subscribers or not.

B. Overlay network

We assume that the sensor network is a connected graph of *n* stationary nodes $\{S_1, S_2, ..., S_n\}$ and that there are one or more nodes $\{S_1^*, S_2^*, ..., S_m^*\}$ ($m \geq 1$) that are reliable like the sink nodes or base stations. Two nodes are called "neighbors" if they can communicate directly with each other.

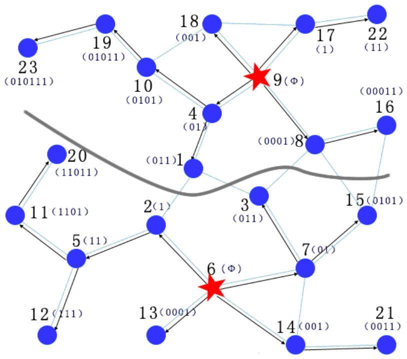


Figure 1. The overlay network with two root nodes (6, 9)

Each node is assigned a binary string chosen from *{0, 1}* called key value to indicate its logical address, denoted by *key*(S_i). Using 0 or 1 symbols facilitates designing hash-based mapping mechanism.

TinyMQ maintains a multi-tree overlay on top of this kind of network, in which the stable sink nodes serve as root nodes and initiate the overlay constructing process. Every node belongs to only one tree. The nodes belonging to the different trees may have same key value, which means more nodes can act as rendezvous nodes for certain query/event message.

The form and structure of logic address is similar with PUB-2-SUB$^+$. The key value of S_i, a child of S_i*, is the shortest string of key(S_i*) + ‘0*1’ unused by any other child node of S_i*. However, PUB-2-SUB$^+$ maintains only one tree across the whole network, which may lead to serious congestion near the root node. Although PUB-2-SUB$^+$ claims that it can build multiple trees to relieve the congestion, each tree must contain all the nodes. It will cause large coordinating and maintaining overhead. In TinyMQ, there are *m* trees rooted by S_i* across the whole network. Each node only belongs to one tree. The network is partitioned into *m* tree-based clusters. Therefore, the possible congestion near the root node can be avoided and better load balance can be achieved. An example of this kind of overlay is illustrated in Fig. 1, where the network comprises of 23 nodes, two of which (node 6 and node 9) serve as the root nodes.

Given a node *X*, the key zone of *X*, denoted by *kzone*(*X*), is defined as the set of all the binary strings, of which *key*(*X*) is the longest prefix. In Fig. 1, the key zone of root 6 is *kzone*(6) = {‘0’, ‘00’, ‘000’, ‘0000’, ‘0000*’}, and for a normal node 10, *kzone*(10) = {‘0101’, ‘01010’, ‘01010*’}.

In this multi-tree overlay, each node *X* has a *state*, represented by *info*(*X*) = < *key*(*X*), *root*(*X*), *dist*(*X*) >, where *key*(*X*) is the key value of node *X*, *root*(*X*) is the root of tree which *X* resides, and *dist*(*X*) = < *dist*(*X*, S_1*) , *dist*(*X*, S_2*) , …, *dist*(*X*, S_m*) > is the vector of shortest distances from the root nodes to *X*. The distance is measured by the number of hops. If X is the root node, let key(X) = φ and *root*(*X*) is itself. The key value is used to route messages inside a tree, whereas the distance information is used to route messages from a tree to another. Exception for the root nodes, all the above information is initially unknown for every node but will eventually be filled as nodes exchange information with their neighbors. Specifically, once a node *X* updates its *info*(*X*), it advertises the *info*(*X*) to all its neighbors. Upon receipt of such a message, each neighbor *Y* updates its state as follows:

(1) Update distance information: set *dist*(*Y*, S_i*) = *min*(*dist*(*Y*, S_i*), *dist*(*X*, S_i*) + 1) for each root node S_i*.

(2) If *key*(*Y*) has not been set, update key value and root information,
   (a) Set root(*Y*) = *root*(*X*) (i.e., *Y* is in the same tree as *X*).
   (b) Set key(*Y*) to a binary identifier according to the Key prefix rule: *key*(*Y*) is a string (shortest length preferred) of the form *key*(*X*) + “0…01” unused by any other neighbor node of *X*. *X* is called the “parent node” of *Y*.

C. *Message mappings*

We now consider the message mapping and routing mechanism and analyze how to accomplish publish/subscribe in WSNs. The notations that we will use in the following sections are defined in Table 1.

TABLE 1 NOTATIONS FOR MESSAGES

Notations	Definitions
d	The number of dimensions in the event space
e	An *d*-dimensional event
$e.a_i$	A *i*-th attribute of event
Q	A *d*-dimensional subscription (query)
$Q.c_i$	A *i*-th constraint of subscription
Ω	A *d*-dimensional event space
Ω_i	Projection of Ω on *i*-th dimension
$\sum$	A *d*-dimensional subscription space

For each publish/subscribe application, we assume the interest has a fixed number of attributes called the dimension. The event is defined as some new information that a node wants to publish. The queries of interest are those that specify a lower-bound and an upper-bound on each event attribute. A query *Q* is represented as a *d*-dimensional interval vector $\{ c_1 , \dots, c_i ,\dots\}$, where $c_i = [q^l{}_i, q^h{}_i]$, $q \in \{0,1\}^k$ is the constraint on the *ith* dimension. A query subscribes to all events belonging to this interval. Each query is subscribed to all the trees and each event is published to all the trees. In PUB-2-SUB$^{+,}$ the event mapping mechanism only deals with one dimension events.

We use the key space-split mapping mechanism proposed in [10]. In this mechanism, message mapping is based on a collection of hash functions $h_i : \Omega_i \rightarrow [0, 1]^l$; h_i maps attribute values in Ω_i to bit strings of length *l*. Given the set of h_i functions, a set of hash functions H_i is defined for constraints $Q.c_i$ to return sets of *l*-length bit strings as follows: $H_i(Q.c_i) = \{h_i (e) \mid e \in \Omega_i \wedge e \text{ satisfies } Q.c_i\}$.

In order to implement the matching, TinyMQ distribute across the nodes in the system the tasks of storing subscriptions, matching events against subscriptions, and delivering notifications to subscribers. Subscriptions in ∑ and events in *Ω* should be assigned to nodes through two mapping functions: *SK*: $\sum \rightarrow 2^N$ and *EK*: $\Omega \rightarrow 2^N$.

SK(*Q*) returns a set of key values, named rendezvous key values of *Q* that indicate the corresponding rendezvous node responsible for storing *Q*, *EK*(*e*) complements *EK* by returning the rendezvous key values of *e*. The nodes with the same key values are the rendezvous nodes responsible for matching *e* against subscriptions registered in the system. So if $e \in Q$, then $EK(e) \cap SK(Q) \neq \Phi$, which we call this property as *mapping intersection rule*.

The *SK* function returns all possible concatenations of bit strings: $SK(Q) = \{s_1 \circ s_2 \circ \dots \circ s_t \mid s_i \in \{0,1\}^l \wedge s_i \in H_i(Q.c_i)\}$ which returns several rendezvous key values. To satisfy the mapping intersection rule, the *EK* mapping is defined as $EK(e) = h(e.a_1) \circ h(e.a_2) \dots \circ h(e.a_t)$, i.e., it returns a single rendezvous key value. It is base on the idea of cascade hashing.

For example, in a monitoring application for cold chain logistics, a query is $Q = \{0 <$ *temperature* < 0.5， $0.85 <$

humidity < 0.9}, and a event is e = {*temp* = 0.25，*humidity* = 0.88}, the mapping process is

$h_1=(\lfloor|temperature|\cdot 10\rfloor)_2$

$h_2=(\lfloor|humidity|\cdot 10\rfloor)_2$

$H(Q.c_1)=\{h_1([0,0.5])\}=\{h_1(0),h_1(0.1),h_1(0.2), h_1(0.3),h_1(0.4),h_1(0.5)\}=\{00,01,10\}$

$H(Q.c_2)= \{h_2([0.85,0.9])\}=\{h_2(0.8),h_2(0.9)\}=\{00\}$

$SK(Q)=\{ H(Q.c_1)\times H(Q.c_2)\}= \{0000,0100,1000\}$

$h(e.a_1)= h_1(0.25)=\{01\}$

$h(e.a_2)= h_2(0.88)=\{00\}$

$EK(e)= h(e.a_1) \circ h(e.a_2)=\{0100\}$.

The query is sent to the nodes with the common prefix of 0000, 0100, or 1000. The event is sent to the nodes with the common prefix of 0100. These nodes are the rendezvous nodes.

D. *Message routing*

For ease of presentation in this section, we assume: (1) query $Q = [q_l, q_h]$ or $Q = \{q_1, q_2, \ldots, q_n\}$ (as $q_x \in \{0, 1\}^m$) is an event set which consists of several events, and every event will be presented by several binary strings; (2) subscription notation <*Q,* S$_i$*> indicates that Q will send to some nodes that belong to tree rooted by S$_i$*, and analogously, event notation <*e,* S$_i$*> indicates that e will be sent and stored in some nodes that belong to the tree rooted by S$_i$*

ALGORITHM 3.**1** SUBSCRIPTION ROUTING

```
1   Initially, the subscription starts at the subscriber node of $Q$, send to the nodes of the tree Tree($S_i$*)
2   At a node $X$ that receives <$Q$, $S_i$*>
3       Quit if node $X$ already received this $Q$ before
4       If ($root(X) \neq S_i$*)
5           $Y = min\{dist(Y_i, S_i*) \mid Y_i$ is the neighbor node of node $X\}$
6           Send the subscription <$Q$, $S_i$*> to the node $Y$
7       Else
8           Let set $Z = \{str \in \{0, 1\}^k \mid Key(X)$ is a prefix of $str\}$
9           If ( $(Q - Z) \neq \Phi$ )
10              Let $NK = \{Key(Y_i) \mid Y_i$ is the neighbor node of $X$ but is not the child node of $X\}$
11              And let $M = \{< K_i, K_j, pre_{ij} > \mid$ as for $K_i (\in Q - Z)$, exists a $K_j (\in NK)$ that $pre_{ij}$ is the largest common prefix of $K_i, K_j\}$
12              Traverse the set $M$
13                  If ($pre_{ij}$ is longer than $pre_{i,parent}$)
14                      forward <$Q$, $S_i$*> to node $K_j$
15                  else
16                      forward <$Q$, $S_i$*> to parent node $K_{parent}$ of the node $X$
17          else if $((Q \cap Z) \neq \Phi)$
18              store $Q$ at $X$ if $kzone(X)$ overlap with $Q$
19              forward $Q$ to all children of $X$ in Tree($S_i$*)
```

Every node has been assigned a key value as a result of an overlay establishing process. A query can be initiated by any node in the WSN at any time. The routing algorithms to subscription (query) and publication (event) are presented below in Algorithm 3.1 and 3.2, respectively.

In the Algorithm 3.1, when the node X receives the query Q, if the node X does not belong to Tree S_i*, the subscription will be send toward Tree S_i* (Line 4-6). Otherwise, the subscription will be stored and processed in the node X (Line 18) if it overlaps the zone $kzone(X)$ of node X, or send to child nodes of node X (Line 19) if it overlaps the zone $kzone(Y)$ of the child nodes of node X, or to neighbor nodes (Line 13, 14), or to parent node (Line 15, 16) of the node X according to the largest common prefix (Line 8-12).

ALGORITHM 3.2 EVENT ROUTING

```
1   Initially, the event starts at the publisher node of $e$, send to the nodes of the tree Tree($S_i$*)
2   At a node $X$ that receives <$e$, S$_i$*>
3       If ($root(X) \neq S_i$*)
4           $Y = min\{dist(Y_i, S_i*) \mid Y_i$ is the neighbor node of node $X\}$
5           Send the event <$e$, $S_i$*> to the node $Y$
6       Else
7           If ($key(X)$ is not the prefix of $e$)
8               Let $NK = \{Key(Y_i) \mid Y_i$ is the neighbor node of $X$ but is not the child node of $X\}$
9               And let $M = \{< e, K_j, pre_j > \mid$ as for $e$, exists a $K_j (\in NK)$ that $pre_j$ is the largest common prefix of $e, K_j\}$
10              If ($pre_j$ is longer than $pre_{parent}$)
11                  forward <$e$, $S_i$*> to node $K_j$
12              else
13                  forward <$e$, $S_i$*> to parent node $K_{parent}$ of the node $X$
14          else if ($key(X)$ is the prefix of $e$)
15              find the child node $Y$ such that $key(Y)$ is a prefix of $e$
16              if ($Y$ exists) then forward $e$ to $Y$
17              else search node $X$ for those subscriptions matching $e$
```

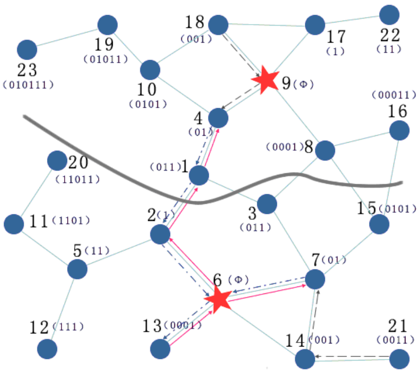


Figure 2. Message routing

In the Algorithm 3.2, at the node X receiving the event e, the event will be send toward Tree S_i* (Line 3-5) if the node X does not belong to Tree S_i*. Otherwise, the event will be stored and processed in the node X (Line 17) if it belongs to the zone $kzone(X)$ of node X, or send to child nodes of node X (Line 15-16) if it belongs to the zone $kzone(Y)$ of the child node of node X, or to neighbor nodes (Line 10, 11), or to parent node (Line 12, 13) of the node *X* according to the largest common prefix (Line 7-9).

As shown in Fig. 2, based on our proposed routing algorithms, the subscription “01” is disseminated though 13→6→7 and 13→6→2→1→4 respectively; and the event “01” is disseminated through 21→14→7→6→13 and 18→9→4→1→2→6→13 respectively. Node 7 (with the common prefix 01) is the rendezvous node in the tree rooted from node 6, and node 4 (with the common prefix 01) is the rendezvous node in the tree rooted from node 9.

IV. EVALUATION STUDY

We use OPNET simulator to evaluate TinyMQ performance. The simulated network consists of 1500 sensor nodes uniformly placed in a 500m × 200m field, each node having a communication radius of 20m.

The event space is 2-dimension and any event in this space can be mapped into a bit string whose length is less than 128. Each subscription is a random continuous event set in this event space. Every node in the network can subscribe and publish.

In the simulation, 10,000 events and 100 queries are generated and disseminated. The length of the query interval (determining the size of event set) follows Zipf’s distribution. The reason to choose Zipf’s distribution is that recent work observed a Zipf like long tail distribution [11] of object annotation and the query terms in ad hoc, self-organized applications like WSN applications.

To evaluate the efficiency of TinyMQ, we consider the following performance metrics: subscription efficiency, notification delay and effect of failure. We also compared TinyMQ with PUB-2-SUB$^+$ in terms of subscription efficiency and notification delay. PUB-2-SUB$^+$ is selected because it is also a content-based subscribe/publish middleware built on a logic overlay without location information.

To measure subscription efficiency, we compute the average number of nodes that store a query and the number of hops that the query is forwarded until reaching those nodes. Fig. 3 and Fig. 4 show TinyMQ achieves higher subscription efficiency both in terms of storage and communication especially when the number of trees is larger. In Fig. 3, the x axis represents the number of trees, and the y axis represents the number of nodes storing the query. In Fig. 4, the x axis represents the number of trees, and the y axis represents the number of hops that the query is forwarded. For example, when the number of trees is 10, the average number of nodes storing the subscription decreases from 23 to 18, and the average number of hops forwarding the query decreases from 18 to 10. It is because the trees are not overlapped in TinyMQ and distance-based inter-tree routing can balance the load.

Notification delay is computed by the equation $d_t/d_{optimal}$, where d_t is the total hop count distance from the source node to the destination node via the rendezvous node, and $d_{optimal}$ is the shortest hop count distance from the source node to the

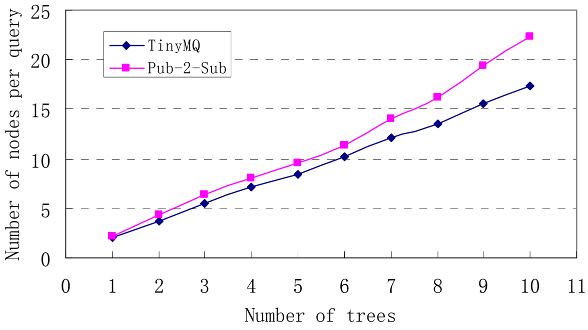


Figure 3. Number of nodes per query

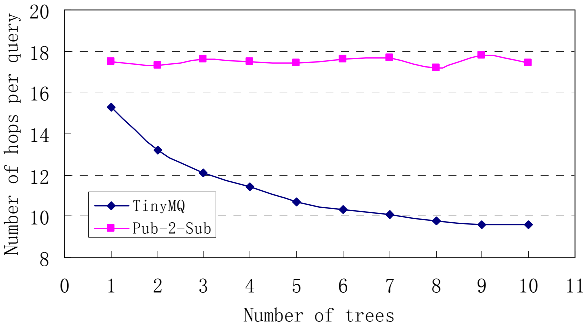


Figure 4. Number of nodes per query

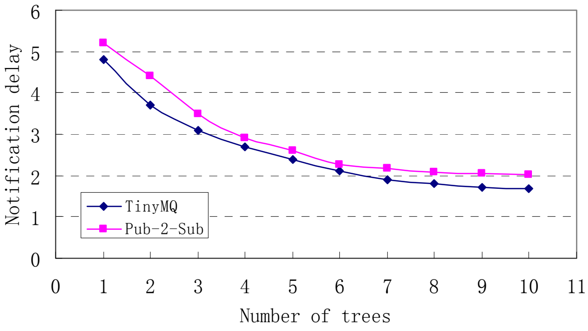


Figure 5. Notification delay

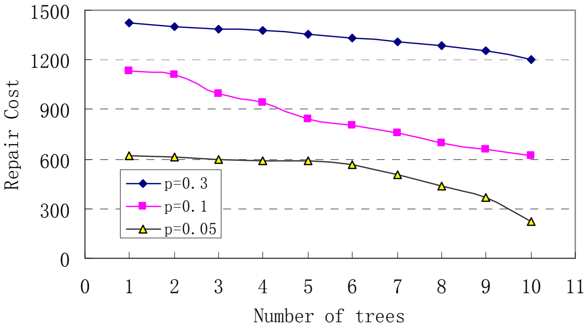


Figure 6. Repair cost

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

destination directly. Fig. 5 illustrates notification delay of TinyMQ is lower than PUB-2-SUB$^+$. The x axis represents the number of trees, and the y axis represents the value of $d_t/d_{optimal}$. Since both TinyMQ and PUB-2-SUB$^+$ do not create new network links, $d_{optimal}$ value should be same. The main reason for this improvement in notification delay is that the numbers of hops per query is lower in TinyMQ.

To evaluate the effect of failure, we let a random fraction *p* of the network down and investigate the impact in terms of repair cost which is defined that the total number of nodes that need to update its state to remain valid. Fig. 6 illustrates the repair cost versus different number of failing nodes. The x axis represents the number of trees, and the y axis represents the number of nodes that must update their status. Since TinyMQ needs to maintain the distance information, most nodes need to update its information. When the number of trees increase, the network partitioned into more small clusters, the number of affected nodes decreases due to this separation.

V. Conclusions AND FUTURE WORKS

Without location information the dissemination of queries and events in publish/subscribe services in sensor networks usually relies on a gossiping protocol which ignores the message content during the dissemination. Our proposed middleware, TinyMQ, provides an efficient content-based publish/subscribe architecture in WSNs. TinyMQ is based on a partition of the network, and nodes in each tree are assigned unique key values in such a way, that is convenient for content-based routing. Our simulation results have demonstrated several promising properties of TinyMQ in terms of subscription efficiency and notification delay.

Future work includes implementing and deploying TinyMQ in real sensor nodes such as MicaZ or IRIS. The issues of optimizing multi-tree based logic topology according to the network environment and studying the impact of node mobility will be also addressed.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China under Grant No.50805058, the Fundamental Research Funds for the Central Universities (HUST 2010MS083), and International S&T Cooperation Program of Hubei Province under Grant No. 2010BFA008.

References

[1] T. He, et al., Vigilnet: an integrated sensor network system for energy-efficient surveillance, ACM Transaction on Sensor Networks, vol. 2, no. 1, pp. 1–38, 2006.

[2] L. Fiege, M. Cilia, G. Mühl, and A.P. Buchmann, Publish-subscribe grows up: Support for management, visibility control, and heterogeneity, IEEE Internet Computing, vol. 10, no. 1, pp. 48–55, 2006.

[3] J.-H. Hauer, V. Handziski, A. Köpke, A. Willig, and A. Wolisz, A component framework for content-based publish/subscribe in sensor networks, in: Proc. of the 5th European Workshop on Wireless Sensor Networks, pp. 369-385, 2008.

[4] M. Rudafshani and S. Datta, Localization in wireless sensor networks, in: Proc. of the 6th International Conference on Information Processing in Sensor Networks, pp. 51–60, 2007.

[5] P. Costa, G.P. Picco, and S. Rossetto, Publish-Subscribe on Sensor networks: a semi-probabilistic approach, in: Proc. of the 2nd IEEE International conference on Mobile Ad-hoc and Sensor Systmes, pp. 322-332, 2005.

[6] E. Souto, et al., Mires: a publish/subscribe middleware for sensor network, in: Proc. of the 2nd IEEE International conference on Pervasive and Ubiquitous Computing, pp. 37-44, 2006

[7] P. Boonma and J. Suzuki, Toward Interoperable Publish/Susbscribe Communication Between Wireless Sensor Networks and Access Networks, in: Proc. of International Workshop on Information retrieval in Sensor networks, pp. 1-6, 2009.

[8] D.A. Tran and C. Pham, PUB-2-SUB: A Publish/Subscribe Content-based Framework for Cooperative P2P Networks, in: Proc. of the 8th IFIP Networking Conference, pp. 770-781, 2009.

[9] D.A. Tran and C. Pham, A Content-Guided Publish/Subscribe Mechanism for Sensor Networks without Location Information, Journal on Computer Communications, vol. 33, no. 13, pp. 1515-1523, 2010.

[10] R. Baldoni, C. Marchetti, A. Virgillito, and R. Vitenberg, Content-Based Publish-Subscribe over Structured Overlay Networks, in: Proc. of the 25th IEEE International Conference on Distributed Computing Systems, pp. 1-10, 2005.

[11] W. Acosta and S. Chandra, On the need for query-centric unstructured peer-to-peer overlays, in: Proc. of the 5th IEEE International Workshop on Hot Topics in Peer-to-Peer Systems, pp. 1-8, 2008.

iOLSR: OLSR for WSNs Using Dynamically Adaptive Intervals

Erlend Larsen, Joakim Flathagen, Vinh Pham
FFI, Norway
Email: {erl, jfi, vph}@ffi.no

Lars Landmark
Q2S, NTNU, Norway
Email: larslan@q2s.ntnu.no

***Abstract*—Proactive link state routing protocols, as used within the Mobile Ad hoc NETwork framework, have not been as successful in wireless sensor networks. This is mainly due to the extensive energy usage by control traffic transmissions and state requirements. However, such protocols are in many situations a more suitable candidate than their counterparts. The benefits are their topology overview, and more importantly the already available spanning trees for information distribution. The high signaling overhead associated with proactive protocols can be reduced by taking advantage of the static nature of wireless sensor networks. In this paper, we investigate how the Optimized Link-State Routing (OLSR) protocol, as a proactive routing protocol candidate, can be adapted to work better in a wireless sensor network environment. The basis for the solution is that control messages are sent with a low frequency when the network is stable, and more often if topology changes occur. The proposed solution is investigated using simulations from no loss to lossy link environments showing promising results.**

***Keywords*-Ad hoc networks; Routing; Wireless sensor networks.**

I. Introduction

Mobile Ad Hoc Networks (MANETs) and Wireless Sensor Networks (WSNs) are regarded as two distinctly different types of ad hoc networks, requiring routing protocols with specialized attributes. While MANET routing protocols are challenged with mobility, the main limitations for routing in WSNs are energy and memory. The IETF 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) working group [1] have made great efforts to bring IP to WSNs and other low power wireless networks. Therefore, modifying a MANET protocol that natively supports IP to better fit the challenges of WSNs may help the introduction of IP in WSNs.

Radio communication is a major energy consumer in WSNs. Shwe et al. [2] proposed among other measures to minimize the number of control packet transmissions to reduce energy usage. However, limiting the control traffic could make the routing protocol less able to maintain routes and do route repair. Thus, our goal is to reduce the number of control packet transmissions while avoiding these negative consequences.

The traffic pattern types in WSNs are mainly the following, arranged in probability of occurrence from high to low:

1) Sensors to sink
2) Sink or a specific controller to all or some sensors
3) Sensor to sensor

In the first and third case, the main challenges that have been addressed are on optimizing energy preservation and memory usage. Less attention has been given to traffic flows from a controller to the sensors, for instance for software updates. The process of software updating would require a more optimized distribution tree. Hence, a proactive protocol enabling optimized message distribution and routing would in many cases be more advantages than a protocol optimized for sensor to sink.

The Optimized Link State Routing (OLSR) protocol [3] is a proactive link state MANET routing protocol. It sets up and maintains routes regardless of application layer communication demands. The route maintenance is based on the regular transmission of control traffic. A high number of control packet transmissions will make the protocol less suitable for WSNs. At the same time it offers several advantages that are not as easily available with reactive protocols. For example, it can provide: quick rerouting in case of topology changes, spanning trees for information distribution, node cooperation, and node localization. Sleep functionality may be more challenging with a proactive protocol, but on the other hand, a proactive protocol may offer a distribution tree that may allow more efficient synchronization of sleep state. WSNs often operates over lossy links. In networks where links experiencing radio silence, a protocol enabling fast recovery is a requirement. Reactive protocols tend to increase their path distance as their have no mechanism to roll back after expiring radio silence as proactive protocols.

The OLSR protocol is not specifically designed for fixed topologies. Furthermore, due to its link state properties it also has a larger state requirement, than other protocols tailored for WSNs. There are no specified mechanisms to adapt the emission interval of control messages, depending on the grade of topology change. This means that the rate of control messages must be decided before the network deployment, based on the expected dynamics and the wanted reaction time to such dynamics. WSNs can be perceived as static and fixed without any dynamics. Nonetheless dynamics will occur, due to fluctuating links, new deployed nodes or nodes disappearing due to energy depletion or other malfunction. In a more dynamic network, where links or nodes break frequently, the routing protocol needs to perform control traffic dissemination more often.

The main contribution of this paper is the adaption of OLSR to exploit the static nature of WSNs through dynamically increasing the intervals of the control messages. This proposed solution is named iOLSR. Scaling OLSR used in WSNs has been criticized for the extensive use of state, which makes scaling a challenge. However, for medium WSNs, the required memory for holding OLSR state will likely be below current memory limitations of many commercial nodes for WSNs. The proposed solution shows its ability to reduce its control traffic, and to deal well with in environments with error prone links.

The rest of the paper is structured as follows. Related work is presented in Section II. The changes proposed to OLSR are presented in Section III. The solution is investigated and compared to alternatives in Section IV. Finally, the paper is concluded along with an outline of further work in Section V.

II. Related Work

Earlier work that address adaptation of OLSR for WSNs include [4], where Benslimane et al. propose a way to perform energy-aware routing using OLSR. Minet and Mahfoudh presents an energy-aware version of the OLSR routing protocol in [5]. These two papers focus on routing traffic over paths that minimize the energy consumed in the end-to-end transmission of a packet flow and avoiding nodes with low residual energy, increasing the network lifetime. iOLSR, on the other hand, does not consider energy levels, but instead focuses on the reduction of control traffic.

The OLSR standard allows for different nodes having different interval settings, but there are no described options or methods to vary the intervals while operating. Fast-OLSR [6] is a proposal to enable the broadcast of Fast-Hello messages with a shorter interval in case high mobility is detected. It is thus a proposal to change the control message intervals with basis in information about the relative mobility of the node, depending on if there is a high number of changes in the node's neighborhood. In addition, Fast-OLSR proposes a Fast-Hello message with a reduced set of neighbors announced, to reduce the increased routing traffic overhead. Our proposed solution does not impose a new network message type and it is tested in a link burst environment. As with Fast-OLSR, iOLSR is compatible with the standard OLSR protocol.

An IPv6 routing protocol for Low Power and Lossy Networks (RPL) is currently being developed in the IETF. Clausen and Herberg investigate RPL-Enabled Optimized Broadcast in [7]. The authors argue that Multi-Point Relay (MPR)-based efficient broadcast is a well performing mechanism for WSNs, and the MPR mechanism is essential to the OLSR routing protocol, upon which we base our proposed solution.

III. Proposed solution

The OLSR protocol uses two different control messages for its most basic routing functionality, *Hello* and *Topology Control (TC)*. The Hello messages are generated by all nodes and are periodically broadcasted to all 1-hop neighbors. Based on the information exchanged in Hello messages, a subset of the nodes in the network are selected as MPR nodes. These nodes generate Topology Control (TC) messages, which are flooded throughout the network using the other MPR nodes.

The Hello and TC emission intervals affect the reaction latency to topology changes, and the intervals can be set balancing between the energy usage and the topology change discovery latency. The default lengths of the Hello and TC intervals are 2 and 5 s, respectively, and the main motivation for these low values is the ability to cope with high mobility induced topology change.

There are two clear side effects of increasing the control packet intervals. The first is the increased latency in detecting link breaks. To reduce the link break detection time, disappearing nodes causing link failures can be detected using Link Layer Notification (LLN). The second side effect is the latency in detecting new nodes. If new nodes are introduced in the network when the network has been operating for a while, these nodes will only be employed for routing when the network has discovered them. However, until the new node has received TC messages from all elected MPRs in the network, there is a risk that the node will discard packets to destinations it is unaware of. And worse, generate loops if it has a different view of the shortest path than the upstream node. Large message intervals will delay the discovery and the use of more optimal paths.

In static WSNs, there are no topology changes caused by mobility. The topology is stable and static for the most of the time. After performing the initial discovery of the topology, the routing protocol could stop disseminating control messages. However, at any time, a node may disappear or make its appearance in the network, and links may fluctuate. Links may even be broken due to external causes, such as targets entering the the network detection zone. Therefore, even proactive protocols for WSNs must perform control with the network links to detect and recover from topology changes. To reduce the overhead of routing messages, which drains the nodes of energy, the message intervals can be increased or turned off. However, this is at the cost of slower detection of topology changes detected by necessary control packets.

We propose to allow each node to adjust its Hello and TC intervals depending on the local state of the network. In the initial startup phase, where each new received Hello message contains new information, the node keeps the default low interval between each new originated message. As the initialization phase draws to an end, and no more changes are experienced in the neighborhood, the control message intervals are increased. In this way, the energy usage is reduced while the topology is stable and unchanged. If a change is detected in the local neighborhood, the message intervals are reset, and then incremented anew when no changes are detected (i.e., the network is perceived as stable again). The topology changes are detected by using Link Layer Notifications (LLNs) and through Hello messages containing new information. As a consequence, the protocol is able to adjust itself to operate over both stable links and more lossy environments while optimizing the overhead of routing messages through the increasing intervals.

In our proposal, the intervals of the control messages must vary between a lower and an upper limit. If the lower interval is too short, the number of control messages transmissions will drain the network nodes of energy without improving the routing protocol's reaction time. If set very low, collisions of control messages may even impair the network, causing loss of data traffic due to loops or lack of routes. In the upper end, the time fields of the OLSR message header limits the maximum interval. The time is encoded in the header in a mantissa and exponent format, each of 4 bits, into one byte. A time value i is encoded as $i = C \cdot (1 + \frac{a}{16}) \cdot 2^b$ where a is the highest 4 bits of the field, and b is the lowest 4. The scaling factor C is proposed as $\frac{1}{16}$ second, giving a time field range of 0.0625 s – 3968 s. The scaling factor could be increased to achieve a higher maximum time range, which could be advantageous for our proposed solution, but this has not been looked into in this work. In such a case, one would loose the resolution at lower numbers.

The intervals and the corresponding message timeouts (valid times) are increased each time the control message is transmitted. Upon experiencing a change in the neighborhood the intervals are reset to the default values, and the incrementation process begins over again. The rate that the intervals are increased by, can be discussed, and the simulation results show three different takes on this increase.

A generic representation of the calculation of control message intervals and their timeouts has been sought for. The following formula represents the calculation of increasing the intervals continually, either linearly or exponentially. The generic interval v can be calculated as follows:

$$v = v_d \cdot (\alpha^i + \beta \cdot i) \qquad (1)$$

In (1), v is the resulting interval, v_d is the starting (default) interval, α is the base exponential value, β is the linear increment and i is a counter of successfully transmitted control messages. Upon a change in the local topology information, this counter is reset to 0.

The basis for message information timeout has been to follow the proposal of the OLSR RFC [3], using 3 times the interval rate as the timeout value. However, with an expected increasing interval, the timeout vt must be calculated as stated in (2).

$$vt = \sum_{k=0}^{2} v_d \cdot (\alpha^{(i+k)} + \beta \cdot (i + k)) \qquad (2)$$

In the equations, α and β are constants, set for the entire duration of the simulation use or network deployment.

IV. Simulations

A. Setup

The proposed solution was investigated using simulations on the ns-2 network simulator [8] version 2.34. The OLSR protocol as described in [3] and implemented for ns-2 in [9] was used for unicast routing, and the IEEE 802.11 protocol [10] was used as MAC layer. LLN was enabled in all simulations.

TABLE I
Simulation parameter settings

Radio-propagation model	TwoRayGround
Interface queue type	FIFO with DropTail, PriQueue for OLSR packets
Interface queue size	30 packets
Antenna Model	OmniAntenna
Data/basic rate	2 Mbps / 1 Mbps
Data transmission/sensing radius	250 m / 550 m
Simulation/measurement time	6000 s / 600–5900 s
Random seed	Heuristic

The solution was tested on a scenario with fluctuating links where the link loss is a consequence of link failures using an implementation of the Gilbert-Elliot link burst error model [11], [12]. The link has a certain probability of going into a 0–3 s burst error period for each received packet, and while in the burst error period, the link experiences a 100% packet loss. When not in a burst error period, the two-ray-ground radio propagation model is employed with a 250 m transmission radius.

The topology size was set to 40 nodes in a 1500 x 300 m^2 area, and the nodes were placed randomly using software from [13], to allow the examination of a wide network without the very long simulation processing time that follows using ns-2 with a high number of simulated nodes. The sink was randomly positioned. The simulation time was 6000 seconds unless otherwise stated. All nodes generated packets, except those nodes that appeared or disappeared, and the packets were set with the sink as destination, to test the paths toward the sink. The traffic load was 1 packet per second from each traffic-generating node. The traffic type was UDP unicast with a packet size of 50 bytes, and the traffic flows were started at 500 s. All data points are an average of 10 simulation runs, and are presented with a 95% confidence interval. The topologies were the same 10 topologies for each of the simulations. Other simulation parameter settings are presented in Table I.

For our proposed iOLSR solution, the interval incrementation counters were reset at the following events:

- Hello messaging causing link change or timeout.
- A new MPR selector or a timeout of an existing one.
- Link break causing a LLN.

When the static Hello intervals were increased, the TC intervals were increased correspondingly, so that for example a Hello interval of 10 had a TC interval of 25.

B. Results

1) Increasing intervals: First we investigated how the iOLSR solution compared to the regular static interval behavior of OLSR in varying link stability conditions. Three different static Hello intervals were simulated: 2, 20 and 100 s. The corresponding TC intervals are: 5, 50 and 250 s. The iOLSR default intervals were 2 s for Hello and 5 s for TC, and the interval increment was 2-base exponential. Examining the goodput results (Fig. 1) we see that all the variations manage to perform well when the topology is stable without many link errors. When the link error probability

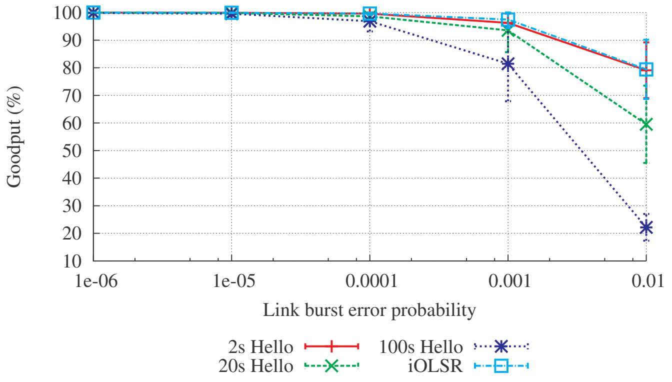


Fig. 1. Average goodput for iOLSR compared to OLSR with static intervals.

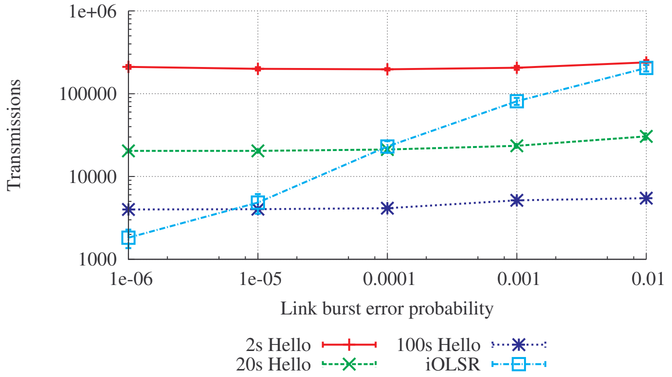


Fig. 2. Number of control packet transmissions for iOLSR compared to OLSR with static intervals.

increases, a higher interval between the Hello messages makes the routing protocol less capable of taking advantage of the links rebounding from a burst error, and this leads to a logical partitioning that reduces the goodput. Interestingly, we mark that iOLSR is able to offer the same performance as that of standard OLSR with 2 s Hello intervals, even at the highest link burst error probability.

While the goodput performance for all alternatives was very good at the lower link break probabilities, it is the number of transmitted control packets that is most interesting in WSNs, since the number of transmissions directly affect the energy use of the nodes. The number of control packet transmissions (Fig. 2) show iOLSR as being highly adaptive to the environment it operates in. When there is low probability of burst errors, the number of routing packets is kept at a much lower level than the comparable 2 s Hello interval results, and even compared to the 100 s Hello results. As the burst error probability increases, the routing protocol dynamically increases the number of Hello and TC messages generated locally in the area around each failing link.

2) Interval increment rate: The control message intervals' rate of increase is important when evaluating iOLSR. We have investigated three increment options where α and β refer to (1) and (2):

1) Linear (lin) ($\alpha = 1$ and $\beta = 1$)

TABLE II
CONTROL MESSAGE INTERVAL PROGRESSION

Hello			TC			
Lin	Exp2	Exp3	Lin	Exp2	Exp3	Exp3 vt
2	2	2	5	5	5	65
4	4	6	10	10	15	195
6	8	18	15	20	45	585
8	16	54	30	40	135	1755
10	32	162	35	80	405	5265
...	...	...	...	...	...	...

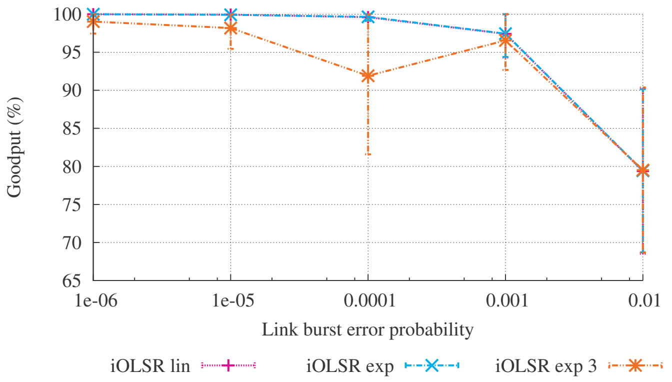


Fig. 3. Average goodput for linear contra exponential increase of iOLSR intervals.

2) 2-base exponential (exp2) ($\alpha = 2$ and $\beta = 0$)
3) 3-base exponential (exp3) ($\alpha = 3$ and $\beta = 0$)

As we see in Table II, the linear option will increase the interval by the default value for each successful transmission, while the 2-base and 3-base exponential options increment the message intervals exponentially according to (1).

The simulation results with varying control message interval incrementation rate shows that the goodput (Fig. 3) is not affected adversely by choosing a 2-base exponential increase of the intervals, even in an environment with a high probability of link burst errors. It follows the linear increment results very closely. The 3-base exponential increase is more prone to errors.

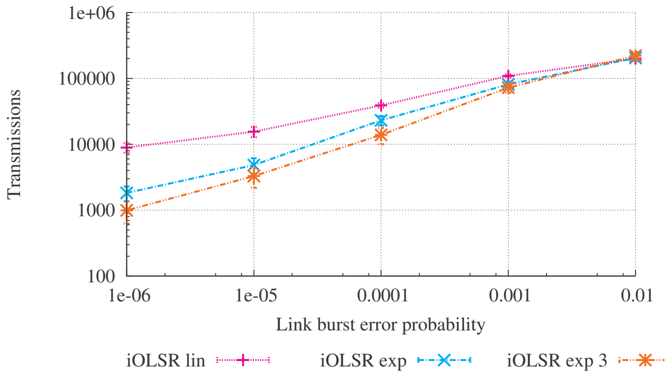


Fig. 4. Number of control packet transmissions for linear contra exponential increase of iOLSR intervals.

ISBN: 978-1-61208-144-1

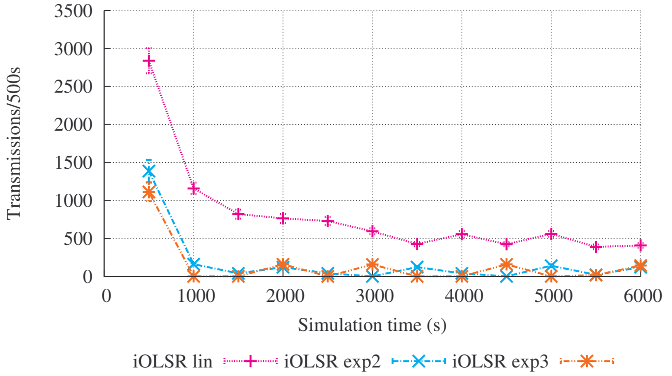


Fig. 5. Number of control packet transmissions for linear contra exponential increase of iOLSR intervals at increasing simulation time, 0–6000 s.

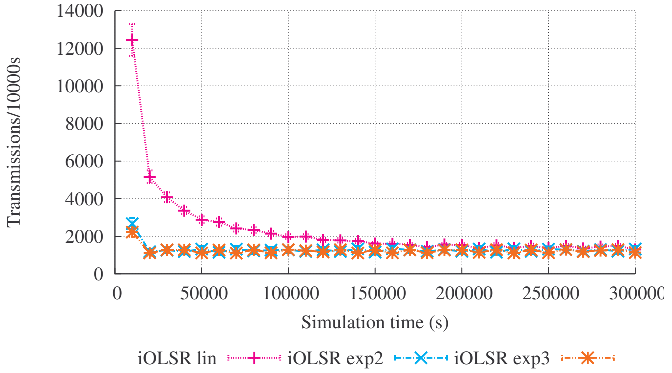


Fig. 6. Number of control packet transmissions for linear contra exponential increase of iOLSR intervals at increasing simulation time, 0–300000 s.

Examining the results for the number of control packet transmissions (Fig. 4), there is evidently great gain in using an exponential interval increment compared to a linear increment. However, the gain is much less in using a larger base exponent such as 3 compared to 2. The reason is twofold. First, the number of transmissions required to reach the maximum time field limit is lower with an increasing increment. After the incrementation phase, there is no difference between the increment values, since the intervals are no longer increased. For the exp3, the maximum interval for TC is reached at the fifth transmission, since the vt (control message information timeout) will be 5265 (Table II), thus exceeding the maximum time (3968). Second, the beginning of the incrementation phase is the phase where changes are most likely happen, especially at the initialization of the network. An interval increment that moves too quickly towards higher control message intervals may actually harm the initialization and convergence of the protocol.

Since there is great impact of how long the initialization phase is, when it comes to the number of transmissions, we have run some simulations without data traffic, and only routing traffic present, where the number of control packet transmissions is examined for the lin, exp2 and exp3 increment options. For a simulation lasting for 6000 s (Fig. 5), clearly the lin option is unable to reach the maximum interval. The two other options, however, reach the maximum interval very early.

In a longer simulation, 0–300000 s, the maximum time limit is even more pronounced, examining the control packet transmissions (Fig. 6). Except for the first measured step, the exponential increase options operate at the maximum for the entirety of the simulation.

The conclusion of the investigation into the increment alternatives is thus that the 2-base exponential increment represents a middleway between a too slow move away from the default intervals toward the maximum, and at the same time a more slow move away from the default intervals in the first phase of a stabilizing network.

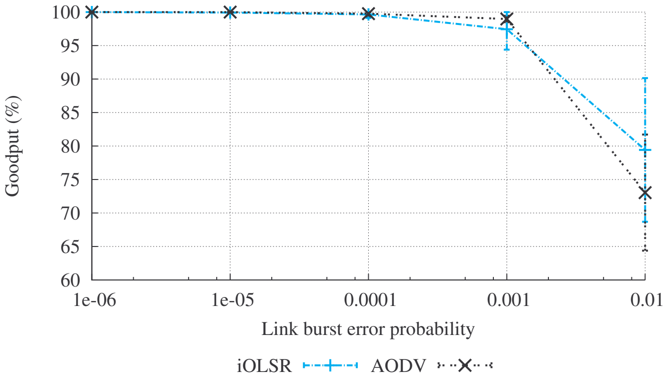


Fig. 7. Average goodput for iOLSR contra AODV.

3) Comparison with AODV: The final comparison in this paper is between the iOLSR (2-base exponential increase) and Ad hoc On-demand Distance Vector (AODV) [14]. Protocols for WSNs normally establish routes from sink to sensors to reduce the control traffic. AODV establishes routes from sensors to sink. In lossy environments it is likely that sensors must take part in path or link recovery. Hence, AODV in this sense resembles many protocols for WSNs in its behavior. In this work we want to compare the impact of burst error on iOLSR to a reactive protocol handling path/link recovery, such as AODV. With AODV, the start of the traffic flows were spaced up with 1 s intervals, to prevent effects of a synchronized route setup process. The goodput results (Fig. 7) show the interesting fact that AODV and iOLSR follow each other closely.

Examining only the number of control packets transmitted (Fig. 8) with the goodput results in mind, AODV is clearly better at low link burst error rates, yielding a much lower number of control packet transmissions than iOLSR.

However, the control packet results only tell part of the story. Investigating further, the number of hops for the data traffic (Fig. 9) is much higher for AODV than for iOLSR. This is due to the way AODV sets up routes only once, flooding the network with a route request from each source in the network. Although the total number of control packets may be low, the

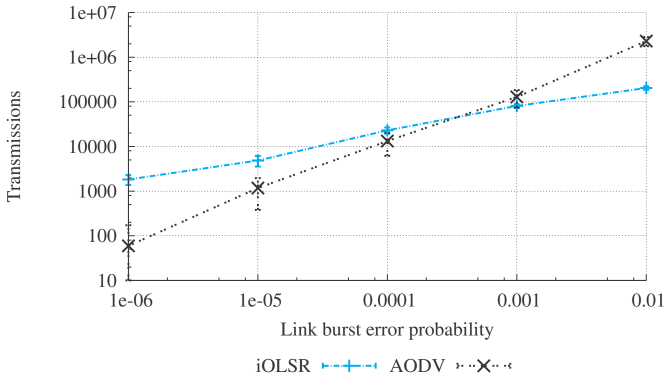


Fig. 8. Number of control packet transmissions for iOLSR contra AODV.

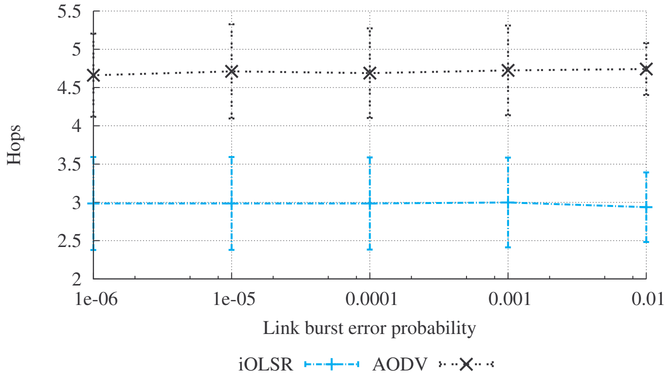


Fig. 9. Average number of hops for the data traffic for iOLSR contra AODV.

forwarding transmissions of the route requests are at risk of collisions, resulting in a failure to propagate the shortest path outwards to the destination. Furthermore, in case of low data traffic and low error rate, the result indicates the benefit of a reactive protocol. However, as traffic increases more traffic is traveling over more hops, thus draining more resources.

The consequence of the higher number of hops for the data traffic using AODV is the higher total number of transmissions (Fig. 10) where both the data traffic and control traffic transmissions are counted.

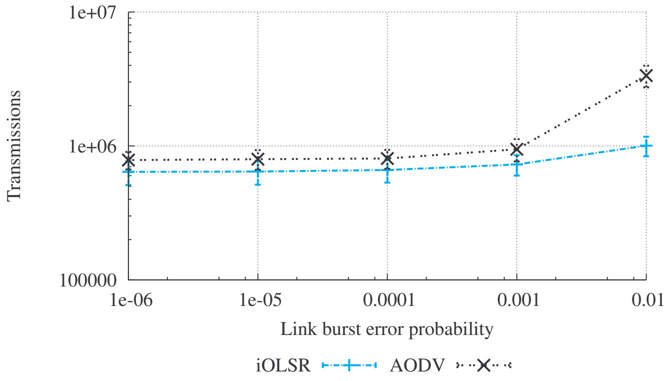


Fig. 10. Total number of transmissions (control and data traffic) for iOLSR contra AODV.

V. Conclusions and Future Work

This paper has presented an adaptation of OLSR for WSNs by introducing dynamically adaptive intervals. The advantages of employing dynamic intervals for control packets were demonstrated. We achieved less control packet overhead than by using the default control packet interval. Also, we demonstrated a faster detection and integration of new nodes than by using a large control packet interval.

The solution induces costs in terms of less route maintenance. Even so, the proposed solution represents a much better alternative for reducing the number of transmissions than that of preset large intervals, since it will depend on the real dynamics of the network whether the routing protocol transmits many or few packets. Last, but not least, using a proactive protocol provide the ability for a more optimized traffic pattern from sink to sensors.

Next, we will elaborate on the benefit of turning off the TC functionality to further reduce the control traffic. Destinations located further than two hops away would be searched for by a request using MPR for request forwarding.

References

[1] IPv6 over Low power WPAN working group. (2010, May) IETF. (Last accessed 2011-05-18). [Online]. Available: http://datatracker.ietf.org/wg/6lowpan/charter/

[2] H. Y. Shwe, X.-H. Jiang, and S. Horiguchi, “Energy saving in wireless sensor networks,” *Journal of Communication and Computer*, vol. 6, no. 5, pp. 20–28, May 2009.

[3] T. Clausen, P. Jacquet (editors), C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L.Viennot, “Rfc 3626: Optimized link state routing protocol (OLSR),” RFC 3626, pages 1–75, pp. 1–75, October 2003, network Working Group.

[4] A. Benslimane, R. El Khoury, R. El Azouzi, and S. Pierre, “Energy power-aware routing in OLSR protocol,” in *Mobile Computing and Wireless Communication International Conference, 2006. MCWC 2006. Proceedings of the First*, 17-20 2006, pp. 14 –19.

[5] S. Mahfoudh and P. Minet, “EOLSR: an energy efficient routing protocol in wireless ad hoc and sensor networks,” *Journal of Interconnection Networks*, vol. 9, no. 4, pp. 389–408, 2008.

[6] M. Benzaid, P. Minet, and K. Al Agha, “Analysis and simulation of Fast-OLSR,” *The 57th IEEE Semiannual Vehicular Technology Conference (VTC)*, vol. 3, pp. 1788–1792, April 2003.

[7] T. Clausen and U. Herberg, “Comparative study of rpl-enabled optimized broadcast in wireless sensor networks,” INRIA, Research Report RR-7296, May 2010, (Last accessed 2011-05-18). [Online]. Available: hal.inria.fr/inria-00488030/PDF/RR-7296.pdf

[8] J. Heidemann and T. Henderson (Editors). (2009, October) Network Simulator 2. (Last accessed 2011-05-18). [Online]. Available: http://nsnam.isi.edu/nsnam/

[9] F. J. Ros and P. M. Ruiz. MANET Simulation and Implementation at the University of Murcia (MASIMUM). (Last accessed 2011-05-18). [Online]. Available: http://masimum.dif.um.es/

[10] IEEE, “Wireless LAN medium access control (MAC) and physical layer (PHY) specification,” IEEE standard 802.11, June 1999.

[11] E. N. Gilbert, “Capacity of a burst-noise channel,” *Bell System Technical Journal*, vol. 39, pp. 1253–1265, 1960.

[12] E. O. Elliot, “Estimates of error rates for codes on burst-noise channels,” *Bell System Technical Journal*, vol. 42, pp. 1977–1997, 1963.

[13] S. PalChaudhuri. Ns-2 code for random trip mobility model. (Last accessed 2011-05-18). [Online]. Available: http://monarch.cs.rice.edu/~santa/research/mobility

[14] C. Perkins and E. Belding-Royer, “Ad-hoc On-Demand Distance Vector Routing,” in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, February 1999, pp. 90–100, New Orleans, LA.

An Effective Mechanism for Handling Open Voids in Wireless Sensor Networks

Mohamed Aissani, Sofiane Bouznad, Abdelmalek Hariza, and Salah-Eddine Allia
Laboratory of Research in Artificial Intelligence, Polytechnic Military School (EMP)
P.O. Box 17, Bordj-El-Bahri 16111, Algiers, Algeria
{maissani, bouznad.sofiane}@gmail.com, {malik-abd, s.alia}@hotmail.com

***Abstract*—Open voids are often formed on the boundary of a deployed wireless sensor network (WSN). Geographical routing protocols must handle these voids where packets fall into local minima. To contribute on resolving this problem, we propose in this paper an effective mechanism for handling this kind of voids. It uses two simple and effective algorithms ensuring discovery and maintenance of the network boundary. Contrary to existing void-handling techniques, our proposal uses the information about this boundary and the destination node for better directing data packets in optimal paths. Thus, open voids are avoided with great efficiency. The proposed mechanism has good performances in terms of packet delivery ratio, average routing path length, boundary energy consumed per delivered packet and average residual deadline of all delivered packets.**

***Keywords*—Sensor networks; geographical routing; open voids; void-handling techniques.**

I. Introduction

The mission of a WSN is generally to supervise a phenomenon, to take measures regularly and to send alarms to a sink node. Many applications using WSN exist in different fields such as defense, safety, health, agriculture and smart houses. Due to several economic and deployment considerations, sensor nodes have small size with limited resources of storage and computation. They use batteries, thus energy conservation becomes a big challenge.

Since they communicate by radio with short range, the multi-hop routing becomes necessary so that captured information reaches the sink node. A simple approach would be to use the geographical routing, which guarantees a good scalability and a positive progression of forwarded packets towards the sink node. Each sensor node forwards the current data packet to its neighbor, which is nearest that itself to the sink node. The fact that no routing information is to maintain in a network, other than tables of neighbors, routing paths of data packets adapt to any topological change.

Nevertheless, the geographical routing has two problems. Firstly, it is not applicable when sensor nodes do not have the possibility of knowing their geographical locations. Virtual coordinates systems, such as NoGeo [1], GEM [2], and BVR [3], can be used in this case. These coordinates require nodes to know the distances from its neighbors to certain points of reference by using periodic messages. Secondly, there can be voids between a source node and a sink. A void is an area without any active node. It can be located inside the network (closed void) or on the network boundary (open void). A geographical routing path towards a sink is interrupting when relay nodes for avoiding voids are absent. Existing solutions present insufficiencies in handling open voids [4-17], so we propose in this paper an effective mechanism for this kind of voids.

The rest of this paper is organized as follows. Section II presents the problem of open voids. Section III describes two algorithms that we propose to discover and maintain open voids on a deployed WSN boundary. Section IV presents the proposed mechanism for handling open voids in WSNs. Section V evaluates performances of our void-avoidance mechanism. Section VI concludes the paper.

II. Open-Void Problem

A void is an area where sensor nodes are unable to route packets or straightforwardly inalienable. It appears when using a random deployment of nodes or because of node breakdown due to various reasons, such as circuit breakdown, destruction or energy exhaustion of some nodes. The problem of geographical routing is that stuck nodes, located on a void boundary, can receive packets destined to the sink. Let us consider the example in Figure 1, where black nodes are located on the void boundary and node i must forward a packet to the destination node d. In this case, node i is stuck because there is any forwarding neighbor closer to node d. Once received by node i, the packet cannot have a positive progression towards node d. This packet will be directed towards node j (or node k) in a negative progression around the void. The node where a packet may get stuck is called a local minimum.

Without an efficient void-handling mechanism, data packets are dropped, wasting the network resources and communications can be lost between a few pairs of nodes. Such a behavior is strongly undesirable in WSNs and the loss of some critical information can harm the network mission.

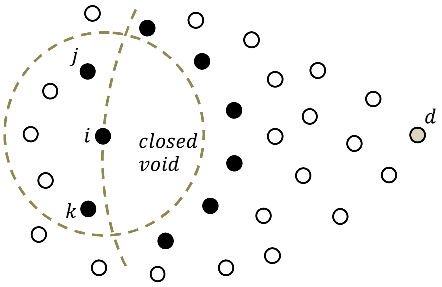
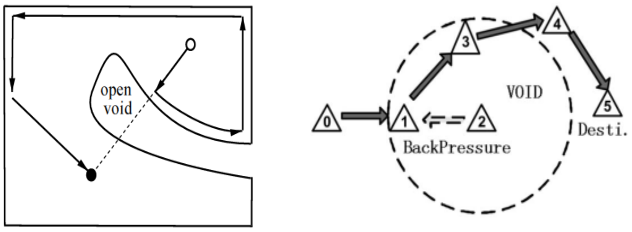


Figure 1. The void problem: i is a stuck node.

Open voids are located on the boundary of a deployed WSN. In order to reduce their negative impact on the routing effectiveness, particularly in case of real-time applications, several void-handling techniques exist in the literature. They gather in two classes (Figure 2): right-hand rule [4-13] and backpressure rule [14-17].



(a) Right-hand rule [4] (b) Backpressure rule [14]

Figure 2. Classes of techniques handling open voids.

The techniques belonging to the first class use boundary nodes to route any stuck packet towards its destination. In [4], the geographical routing algorithm GPSR is proposed. On a non-stuck node, the packet is forwarded by GPSR to the nearest neighbor to the destination node (greedy forwarding mode). Consequently, the destination is approximate hop by hop until reached by the packet. When this mode fails, the current node uses the face routing to overcome the meted void (perimeter forwarding mode). Boundary nodes apply the right-hand rule until the packet arrives at a node closer to the destination. Several other algorithms using the face routing were proposed later [5-9]. However, [18] showed that planarisation algorithms used to obtain a planar graph, such as Gabriel graph [4], reduced the number of usable links in a network. However, sensor networks deployed for real-time applications cannot admit this reduction because of its negative impact on exploring multiples paths towards the packet destination (load balancing and network fluidity).

On the other hand, the techniques belonging to the second class exploit the backpressure beacons broadcasted by the boundary nodes. When receiving these messages, upstream neighbors get alternative paths around the met void for next data packets. SPEED [14] is a spatiotemporal communication protocol proposed for WSN. It assures an end-to-end soft real-time for data parquets, requires each node to maintain information on its neighbors and employs the geographical forwarding to choose routing paths.

Moreover, SPEED maintains a desired delivery speed across sensor networks with a two-tier adaptation included for diverting traffic at the networking layer and locally regulating packets sent to the MAC layer [14]. It considers a routing void as a permanent congestion. In SPEED, a stuck node drops the received packet and sends out a backpressure beacon informing its neighbors about its final incapacity to forward the next packets. When its forwarding neighbors are stuck nodes, the current node drops the packet and broadcasts a backpressure beacon. This process is repeated until an alternative path is found or the source node reached by the beacons. To improve QoS guarantees, former works [15][16] proposed extensions to SPEED but they not changed the technique for handling routing voids.

The right-hand rule is less effective when handling open voids. It excessively uses boundary nodes and consumes rapidly their energy. In this case, several sessions can use a same boundary, where the problems of collisions and delays of packets. In the same way, the backpressure rule generates not only many control packets but also drops data packets in concave zones of voids. Routing paths are long because of backpressure beacons, from where links are overloaded and packets delayed. These packets will be dropped after their deadline expires, a non-desirable situation in case of real-time application.

To mitigate these insufficiencies, we propose an effective mechanism for handling open voids in WSNs. Called OVA-nb (*Oriented Void-Avoidance on network boundary*), the proposed mechanism orients each stuck packet on the network boundary towards its destination node. It uses the geographical coordinates of the current node, those of the network center and those of the packet destination node to compute the packet orientation around an open void. It is based on two simple and effective algorithms: NBD (*Network Boundary Discovery*) and NBM (*Network Boundary Maintenance*). The first algorithm identifies nodes forming the network boundary just after its deployment and the second one maintain this boundary in reactive manner. Unlike existing techniques using long routing paths (Figure 2), OVA-nb uses short paths to avoid open voids (Figure 3).

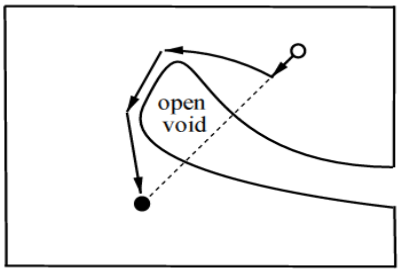


Figure 3. A short path used by our mechanism.

III. PROPOSED ALGORITHMS

Existing algorithms to discover and maintain voids, such as BOUNDHOLE [10] and the right-hand rule [11-13], inserts information about each boundary node in the VD (Void-boundary Discovery) packet, increasing the node memory requirements and reducing the algorithm scalability. Moreover, these algorithms periodically check an eventual failed node and rediscover the entire void if a boundary node fails. It would be interesting to rediscover only the affected local section of the void. The VD packet size grows whenever it moves forward on the boundary of a void to discover. Therefore, existing algorithms [10-13] deplete a significant portion of boundary nodes energy. The same drawback is true for the void maintenance procedure used by these algorithms. BOUNDHOLE [10] does not address the open void as a special case. The outside of the network deployment scope, including the open void shown in Figure 2-a, is considered as a great void. For each stuck packet on the network boundary, the algorithm uses a long routing path formed mainly by boundary nodes. At the same time, the right-hand rule does not consider an open void as a particular problem. It handles only the closed voids located inside a deployed sensor network.

To overcome these limits, we propose two simple and efficient algorithms. The NBD algorithm brings back all the nodes forming the boundary of a deployed WSN and then calculates and communicates its center. The NBM algorithm detects and then updates any topology change that can occur on the network boundary during its mission.

A. NBD algorithm

A designed sink (node c_i in the Figures 4 and 5) initiates the NBD algorithm when deploying a WSN. The algorithm operation is based on the GPSR protocol [4] to find the node closest to a virtual point located at one end of the network field. This node will complete the process of exploring the network boundary. The NBD algorithm takes place in three phases: initial phase, intermediate phase and final phase.

1) Initial phase: sink c_i selects the nearest border of a network field; i.e., the line which passes by one of the points B1, B2, B3 or B4 in Figure 4. Then node c_i projects its geographical location on the selected border. The resulting point (B1 in our example) represents the fictitious destination d_f used by the NBD algorithm to discover the nodes forming the network boundary.

2) Intermediate phase: the sink c_i sends to the fictitious node d_f a new packet ND (Network-boundary Discovery), whose header fields are summarized in TABLE I, to identify the fields Min and Max of the network boundary. The packet ND is routed by using greedy and perimeter modes of the GPSR protocol. When node b_i receives the packet ND, it launches the perimeter mode on the network boundary (Figure 5-a). During this process, the fields N1Up (1-hop upstream boundary node), N1Down (1-hop downstream boundary node) and N2Down (2-hops downstream boundary node) of each intermediate node are updated. When b_i (node that initiated the last perimeter mode) receives the packet ND for a second time, it deduces that it is the closer node to d_f. Thus, b_i execute the final phase of the NBD algorithm.

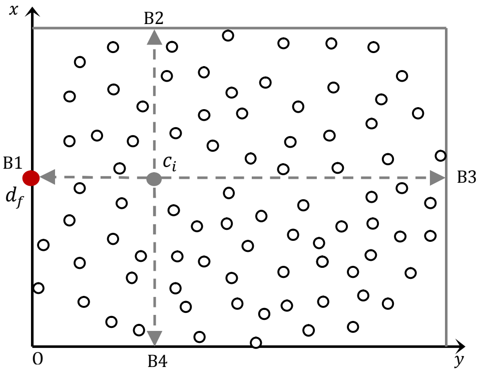


Figure 4. Fictitious destination d_f for the NBD algorithm.

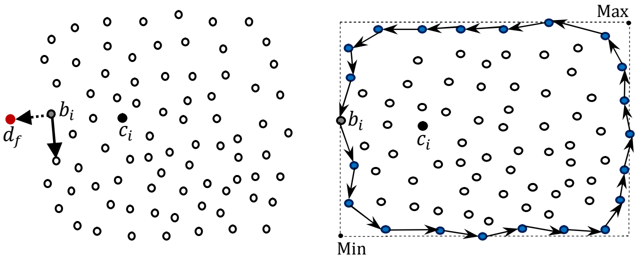
3) Final phase: when receiving the packet ND, node b_i computes the network center (the midpoint of $\overline{\text{Min Max}}$), drops the packet ND and sends a new packet NU (Network-boundary Update), marked by its identifier, to browse the network boundary in the opposite direction of the packet ND. The header fields of the packet NU are summarized in TABLE II. Each boundary node b_i that receives the packet NU updates its boundary information (NBorder=1 and NCenter=NU.NCenter) and verifies the field NodeUp of packet NU. If this field identifies a neighbor of b_i then node b_i updates its field N2Up by NodeUp, otherwise N2Up receives N1Up. Note that N1Up and N1Down are used to maintain the network boundary, N2Up (2-hops upstream boundary node) and N2Down to route packets using two hops on the network boundary. This routing technique reduces energy consumption and minimizes end-to-end delays of the routed packets.

TABLE I. THE HEADER FIELDS OF THE PACKET ND

Field	Mission/Content
PerimID	Identifier of the node having lance the last perimeter mode
DestID	Coordinates of the fitifious destination d_f
Mode	Forwarding mode of the packet ND: Greedy or Perimeter
Distance	Distance from d_f to the last node initiated a perimeter mode
Min	Coordinats of the minimum point on the network boudary
Max	Coordinats of the maximum point on the network boudary
NodeUp	Identifier of the boundary node having sent the packet ND

TABLE II. THE HEADER FIELDS OF THE PACKET NU

Field	Mission/Content
NCenter	Coordinates of the network center
NodeUp	Identifier of a node having sent the pauqet NU



(a) Fictitious destination d_f.

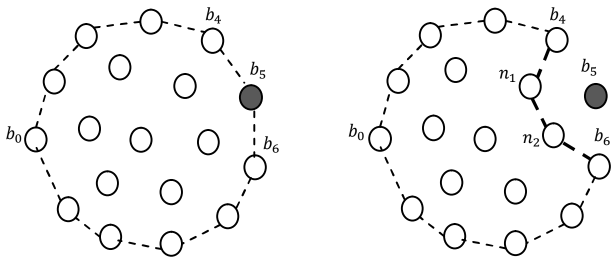
(b) Nodes forming the network boundary

Figure 5. Discovery process of the network boundary.

B. NBM algorithm

Some network-boundary nodes may stop working because of insufficient energy or hardware failure. The network boundary can also change shape following the redeployment of nodes on the outside the network. For information usable by any routing process, the algorithm NBM distinguishes two cases: (a) Failed node on the network boundary, (b) Redeployed node outside the network but near its boundary.

1) Failed node on the network boundary: through its field N1Up, each boundary node b_i can detect the absence of its direct upstream boundary node b_{i-1}. On expiry of the validity time of b_{i-1} in its neighbors table, node b_i discovers a new boundary segment to connect to the old one. Following the failure of boundary node b_5 in Figure 6-a, node b_6 discovers the new segment $b_6n_1n_2b_4$ that connects to the old segment $b_4b_0b_6$ of the network boundary (Figure 6-b). For this discovery, b_i considers b_{i-1} as fictitious destination, sets forwarding mode to perimeter in the packet ND and executes the intermediate phase of the algorithm NBD. The discovery of new nodes is completed in the first node encountered in the old boundary segment (node b_4 in Figure 6-b). This node is recognized by its field N1Up that is different from the default value. Once the two segments connected, the packet ND will continue its travel to restore the full information of the new network boundary. Upon receiving the packet ND, b_i (node b_6 in Figure 6-b) executes the final phase of the NBD algorithm updating fields of nodes on the network boundary.

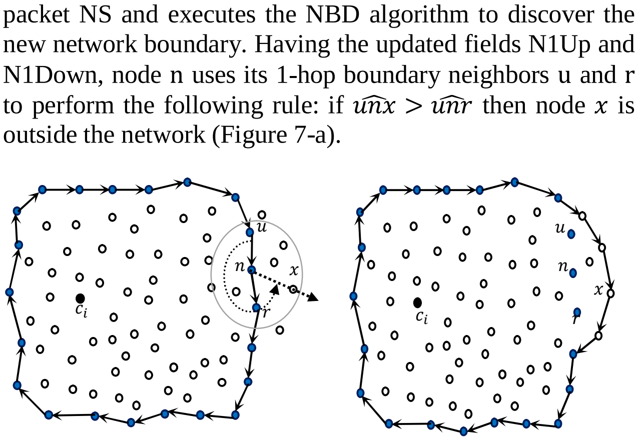


(a) Boundary node b_5 failed.

(b) Network boundary updated.

Figure 6. Network boundary updating after a node failure.

2) Redeployed node outside the network: upon receiving a location beacon from a neighbor x, boundary node n checks its neighbors table. If x is outside the network, node n sends a new packet NS (Network-boundary Suppression), marked by its identifier, on the actual network boundary. Its mission is removing the information concerning this boundary. When receiving the packet NS, each intermediate node b_i resets the fields concerning the network boundary (NBorder, N1Up, N2Up, N1Down and N2Down). At the end, node n drops the packet NS and executes the NBD algorithm to discover the new network boundary. Having the updated fields N1Up and N1Down, node n uses its 1-hop boundary neighbors u and r to perform the following rule: if $\widehat{unx} > \widehat{unr}$ then node x is outside the network (Figure 7-a).



(a) Node x deployed outside the network.

(b) Network boundary updated

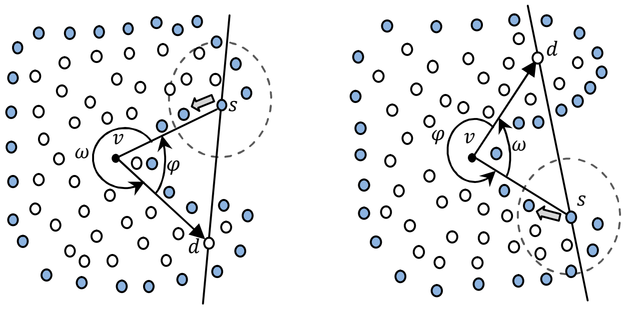
Figure 7. Network boundary updating after a node deployment.

IV. Proposed Mechanism

The proposed OVA-nb mechanism orients towards the sink all packets arriving on the network boundary. Its role is to prevent these packets from drops by nodes located on boundaries of open voids. Having the network center and the updated fields NCenter, N2Up and N2Down, boundary node s (s.NBorder=1) forwards any received packet p to its destination node d by using the angles $\varphi = \widehat{dvs}$ and $\omega = \widehat{svd}$, shown in Figure 8. When receiving p, node s performs the following rules:

- If φ < ω (Figure 8-a) then p is forwarded at the right of the line (sd). Thus, node s updates the orientation field in p if necessary, constructs its set R (greedy forwarding neighbors of s located at the right of line (sd)) and executes the following rule: if R is empty the next-hop node n of p is identified by the field s.N2Down, otherwise n is chosen from R.
- If φ ≥ ω (Figure 8-b) then p is to forward at the left of the line (sd). In this case, node s updates the orientation field of p if necessary, constructs its set L (greedy forwarding neighbors of s located at the left of line (sd)) and executes the following rule: if L is empty the next-hop node n of p is identified by the field s.N2Up, otherwise n is chosen from L.

Note that the next-hop node n is chosen from the set R (or L) according to the routing strategy of the implemented protocol that uses the mechanism OVA-nb. Associated with SPEED for performance evaluation, OVA-nb uses the neighbor delivery speed as a criterion to choose the next-hop of the packet p. Also, when s is not a network-boundary node (i.e., s.NBorder=0), it executes the routing strategy used by the implemented protocol for choosing the next-hop of each data packet p.



(a) Forwarding at the right of (sd)

(b) Forwarding at the left of (sd)

Figure 8. Packet orientation by the mechanism OVA-nb.

V. Performance Evaluation

Since we are interested by critical applications using WSNs, we first implemented the well-known real-time routing protocol SPEED by using the network simulator ns-2 [19]. For better performance, we associate with SPEED the mechanism OVA-nb, to handle open voids, and the resulting protocol is called SPEED-nb.

We compare SPEED-nb performance to those traditional protocols SPEED and GPSR. We use simulation scene with a grid distribution of nodes and the parameters summarized in TABLE III. Our objective is to show the inadequacy of existing techniques in handling open voids. This scene has a size of 800m×800m and contains 925 nodes. It contains an open void with 120m as radius, located on the right boundary of the scene. Six source nodes, selected randomly and located at the top of the void, periodically send data packets to a destination node located at the bottom of the void. Note that to enable a minimum of forwarded packets to the same destination node by the evaluated protocols, two other source nodes are selected from the left side of the void.

We evaluate performance of the protocols SPEED-nb, SPEED and GPSR at packet rate of 2 p/s. We vary the packet deadline between 50ms and 300ms. At the end of each simulation and for each protocol, we measure the packet delivery ratio, the average routing-path length, the average boundary-energy consumed and the average gain in deadline for each received packet. Each point in our graphs represents the average results of 15 simulations, with random source nodes for each simulation, performed under same conditions and during 221s.

TABLE III. SIMULATION PARAMETERS

MAC layer	IEEE 802.11
Radio model	RADIO-NONOISE
Propagation model	TwoRayGround
Antenna model	OmniAntenna
Queue model	Queue/DropTail/PriQueue
Size of the queue	50 packets
Canal de transmission	WirelessChannel
Wireless interface	WirelessPhy
Bandwidth	200 Kb/s
Size of data packets	32 bytes
Energy model	Energymodel of ns-2
Radio range	40 m
Transmission power	0.666 w
Reception power	0.395 w

Figure 9 shows that 75ms of packet deadline is sufficient for SPEED-nb to route successfully all the packets because it proposes to take a short path toward the destination node as shown in Figure 10. To reach the same performance, GPSR needs 300ms as packet deadline. This is because the face routing of GPSR which uses many boundary nodes before reaching the sink. Therefore, too long and busy routing paths are used by GPSR, as shown in Figure 10, and many packets are dropped because their deadline expires.

SPEED also removes many data packets, as shown in Figure 9, because backpressure beacons that generate stuck nodes delay the next packets in their progression and block definitely some source nodes. For this reason, packet delivery ratio of SPEED remains weak despite the growth of packet rate (Figure 9). For packet deadline exceeding 250ms, GPSR uses long routing paths to deliver the maximum number of packets, but the protocols SPEED and SPEED-nb use short paths for all delivered packets (Figure 10).

Figure 11 show that excessive use of boundary nodes in GPSR has led to large energy depletion in these nodes, but SPEED-nb consumed less energy of boundary nodes because it uses short routing paths. Consequently, it delivers data packets with significant residual deadline (i.e., reduced end-to-end delays), as shown in Figure 12.

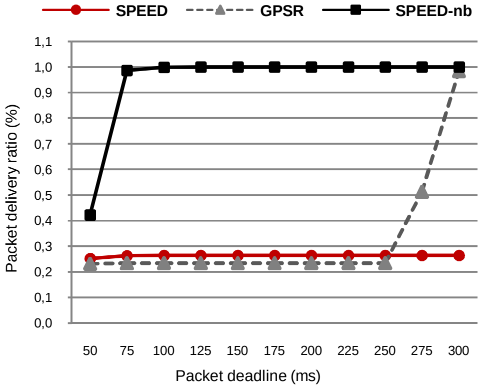


Figure 9. Success rate in delivering data packets.

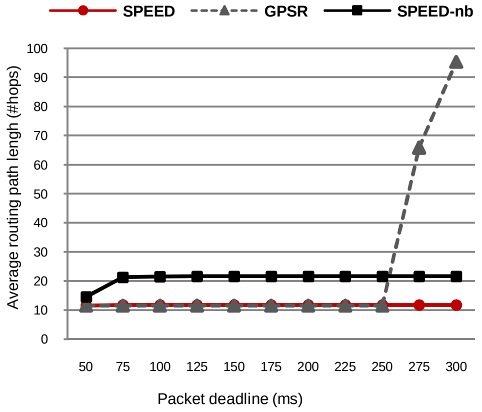


Figure 10. Average routing path length of delivered packets.

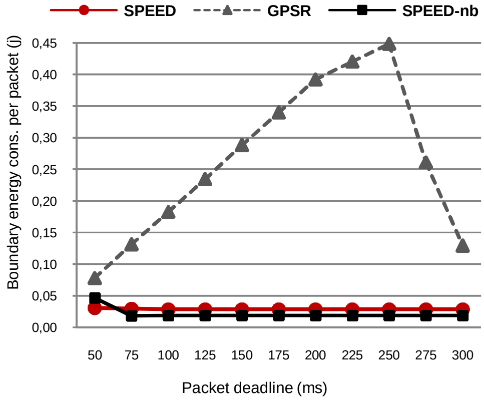


Figure 11. Boundary energy consumed per delivered packet.

VI. CONCLUSION

We proposed the mechanism OVA-nb whose role is to orient each stuck packet from the network boundary towards its destination node. We also proposed two simple and effective algorithms used by OVA-nb to discover and maintain all boundary nodes of a deployed sensor network; where open voids are frequently formed. To evaluate the OVA-nb performances, we associated it with the well-known protocol SPEED. Evaluated by simulation, obtained protocol SPEED-nb outperformed the traditional protocols SPEED and GPSR in terms of packet delivery ratio, average routing path length, boundary energy consumed for each delivered packet and average residual deadline of delivered packets. Our mechanism OVA-nb resolved the insufficiencies of existing techniques in handling open voids. It is simple to implement, effective in handling open voids and can be easily associated with any geographical routing protocol.

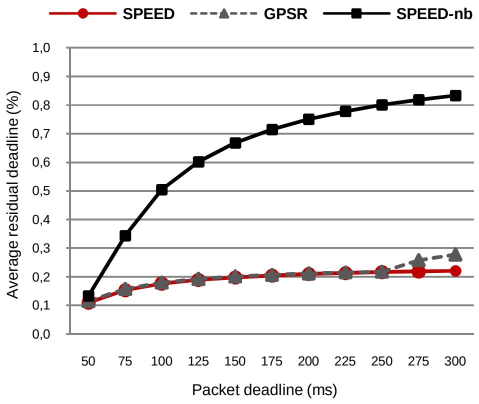


Figure 12. Average residual deadline of delivered packets.

Using the same approach, our current work is to propose a novel mechanism to deal with closed voids in WSNs, which will improve performances of the void-avoidance mechanism that we already proposed in [20][21]. We also plan to implement our proposals in a real scenario based on Imote2 sensor nodes.

REFERENCES

[1] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica, “Geographic Routing without Location Information,” Proc. of 9th Annual Int’l Conference on Mobile Computing and Networking, pp. 96-108, San Diego, CA, September 14-19, 2003.

[2] J. Newsome and D. Song, “GEM: Graph Embedding for Routing and Data-Centric Storage in Sensor Networks without Geographic Information,” Proc. of the 1st Conference on Embedded Networked Sensor Systems, pp. 76-88, Los Angeles, USA, Nov. 5-6, 2003.

[3] R. Fonseca, S. Ratnasamy, J. Zhao, C. Tien Ee, D. Culler, S. Shenker, and I. Stoica, “Beacon Vector Routing: Scalable Point-to-Point in Wireless Sensornets,” Proc. of the 2nd Symposium on Networked Systems Design and Implementation (NSDI), pp. 329-342, Boston, Massachusetts, USA, May 2-4, 2005.

[4] B. Karp and H. Kung, “GPSR: Greedy perimeter stateless routing for wireless networks,” Proc. of the ACM/IEEE Conference on Mobile Computing and Networking (MobiCom), pp. 243-254, Boston, Massachusetts, USA, August 6-11, 2000.

[5] B. Leong, S. Mitra, and B. Liskov, “Path Vector Face Routing: Geographic Routing with Local Face Information”, Proc. of the 13th IEEE International Conference on Network Protocols (INCP’05), Boston, MA, USA, November 6-9, 2005.

[6] L. Moraru, P. Leone, S. Nikoletseas, and J.D.P Rolim, “Near optimal geographic routing with obstacle avoidance in wireless sensor networks by fast-converging trust-based algorithms,” Proc. of the 3rd ACM Workshop on QoS and security for wireless and mobile networks, pp. 31–38, Chania, Crete Island, Greece, Oct. 22-26, 2007.

[7] L. Moraru, P. Leone, S. Nikoletseas, and J. Rolim, “Geographic Routing with Early Obstacles Detection and Avoidance in Dense Wireless Sensor Networks,” Lecture Notes in Computer Science (LNCS), Vol. 5198, pp.148-161, 2008.

[8] F. Kuhn, R. Wattenhofer, and A. Zollinger, “An Algorithmic Approach to Geographic Routing in Ad Hoc and Sensor Networks,” IEEE Transactions on Networking, Vol. 16(1), pp. 51–62, 2008.

[9] F. Huc, A. Jarry, P. Leone, L. Moraru, S. Nikoletseas, and J. Rolim, “Early Obstacle Detection and Avoidance for All to All Traffic Pattern in Wireless Sensor Networks,” Lecture Notes in Computer Science (LNCS), Vol. 5804, pp. 102–115, 2009.

[10] Q. Fang, J. Gao, and L.J. Guibas, “Locating and bypassing routing holes in sensor networks,” Journal of Mobile Networks and Applications, Vol. 11(2), pp. 187–200, Oct. 2006.

[11] F. Yu, E. Lee, Y. Choi, S. Park, D. Lee, Y. Tian, and S.H. Kim, “A modeling for hole problem in wireless sensor networks,” Proc. the ACM International Conference on Wireless Communications and Mobile Computing, pp. 370-375, Amman, Jordan, Sept. 6-8, 2007.

[12] F. Yu, Y. Choi, S. Park, E. Lee, Y. Tian, M. Jin, and S.-H. Kim, "Anchor node based virtual modeling of holes in wireless sensor networks," Proc. of the IEEE International Conference on Communications, pp. 3120-3124, Beijin, China, May 19-23, 2008.

[13] F. Yu, S. Park, E. Lee, and S.H. Kim, “Hole Modeling and Detour Scheme for Geographic Routing in Wireless Sensor Networks,” Journal of Communications and Networks, pp. 327-336, 2009.

[14] T. He, J.A. Stankovic, C. Lu, and T. Abdelzaher, “A Spatiotemporal Communication Protocol for Wireless Sensor Networks,” IEEE Transactions on Parallel and Distributed Systems, Vol. 16(10), pp. 995-1006, October 2005.

[15] E. Felemban, C.G. Lee, and E. Ekici, “MMSPEED: Multipath Multi-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks,” IEEE Transactions on Mobile Computing, Vol. 5(6), pp. 738-754, June 2006.

[16] W. Cheng, L. Yuan, Z. Yang, and X. Du, “A real-time Routing Protocol with Constrained Equivalent Delay in Sensor Networks,” Proc. of the 11th IEEE Symposium on Computers and Communications, pp. 597-602, Sardinia, Italy, June 26-29, 2006.

[17] W. Jia, T. Wang, G. Wang and M. Guo, “Hole Avoiding in Advance Routing in Wireless Sensor Networks,” in Proc. of the IEEE Wireless Communications and Networking Conference, pp. 3519-3523, Hong Kong, China, March 11-15, 2007.

[18] K. Seada, A. Helmy, and R. Govindan, “Modeling and analyzing the correctness of geographic face routing under realistic conditions,” Ad Hoc Networks, Vol 5(6), pp. 855–871, 2007.

[19] Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, “The ns Manual”, available at: *http://www.isi.edu/nsnam/ns/, January 12, 2011.*

[20] M. Aissani, A. Mellouk, N. Badache, and B. Saidani, “Oriented Void Avoidance Scheme for Real-Time Routing Protocols in Wireless Sensor Networks,” Proc. of the IEEE GLOBECOM Conference, pp. 83-87, New Orleans, LA, USA, 30 Nov. - 04 Dec. 2008.

[21] M. Aissani, A. Mellouk, N. Badache, and M. Boumaza, “A Novel Approach for Void Avoidance in Wireless Sensor Networks,” Int’l Journal of Communication Systems (IJCS), John Wiley & Sons Editions, Vol. 23(8), pp. 945–962, 2010.

A Framework for Sensor Stream Reduction in Wireless Sensor Networks

Andre L. L. Aquino
Computer Institute – Federal University of Alagoas
Campus A. C. Simoes, Av. Lourival Melo Mota, s/n, Tabuleiro do Martins
Maceio-AL, Brazil, zipcode: 57072-970
Email: alla@ic.ufal.br

***Abstract*—This work presents a general methodology to perform sensor stream reduction in wireless sensor networks. This methodology considers the application requirements, the reduction design, and the data reduce validation. Specifically, the reduction design, we present a architecture that can be applied to reduce the data when it is sensed or routed through to sink. The objective of this work is to show step-by-step how we can realize reduction applications in wireless sensor networks by using our methodology. The study cases show the usefulness of our methodology applied on a general sensing and a real time scenarios.**

Keywords-Sensor stream; Reduction; Wireless sensor networks.

I. Introduction

A wireless sensor network (WSNs) [1] is a special type of network represented by a set of sensor nodes, where each node works by detecting events, performing quick local data processing, and transmitting data through a ad-hoc wireless communication. Sensor nodes work in a cooperative way, and all their measurements are sent to a special node called sink. WSNs are commonly used in applications such as environmental, habitat, or industrial monitoring [2]. The phenomena monitored usually require measurements of physical variables, such as temperature, pressure, and humidity, and a single node can monitor one or more of these variables.

The data generated by WSNs have particular characteristics, as they arrive at the sink node in an online fashion, are unlimited, and there is no control in the messages order of arrival. This type of data is nowadays referred as *data stream* [3]. Besides the usually characteristics of any data stream, a *sensor* data stream has other peculiar features, since it represents only a sample of the entire population, is usually imprecise and noisy, and of moderate size.

There would be no problem in collecting, processing and transmitting a data stream if a WSN was not limited by a series of imposed restrictions. WSNs have a limited energy source (being a microelectronic device, a sensor node can be only equipped with a limited power source), low computational power, and reduced bandwidth, plus the weakness of a wireless medium communication. From these restrictions, the energy is the most critical, as the sensor lifetime strongly depends on the battery lifetime, and in some applications renewal of power resources might not be possible.

In these conditions, dealing with all the data stream becomes an unfeasible task. If the sensor node transmits all its measurements, it spends a lot of energy, and there is no guarantee that the data will not be delayed or lost. In order to respect the WSN restrictions, many strategies for data processing were proposed, including data aggregation, data fusion and data reduction.

The simplest strategy of the aforementioned is data aggregation [4]. Data aggregation reduces the data considering some application requirement, for example the data location. The main objective of this approach is to reduce the network data traffic by reducing the number of packets by aggregating them, regardless of data semantics. Data fusion [5], in contrast, is a more sophisticated strategy that focuses on processing data gathered by sensor nodes by benefiting from their processing capability. By exploiting the synergy among the available data, fusion techniques can reduce the amount of data traffic, filter noisy measurements, and make predictions and inferences about the monitored entity. Finally, the sensor stream reduction [6] strategies take advantage of the data stream algorithms characteristics to allow an *online* reduction of the data sensing based on application requirements, and are simple to implement.

From the three strategies, considering the application restrictions and the advantages of online data processing, we focus on sensor data stream reduction. Regarding the WSN application restriction, its use is motivated by three main factors. First, data transmission requires more energy than data measurement. Hence, reducing the data transmitted reduces the energy spent. Second, in order to reduce the data, the sensor node needs to perform constant and quick large local data processing, which requires simple and smart reduction strategies. Reduction strategies based on data streams are suitable here since they process the data locally and independently of previous data, avoiding the complete data storage and/or preprocessing. At last, as we have reduced bandwidth, sending large amounts of data can be problematic, causing excessive delay in response time and invalidating the data.

Although there are many sensor stream reduction strategies available in the literature [3], they usually consider application specific conditions in their implementation, turning their portability to different scenarios a difficult task. Hence, every time a new application comes at hand, a lot of work is involved in adapting these technics to the new scenarios. In this direction, this paper proposes a general framework for *sensor data stream reduction*. The framework takes into account the reduction methodology design, which shows how to reduce data as it is sensed or routed through the sink, and the reduced data validation process, which assesses if the reduced data is as representative as the original data.

The remainder of this article is organized as follows. We first describe the reduction design focusing in the architecture reduction. We show some data reduce validation that can applied. We present some specifics study case showing the efficiency of our methodology afterward. Finally, the open issues and conclusions are discussed.

II. Reduction Architecture Design

This section presents a general architecture for sensor stream reduction that can be easily applied to any WSN scenario aiming to reduce its energy consumption and data delay. The way the streams are reduced depends on the moment the reduction is going to be performed, i.e., during sensing or routing streams, and the type of data stream we are dealing with, i.e., the number of phenomena monitored by the stream generated by the sensor node.

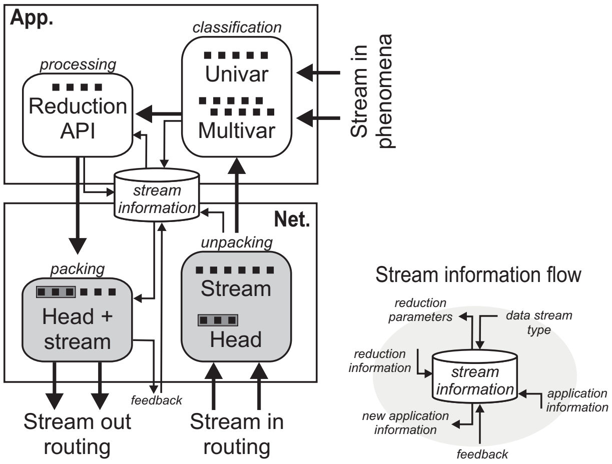


Figure 1. General architecture

The proposed methodology is illustrated in Fig. 1. As observed, the input streams can have been originated by the phenomena (sensing stream) or sent to the sensor node by another node (routing stream). The sensing reduction is recommended when the sensor device gets an excessive number of samples, and cannot be dynamically calibrated to deal with more data than it currently deals with. The routing reduction, in contrast, is performed when the network does not support the amount of data being transmitted. For instance, if the application has some requirements regarding the amount of data supported by each sensor node, data stream reduction can avoid uncontrolled data loss while guaranteeing the application requirements. Note that the sensing stream arrives in the application layer, while the routing stream arrives in the network layer.

When data arrives in the network layer, the network packets first need to be unpacked, separating the data stream from the header (that may contain some specific application information/restriction). Once it is unpacked, it is sent to the application layer, to be processed in the same way that sensing streams. At the same time, stream information is given to a cross-layer (labeled in Fig. 1 as “stream information”), responsible for making the interface between the application and network layers. The “stream information”, highlighted in Fig. 1, is responsible for choosing which reduction algorithm should be executed and its parameters. The information stored in this cross-layer includes:

- **Feedback**: Data received from other sensor nodes in order to perform the reduction calibration in an online fashion. For example, if the data reduction is dynamic, other nodes can inform the current node if more or less data can be propagated.
- **Application information**: Data received from the network layer when the stream is unpacked. Examples of application information are the deadline to stream delivery or global energy constraints.
- **Data stream type**: Data received from the application layer after the data stream is classified, and can be univariate or multivariate.
- **Reduction parameters**: Data given to the application layer in to guide it to perform the more appropriated reduction, considering the “application information” and the “data stream type”. Examples of reduction parameter is *Use a sampling algorithm with 50% reduction*.
- **Reduction information**: Data received from the application layer after the reduction. Examples of reduction information are the reduction level achieved or the reduced data size.
- **New application information**: Data given to the network layer for packing the reduced stream out. Examples of application information are the updated deadline for stream delivery, the new global energy constrains, or the new data stream size.

When data streams arrive to the application layer, they first have to classified according to the number of variables they monitor. In this context, data streams can be univariate or multivariate. Univariate streams are represented by a set of values read by a unique type of sensor, e.g., a sensor node that monitors only environmental temperature. On the other hand, multivariate streams are represented by a set of values coming from different sensors of the same sensor node, e.g., a node that monitors temperature, pressure and humidity simultaneously, or by a set of measurements coming from the same sensor type located in different sensor nodes, e.g., a node that processes data from different nodes monitoring only temperature. This classification is important because the data reduction process itself depends directly on the stream type.

After the stream type is known, we have to choose an appropriated stream reduction algorithm to effectively perform the reduction. There are various types of data stream reduction methods, such as online samples, histograms built, and sketches [3]. The reduction algorithms available in our API are depicted in Fig. 2, and explained below:

- **Random sampling**: This algorithm initially builds an histogram from the original stream. Then, for each histogram class, random elements are chosen to compose the reduced stream. The objective of this algorithm is to reduce the data keeping the class frequencies of the original histogram unchanged. It reduces the network energy consumption and delay by reducing the transmitted data while keeping its representativeness [6]. In Fig. 2(a), we show a "stream in" of 100 elements, from each 50% of its elements are randomly chosen to compose the "stream out".
- **Central sampling**: This algorithm is a variation of the random sampling. The main difference is that, instead of performing a random element choice, the central elements of the histogram classes are chosen to compose the reduced stream. In Fig. 2(b), we have a "stream in" of 100 elements, 50 of them are chosen considering the central histogram classes elements, generating the "stream out".
- **Sketch**: A data stream based algorithm that sketches data, reduce it through a data sketch, e.g., the minimum, maximum and average of a data or the data frequency. In our case, the sketch algorithm builds a histogram from the original stream, and uses the histogram class frequencies as the reduced data sketch. The sketch reduces the energy consumption and delay by keeping a constant transmission data rate, since the sketch size is fixed. After the histogram sketch arrives to the sink node, the data represented by the sketch can be artificially generated without loosing data quality. The only trouble in this strategy, when compared with the sampling, is that the sketch looses the sequence of data, despite of the good approximation when the original data is regenerated artificially [6]. In Fig. 2(c), we have a "stream in" with 100 elements, the histogram is built, and then a "stream out" is generated with the histogram class frequencies.
- **Multivariate sampling**: This algorithm uses principal component analyzes to help multivariate sampling. The principal components transformation is one of the most powerful tools for multivariate data treatment [7]. It is a transformation between γ-dimensional spaces, derived from the covariance matrix of the input data (in our case multivariate sensor data), generating a set of new data, where each resultant value is a linear combination of the original values. The number of principal components is equal to the number of dimensions of the original data and these principal components can be sorted according their variance. Thus, the first and the last principal component have the biggest and the smallest variance, respectively. In our algorithm the principal components of the original stream are computed. Then, the first component is sorted and used to rank the original stream. Based on this ranking and the data reduction size, the most correlated data are sampled [8]. In Fig. 2(d), we have a "stream in" with 100×5 elements (where each element represent data coming from different sensors in the same node or different sensor nodes), 50% of "stream in" is sampled considering the raking of first component analyzed, and then a "stream out" is generated with 50×5 elements.

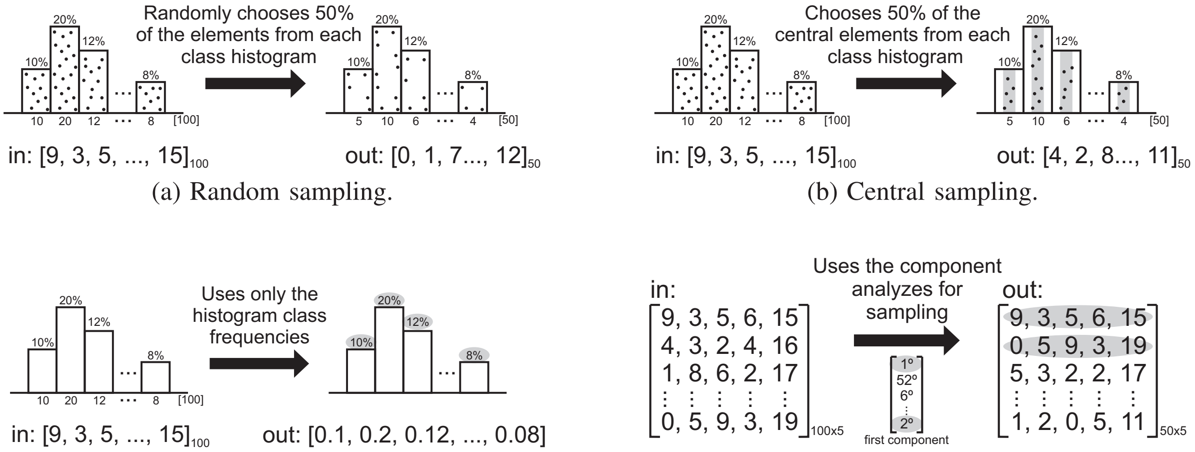
After the reduced data stream is obtained, if the stream was being routed, it is passed back to the network layer, which packs the stream and any information gathered from the cross-layer, and sends it to the sink.

III. Data reduced validation

After data stream reduction is performed, it is important to verify if the data quality of the original stream was preserved. If a sensor stream item with 100 elements is reduced for 50 elements, are these 50 elements representative?. Most of the time, data reduction validation is application specific. However, there are some simple tests that can be performed to validate the reduced data independent from the target application, considering the distribution of univariate streams, the variance analysis of multivariate streams, and the absolute relative error for both types of streams.

The distribution approximation between the original and reduced item streams can be done by the Kolmogorov-Smirnov test (KS test) [9]. This test evaluates if two univariate samples have similar distributions, and is not restricted to samples following a normal distribution. For example, if the original univariate stream item follows a Poisson distribution, this test verifies if the reduced item stream keeps the distribution characteristics. The analysis of variance (*ANOVA*) [10], used to validate multivariate data reduction, can be used to indicate if there is significant difference between the variances of the original and reduced multivariate streams.

It is also important to evaluate the discrepancy of the values in the reduced streams, i.e., if they still represent the original stream. Considering univariate data, this discrepancy can be quantified using the absolute value of the largest distance between the average of the original data and the lower or higher confidence interval values of the reduced


Randomly chooses 50% of the elements from each class histogram
10%
20%
12%
8%
10 20 12 ••• 8 [100]
5 10 6 ••• 4 [50]
in: [9, 3, 5, ..., 15]$_{100}$
out: [0, 1, 7..., 12]$_{50}$
Chooses 50% of the central elements from each class histogram
in: [9, 3, 5, ..., 15]$_{100}$
out: [4, 2, 8..., 11]$_{50}$
Uses only the histogram class frequencies
in: [9, 3, 5, ..., 15]$_{100}$
out: [0.1, 0.2, 0.12, ..., 0.08]
in:
9, 3, 5, 6, 15
4, 3, 2, 4, 16
1, 8, 6, 2, 17
0, 5, 9, 3, 19
100x5
Uses the component analyzes for sampling
1º
52º
6º
2º
first component
out:
9, 3, 5, 6, 15
0, 5, 9, 3, 19
5, 3, 2, 2, 17
1, 2, 0, 5, 11
50x5

(a) Random sampling.

(b) Central sampling.

(c) Sketch.

(d) Multivariate sampling.

Figure 2. Architecture API.

data average [6]. This same process can be applied to multivariate data, but in this case the final error used is the largest error among all relative errors for each variable, i.e., the error is calculated for each sensor (or variable) and only the highest of them will be considered [8].

IV. Methodology case study

In order to illustrate the application of the methodology proposed, this section presents two problems and the simulations performed to solve them following the steps described in the previous section. The first problem considers a general sensor stream application where a scheduled reduction has to be performed in the source node, i.e. it receives data about a phenomena for a certain time and then sends it to the sink. The second problem addresses a real time application where the reduction is performed during routing. Fig. 3 shows how the problems are addressed in the phenomenon view and the routing architectures.

In both scenarios, we consider a flat network that uses a shortest path tree based routing algorithm. The network density is kept constant (8 neighbors per node), and all nodes have the same software and hardware configuration. The phenomena monitored is always the same, and is represented by a normal distribution. The evaluation is performed through simulations using the NS-2 (Network Simulator 2) version 2.33. Other default simulation parameters, such as like radio range and bandwidth, were kept as $50\,\mathrm{m}$ and $250\,\mathrm{kbs}$, respectively. Each simulated scenario was executed with 33 random topologies.

In the first scenario, the network has 128 nodes, with different numbers of nodes (1, 5, 10, and 20) generating 256 items ($n = 256$) every 60s. Note that in this scenario only the application layer of our architecture is considered (phenomenon view on Fig. 3). Once the data is sensed, it is classified as univariate, and this information is sent to the "stream information" module, which sets the reduction parameters selecting the central sampling reduction algorithm (Fig. 2(b)), considering a reduction of $n/2$ and $\log n$, where n is the size of the stream sensed, i.e., n temperature or pressure samples. After the reduction step, the reduced stream is routed to sink. Observe that feedback and reduction information are not considered in this design, because the application requires only the local reduction, i.e., the reduction in sensing moment.

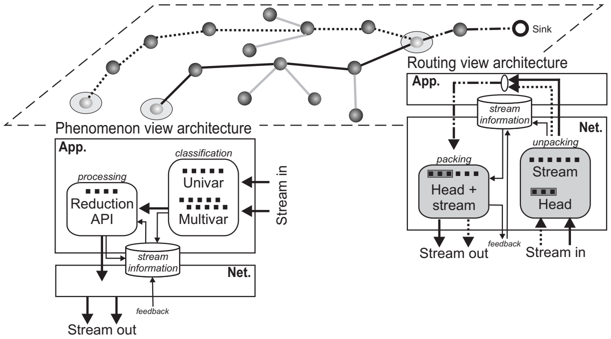
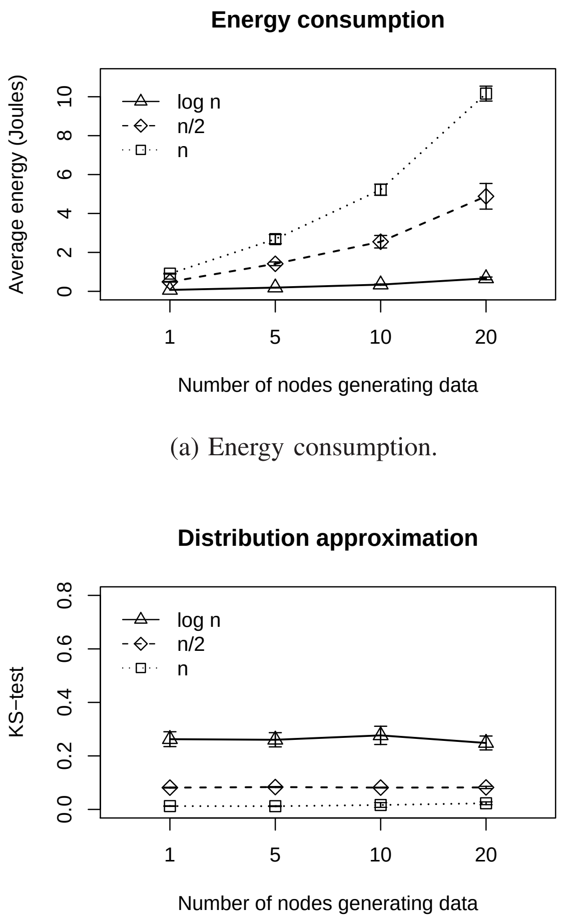


Figure 3. General architecture

Fig. 4 shows the average value of energy consumption and the difference in the original e reduced data distribution using the KS-test with a 95% confidence interval. As showed in Fig. 4, the sample $\log n$ reduces the energy consumption by reducing the transmitted data. However, the original data distribution is affected by 20%. This quality is acceptable by the large majority of applications when the network restrictions are strong. The sample of $n/2$ is interesting when the application does not have strong restrictions.

The second scenario simulated concerned a real time application, which works in both the application and network layers (phenomenon and routing view on Fig. 3). Again, only

ISBN: 978-1-61208-144-1



(a) Energy consumption.

(b) Data validation.

Figure 4. Example of data reduction in a general sensor stream application.

univariate data is considered, and processed by the random and central sampling reduction algorithms, considering a reduction controlled by stream information. The stream items arrive in the network layer and is unpacked. The application information (packet head) is separated from the data application (stream) and it is passed to the stream information. However, the stream information receives the feedback of other nodes to reduce more or less data and the reduction parameters used. The packing receives from the stream information the new application information and can sent some feedback to other nodes.

The real time application operation considers: (i) again the stream item represents n samples of environment; (ii) the application has a soft deadline to deliver the stream item sensed, this deadline is packing with the stream item; (iii) the stream item is fragmented and routed through to sink; (iv) the router nodes look each packet and check if the stream item can be delivered; (v) if the deadline cannot be achieved the stream item is resembled (remember that it was fragmented) and reduced according the acceptable amount of data, e.g., the stream item with 256 elements cannot be delivered, so the router reduce it to 100 elements that can be delivered on time; and (vi) the stream reduced is repacking, now with the new application deadline, fragmented and forwarded through to sink.

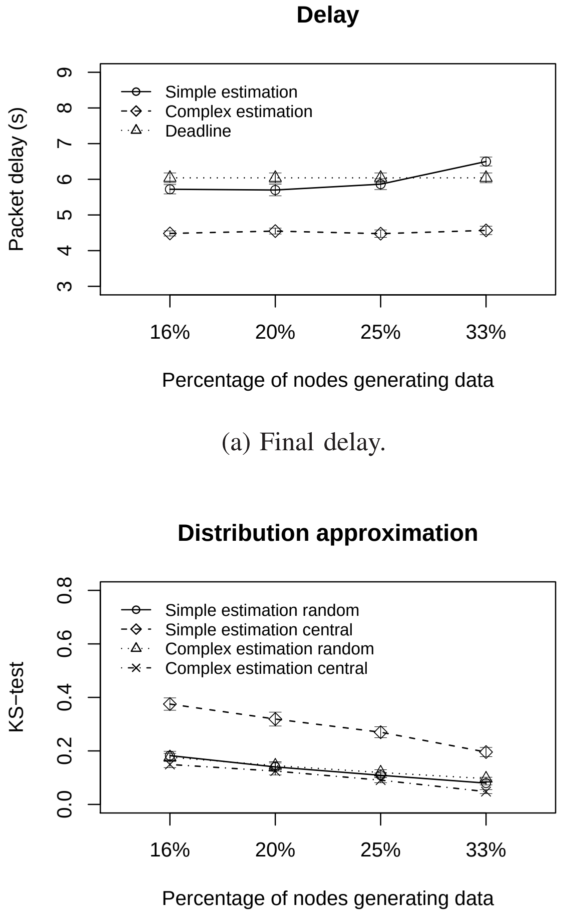
To illustrate the reduction solution in this real time applications we consider again a flat network that uses a shortest path tree based routing algorithm, the network density is kept constant (8 neighbors per node), and all nodes have the same software and hardware configuration. The phenomenon monitored is always the same, it is represented by a normal distribution. Like in general application, we perform our evaluation through simulations and use the NS-2 (Network Simulator 2) version 2.33. Each simulated scenario was executed with 33 random topologies. At the end, for each scenario we plot the average value with 95% of confidence interval. Other default simulation parameters are used, like radio range $50\,\text{m}$ and bandwidth $250\,\text{kbs}$.

To simulate the real time application we consider a minimum deadline (get empirical) that the "perfect network" supports. To force the in-network reduction we use concurrent traffic and all router nodes delay the packet fragments at 0.01% of the initial deadline. We set among the 128 nodes distributed in the network 16%, 20%, 25%, and 33% of this nodes generating extra traffic. However we stress the system to consider 2048 elements (n) in stream item. However, in order to highlighted the importance of the reduction controlled by stream information, we consider two estimation way: a simple that analyzes only the local node time; and a complex that tries to infer what happens during the data traffic.

In the Fig. 5(a) the application deadline is met in almost cases. The complex estimation presents a more scalable behavior considering the percentage of nodes generating data. This occurs because this estimation infers better the data traffic behavior during the routing. The Fig. 5(b) shows the simple and complex estimation using the random and central sample reduction algorithms. It is showed that in all cases we have a distribution approximation $\leq 40\%$. The central sampling algorithm with the complex estimation has a smaller error. The reason is that the complex estimation performs the maximum reduction sooner (the central algorithm is executed once or twice). This result shows that because fewer successive reductions are performed, more representativeness is kept in the reduced data, i. e., data degradation is mitigated.

V. Open issues and conclusions

This work presented a general methodology to perform sensor stream based reduction in WSNs. This methodology considered the application requirements, the reduction design, and the data reduce validation. Specifically, to reduction design was presented a architecture that can be applied



(a) Final delay.

(b) Data validating.

Figure 5. Reduction in real time stream applications.

to reduce the data when it is sensed or routed through to sink. The study cases showed the usefulness of our methodology applied on a general sensing and a real time applications. Furthermore, the methodology proposed is general enough to be applied to design reduction scenarios in which we have some application requirements.

Among some open issues, we can consider a better evaluation of the proposed methodology by considering other network scenarios, and matching the proposed application level solution with lower level ones. However, consider the architecture, not only the data from a source is reduced, but similar data from different sources can be also reduced, resulting in a more efficient reduction solution.

Acknowledges

This work is partially supported by the Brazilian National Council for Scientific and Technological Development (CNPq) under the grant number 477946/2010-0 and the Research Foundation of the State of Minas Gerais (FAPEMIG) under the grant number CEX-APQ-00577-09.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, August 2002.

[2] T. Arampatzis, J. Lygeros, and S. Manesis, "A survey of applications of wireless sensors and wireless sensor networks," in *13th IEEE Mediterranean Conference on Control and Automation (MED'05)*. Hawaii, USA: IEEE Computer Society, June 2005, pp. 719–724.

[3] S. Muthukrishnan, *Data Streams: Algorithms and Applications*. Hanover, MA, USA: Now Publishers Inc, January 2005.

[4] S. Santini and K. Romer, "An adaptive strategy for quality-based data reduction in wireless sensor networks," in *3rd International Conference on Networked Sensing Systems (INSS'06)*. Chicago, IL, USA: Transducer Research Foundation, 31 May – 2 June 2006, pp. 29–36.

[5] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Computing Surveys*, vol. 39, no. 3, pp. 9/1 – 9/55, April 2007.

[6] A. L. L. Aquino, C. M. S. Figueiredo, E. F. Nakamura, L. S. Buriol, A. A. F. Loureiro, A. O. Fernandes, and C. N. C. Junior, "Data stream based algorithms for wireless sensor network applications," in *21st IEEE International Conference on Advanced Information Networking and Applications (AINA'07)*. Niagara Falls, Canada: IEEE Computer Society, May 2007, pp. 869–876.

[7] J. E. Jackson, *A User's Guide to Principal Components*, 1st ed. Wiley-Interscience, 2003.

[8] O. S. Junior, A. L. L. Aquino, R. A. F. Mini, and C. M. S. Figueiredo, "Multivariate reduction in wireless sensors networks," in *IEEE Symposium On Computers and Communications (ISCC'09)*. Sousse, Tunisia: IEEE Computer Society, July 2009.

[9] S. Siegel and J. N. John Castellan, *Nonparametric Statistics for the Behavioral Sciences*, 2nd ed. Columbus, OH, USA: McGraw-Hill Humanities/Social Sciences/Languages, January 1988.

[10] N. Thomson, "Understanding ANOVA the APL way," *ACM SIGAPL – APL Quote Quad*, vol. 24, no. 1, pp. 295–303, August 1993.

HDLS: Improved Localization via Algorithm Fusion

Ralf Behnke, Jakob Salzmann, Philipp Gorski, Dirk Timmermann
Institute of Applied Microelectronics and Computer Engineering
University of Rostock
Rostock, Germany
{*ralf.behnke, jakob.salzmann, philipp.gorski, dirk.timmermann*}@*uni-rostock.de*

***Abstract*—Wireless Sensor Networks (WSNs) have been of high interest during the past couple of years. One of the most challenging tasks of WSN research is still location estimation. As a well performing fine grained localization approach, Distributed Least Squares (DLS) was introduced, splitting the costly localization process in a complex *precalculation* and a simple *postcalculation*, which is performed on constrained sensor nodes. Nevertheless, as size of precalculation and consequently, cost of computation and communication are growing with network size, it was shown that this algorithm is unsuitable for large WSNs. This restriction has been overcome by scalable DLS (sDLS), which enables to use the idea of DLS in large WSNs for the first time. Although cost of computation of sDLS is independent of the network size, it was relatively high, due to costly matrix updates. Consequently, this cost was reduced by sDLS with normal equation (sDLS^{ne}), circumventing the updates. Unfortunately, sDLS^{ne} comes along with a decreased localization accuracy. The approach, presented in this work, combines the efficient sDLS^{ne} approach with various coarse grained localization techniques to improve localization accuracy. The resulting localization accuracy overcomes the efficient sDLS^{ne} approach as well as the more precise sDLS approach, while cost of computation still outperforms sDLS.**

***Keywords*-wireless sensor networks, localization, scalability.**

I. Introduction

Recent technological advances enabled development of tiny wireless devices, which are able to sense their environment, compute simple tasks and exchange data among each other. Interconnected assemblies of such devices, called Wireless Sensor Networks (WSNs), are commonly used to observe large inaccessible areas. In many applications of WSN, knowledge of nodes' locations is mandatory for a meaningful interpretation of measured data. In addition, location-awareness is also necessary for geographic routing [1][2] or location based clustering [3]. Due to existing limitations in terms of size, financial cost and energy consumption, local positioning within the network is preferred over utilizing Global Navigation Satellite Systems (GNSSs) like GPS [4]. Therefore, the presence of location-aware sensor nodes, referred to as beacon nodes, is typically assumed. The remaining nodes, which we refer to as blind nodes, are assumed to use communication and any kind of distance estimation or neighborhood information to estimate their positions with the help of beacon nodes.

Localization algorithms can be divided into centralized and decentralized on the one hand, and fine-grained and coarse-grained on the other hand. Coarse-grained approaches like Centroid Localization (CL) [5], Weighted Centroid Localization (WCL) [6] and Adaptive Weighted Centroid Localization (AWCL) [7] often abstain from exact distances, require less communication and computation, and provide lower precision estimates. In contrast, fine-grained approaches use costly computations and distance estimations to achieve localization with high precision. High precision and low complexity have been firstly combined by Distributed Least Squares (DLS) [8], which splits the costly localization calculation into *precalculation* and *postcalculation*. Independent from a specific blind node, the complex precalculation is performed on a high performance sink. The remaining postcalculation is less complex and performed on resource-constrained blind nodes.

The concept of DLS has been adapted by scalable DLS (sDLS) [9], which enables the idea of DLS to be used in large WSNs. In contrast to DLS, sDLS provides costs of computation and communication, incurred on blind nodes, which are independent from network size, i.e., independent from total number of beacon nodes. This is achieved by use of individual precalculations instead of one global precalculation. A fundamental enhancement is given by sDLS with normal equation (sDLS^{ne}) [10], which significantly reduces the cost of computation by circumventing costly updates, introduced with sDLS.

Using sDLS, blind nodes are assumed to choose one precalculation out of several precalculations provided by neighbouring beacon nodes, according to their distances. Commonly, the set of beacon nodes included in the chosen precalculation differs from the set of beacon nodes within a blind node's communication range. This causes a suboptimal localization accuracy and offers possibilities for further improvements. The present work combines multiple position estimates, based on sDLS^{ne}, by use of coarse grained localization techniques, to improve localization accuracy.

The remainder of the paper is organized as follows. Section II covers basic informations about sDLS algorithms. In Section III, the new hybrid localization approach is presented in various variants, using several optimizing parameters. Section IV covers performed simulations. Simulation results are presented in Section V. Finally, the presented work is summarized in Section VI.

II. Related Work

The DLS algorithm was developed to diminish trade off between precision and cost of localization [8]. It provides localization with high precision at low cost. The basic idea of splitting the calculation into precalculation and postcalculation was adapted by sDLS and its successor sDLS^{ne} to support large WSNs with network size independent cost for blind nodes. Both approaches are briefly described in this section.

The system of equations, which have to be solved for localization of a blind node is originally built by distance equations as given in equation (1).

$$(x-x_i)^2+(y-y_i)^2 = r_i^2 \quad (i \in I; I = \{1,2,\ldots,m\}) \tag{1}$$

Here x and y give the unknown position of a blind node. The known position of a beacon node is denoted as x_i and y_i, while the distance between both nodes is denoted as r_i. The number of beacon nodes utilizable for localization is given as m.

This system of equations is linearized by use of a linearization tool [11], using one beacon node as linearizer, denoted with index L. After restructuring, the system of equations consists of equations as given in equation (2), where r_L denotes the distance between blind node and linearizer, r_i is the distance between blind node and beacon node, and d_{iL} denotes the distance between linearizer and beacon node.

$$\begin{aligned} b_{iL} &= (x-x_L)(x_i-x_L)+(y-y_L)(y_i-y_L) \\ &= \frac{1}{2}\left[r_L^2 - r_i^2 + d_{iL}^2\right] \end{aligned} \tag{2}$$

After further restructions, the system of equations matches the matrix form $\mathbf{Ax} = \mathbf{b}$, using $\mathbf{A}$, $\mathbf{x}$ and $\mathbf{B}$ as given in equation (3).

$$\begin{gathered} \mathbf{A} = \begin{pmatrix} x_{k_1}-x_L & y_{k_1}-y_L \\ x_{k_2}-x_L & y_{k_2}-y_L \\ \vdots & \vdots \\ x_{k_n}-x_L & y_{k_n}-y_L \end{pmatrix}, \\ \mathbf{x} = \begin{pmatrix} x-x_L \\ y-y_L \end{pmatrix}, \mathbf{b} = \begin{pmatrix} b_{k_1L} \\ b_{k_2L} \\ \vdots \\ b_{k_nL} \end{pmatrix} \end{gathered} \tag{3}$$

Here, the beacon nodes, used for localization, are denoted with indices $K = \{k_1, k_2, \ldots, k_n\}$ with $K = \{I \setminus L\}$. Matrix $\mathbf{A}$ of equation (3) only consists of beacon position data, while $\mathbf{b}$ contains distances between beacon nodes and blind nodes. Therefore, calculations on matrix $\mathbf{A}$ are to be performed as part of the precalculation at a powerful sink outside the WSN. The localization will be finalized on each blind node by performing the remaining part of the calculation.

To solve the linear system of equations, using normal equations, equations (4a) to (4c) are used. While (4a) shows the entire equation, (4b) presents the precalculation, performed on the sink, and (4c) presents the postcalculation, performed on blind nodes.

$$\mathbf{x} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\frac{1}{2}\left[r_L^2 - \mathbf{r}^2 + \mathbf{d}^2\right] \tag{4a}$$

$$\begin{aligned} \mathbf{A}_p &= \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T \\ \mathbf{d}_p &= \mathbf{d}^2 \end{aligned} \tag{4b}$$

$$\mathbf{x} = \mathbf{A}_p\frac{1}{2}\left[r_L^2 - \mathbf{r}^2 + \mathbf{d}_p\right] \tag{4c}$$

The main difference between DLS and sDLS^{ne} is given by number and size of precalcultions. Regarding beacon nodes in a WSN, $\mathbb{G}$ is considered as the global set of all beacon nodes and $\mathbb{L}_i \subseteq \mathbb{G}$ denotes a local set of beacon nodes within the communication range of beacon node i. While DLS uses only one precalculation, including all beacon nodes, i.e., equation (3) with conditions $K = \{\mathbb{G} \setminus L\}$ and $L = 1$, sDLS^{ne} uses individual precalculations for all beacon nodes, i.e., $|\mathbb{G}|$ precalculations using $K = \{\mathbb{L}_i \setminus L\}$, $L = i$, $\forall i \in \mathbb{G}$. Therefore, the sDLS^{ne} algorithm starts with an additional discovery phase to find other beacon nodes in one hop distance, as illustrated in Figure 1.

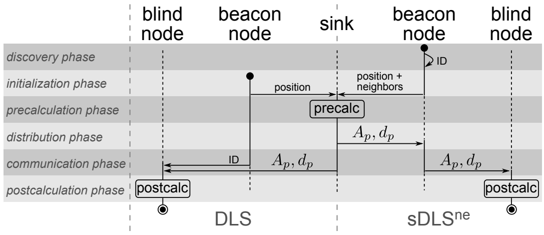


Figure 1. Algorithmic comparison of DLS and sDLS^{ne}

Furthermore, DLS needs an explicit communication with all beacon nodes during the communication phase for distance estimation. Using sDLS^{ne}, this is an implicit process as each blind node receives precalculations from beacon nodes in its own communication range.

Using sDLS^{ne}, each beacon node provides its own precalculation, which would perfectly fit for a blind node on the same position. From all offered precalculations, blind nodes are expected to chose the one of the closest beacon node.

III. Hybrid Localization Approach

The original intention of sDLS was to use exactly those beacon nodes, which are located within the communication range of the blind node, attempting to estimate its own position. To achieve this goal, a blind node is expected to choose the precalculation provided by the beacon node closest to its own position. Consequently this precalculation includes most of the beacon nodes within the blind node's communication range. Nevertheless, in most cases some beacon nodes included in this precalculation are outside the communication range of the blind node and vice versa. While sDLS locally updates this precalculation, by use of matrix updates, to achieve the initial intention, sDLS^{ne} estimates the unknown position with this unprecise precalculation.

Due to this displaced set of beacon nodes as well as the high influence of the node geometry, especially the given choice of the linearizing beacon node, the resulting position estimation tends to be drawn in the direction of this beacon node. In addition, the used distance estimation also causes an impairement of the position estimation. Furthermore, a defective distance estimation may cause the blind node to spuriously choose a precalculation of a beacon node, which is not the closest.

The aim of Hybrid Distributed Least Squares (HDLS) is to use multiple precalculations of nearby beacon nodes. The resulting position estimates, according to each chosen precalculation, serve as tentative results. These results can be seen as virtual beacon nodes. They will be combined to a final position estimate using coarse grained localization techniques. For that aim, various approaches have been studied in this work. The used coarse grained localization approach presents only one factor, that influences the resulting accuracy. The following factors, studied in our work, are to be further explained in this section:

Strategy:	number of virtual beacon nodes
Technique:	used coarse grained approach
Weightage:	used weight factor
Reduction:	reduction part, used by AWCL
Approximation:	distance approximation of inaccessible beacon nodes

A. Virtual Beacon Strategy

To control the number of virtual beacon nodes that are to be created using sDLS^{ne}, the following strategies have been investigated:

Closest Two – Virtual beacon nodes are created from precalculations of the two closest beacon nodes.
Closest Three – Virtual beacon nodes are created from precalculations of the three closest beacon nodes.
Great Deal – Virtual beacon nodes are to be created, using precalculations of all beacon nodes in range.
Range Based – Beacon nodes in a range, given as a multiple of the distance to the closest beacon node, are used for creation of virtual beacon nodes. This strategy extends the before mentioned strategies, which serve as upper bound. Within our investigations, this range has been varied from 125% up to 250% of the closest beacon node.

B. Coarse Grained Estimation Technique

Created virtual beacon nodes are combined to a resulting position estimation P_b using coarse grained localization techniques. The following techniques have been studied:

CL – The plain Centroid Localization (CL) approach is used to combine the virtual beacon nodes, i.e., unweighted arithmetic mean is used as given in equation (5). Here, $\mathbb{V}$ indicates a set of given virtual beacon nodes and P indicates a position.

$$P_b = \frac{1}{|\mathbb{V}|} \sum_{i \in \mathbb{V}} P_i \tag{5}$$

WCL – Virtual beacon nodes are combined using Weighted Centroid Localization (WCL) as given in equation (6). Suitable substitutions for weight w_i are to be presented subsequently. Common weights rely on measured distances or received signal strength (RSS).

$$P_b = \frac{\sum_{i \in \mathbb{V}} P_i * w_i}{\sum_{i \in \mathbb{V}} w_i} \tag{6}$$

AWCL – Virtual beacon nodes are combined by use of Adaptive Weighted Centroid Localization (AWCL). While WCL simply gives more influence to closer beacon nodes, i.e., beacon nodes with higher weight, the idea of AWCL is to give more influence to the difference of given weights. Therefore, if the weights, e.g., RSS, of beacon nodes in range are similar to each other, they are to be reduced by a reduction part q of the smallest weight, with $\{q \in \mathbb{R} | 0 \leq q \leq 1\}$, as illustrated in Figure 2. Otherwise, i.e., in case of high differences within the weights, AWCL inherently acts as WCL.

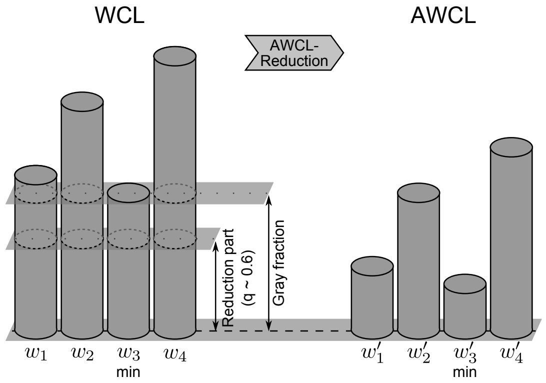


Figure 2. Illustration of the reduction, used by AWCL

Various reduction parts, referred to as q in equation (7), have been investigated, as described as follows.

$$P_b = \frac{\sum_{i \in \mathbb{V}} P_i * (w_i - q * min_{i \in \mathbb{V}}(w_i))}{\sum_{i \in \mathbb{V}} w_i - q * min_{i \in \mathbb{V}}(w_i)} \tag{7}$$

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

C. Weightage

Except from the plain CL algorithm, the presented coarse grained estimation techniques are utilizing weighting factors. The aim of weights is to give higher influence to more important (virtual) beacon nodes. In the given case a precalculation is defined as more important, if the accordant beacon node and therefore the linearizer is closer to the blind node. In the same way, it is more important if the number of beacon nodes included in the precalculation and in the blind node's communication range is high. Consequently the following weights have been studied:

Signal Strength – Virtual beacon nodes are weighted according to the RSS of the beacon node that provided the precalculation used to create the virtual beacon node. On average, the RSS is expected to be higher the closer the beacon node is. Although, variations of shadowing and fading may compromise this relation, it has been investigated as possible weightage. Equation (8) illustrates this weight, with i indicating the linearizer of the according precalculation as well as the resulting virtual beacon node.

$$w_i = RSS_i \tag{8}$$

Similarity – Virtual beacon nodes are weighted according to the rate of beacon nodes, included in precalculation, that are located within the communiction range of the blind node. This weight is given in equation (9), where $\mathbb{P}_i$ indicates the set of beacon nodes included in the precalculation of beacon node i and $\mathbb{B}$ indicates the set of beacon nodes within the communication range of the blind node. This is applied to the WCL approach, which is then called Similarity based WCL (SWCL).

$$w_i = \frac{|\mathbb{P}_i \cap \mathbb{B}|}{|\mathbb{P}_i|} \tag{9}$$

D. Reduction part

AWCL has been shown as more accurate than the original WCL. In advance of an included WCL estimation AWCL reduces all given weights by a certain portion of minimum weight, as given in equation (7). This leads to the behavior that in case of nearby weights the remaining small differences get more importance. For our investigations, the used reduction part q has been varied from 15% to 65%.

E. Distance approximation

To enable a blind node to use beacon nodes outside its own communication range, sDLS^{ne} introduced a distance approximation, given in Figure 3, that utilizes the given distance between linearizer and inaccessible beacon node (d_{iL}), and the estimated distance between blind node and linearizer (r_L), which was assumed to be as close as possible, due to the prior choice of the blind node. The sum of both distances is used as approximation of the unknown distance r_i.

Now, using not only the closest beacon node, but up to all beacon nodes within the communication range, this approximation tends to be more and more inaccurate. Therefore, two variants of this distance approximation have been investigated.

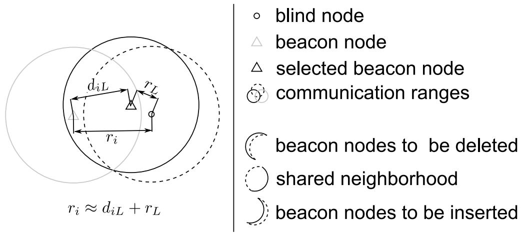


Figure 3. Approximation of a distance between blind node and inaccessible beacon node i by means of the linearizing beacon node L

Independent Approximation – For each precalculation distances to inaccessible beacon nodes are estimated as given in Figure 3. All data used is either directly estimated by use of measurements or provided by the precalculation itself.

Dependent Approximation – Most inaccessible beacon nodes are included in multiple precalculations, provided to the blind node. As illustrated in Figure 4, distance approximations towards such an inaccessible beacon node will differ according to the used precalculation, due to different linearizer nodes used in different precalculations.

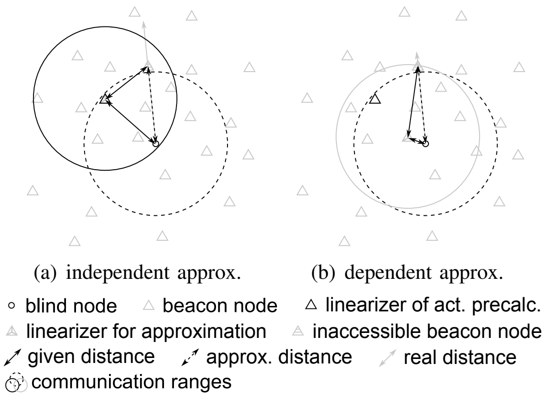


Figure 4. Two approximation strategies: To estimate distances of inaccessible beacon nodes, included in a precalculation, either (a) the beacon node, providing the precalculation or (b) the closest beacon node is used.

To provide the most precise distance estimation the shortest distance, which can be estimated from the given precalculations, have to be selected for calculation of virtual beacon nodes.

To achieve this, one possibility is to firstly determine all possible estimates for inaccessible beacon nodes, i.e., one for each precalculation, which includes the inaccessible node, to subsequently calculate the minimum distance estimations. In most cases a more efficient solution can be applied. Figure 5 illustrates such a solution compared along with independent approximation in the context of the over all position estimation, given on the right hand side. The illustrated approach

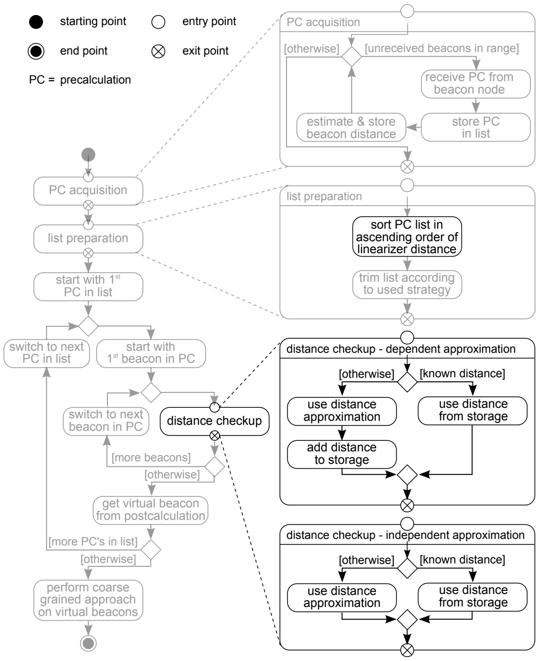

starting point
entry point
end point
exit point
PC = precalculation
PC acquisition
[otherwise]
[unreceived beacons in range]
receive PC from beacon node
estimate & store beacon distance
store PC in list
PC acquisition
list preparation
list preparation
sort PC list in ascending order of linearizer distance
trim list according to used strategy
start with 1st PC in list
switch to next PC in list
start with 1st beacon in PC
switch to next beacon in PC
distance checkup
[more beacons]
[otherwise]
get virtual beacon from postcalculation
[more PC's in list]
[otherwise]
perform coarse grained approach on virtual beacons
distance checkup - dependent approximation
[otherwise]
[known distance]
use distance approximation
use distance from storage
add distance to storage
distance checkup - independent approximation
[otherwise]
[known distance]
use distance approximation
use distance from storage

Figure 5. Differences of independent and dependent distance approximation (left), illustrated in the context of the HDLS algorithm (right).

processes precalculations individually but in ascending order of their distances towards the blind node, as illustrated on the right side. For this purpose, the distance between blind node and linearizer acts as the distance towards its according precalculation. Once a distance towards an inaccessible beacon node has been approximated by a close precalculation, this distance will be marked as known, as illustrated in the last but one box on the left side of Figure 5. If the same beacon node occurs in a further precalculation, the before calculated distance will be taken and the beacon node will be not treated as inaccessible.

Figure 4 illustrates the presented strategies by giving a worst case example for both approaches. Using independent approximation for the precalculation illustrated in Figure 4(a) highly overrates the distance towards the inaccessible beacon node. By use of dependent approximation instead, the better approximation provided by a closer precalculation illustrated in Figure 4(b) would be used.

IV. Simulations

To verify performance of the introduced HDLS approaches, the MATLAB® based network simulator Rmase is used [12]. The simulator provides a realistic radio communication model including spatial and temporal normal distributed fading. A static bidirectional spanning-tree routing was used to send data packets from nodes to sink and vice versa. Distance estimations performed by blind nodes rely on the simulators radio model.

A random deployment of n^2 nodes within a field of $n * n$ arbitrary distance units (adus) was utilized. The first node was always used as sink, while the remaining nodes have been randomly chosen as blind nodes (50%) or beacon nodes (50%). Note that the low number of blind nodes has been proofed to has no significant influence on the presented results but speeds up the simulation dramatically. The field size parameter n was varied from 5 to 30. The average communication range, given by the radio model, was 3 adus. For each field size the average over 100 simulations has been determined. In each simulated network all presented localization approaches have been performed concurrently.

V. Results

As described in Section III, there are various factors, influencing the accuracy of HDLS. To distinguish between the different approaches, resulting from these factors, a naming scheme is used, illustrated as syntax diagram in Figure 6. This diagram also shows the more than 350 combinations, which have been investigated by simulations. In Figure 6, ”S” symbols similarity based weightage, applied to WCL. Reduction part of AWCL have been varied from 15% to 65%. Range based strategy, indicated with an ”R”, also denotes a percentage of the distance towards the closest beacon node, which limits the catchment area of further beacon nodes. It is used in addition to the fixed upper bound of virtual beacon nodes.

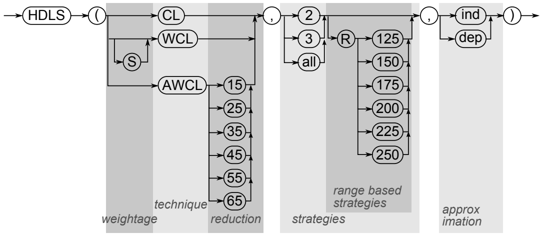


Figure 6. Syntax diagram: Naming of investigated HDLS approaches

First, used coarse grained techniques are analysed along with different virtual beacon strategies, i.e., the number of virtual beacon nodes. Figure 7 shows mean localization error over the number of deployed nodes, using the basic CL approach. It is illustrated, that the hybrid approaches perform significantly better than the underlying sDLS^{ne}, but in most cases not as accurate as the original sDLS approach with costly matrix updates. Furthermore, it is shown that the hybrid approach with two virtual beacon nodes is outperformed by the one, using three virtual beacon nodes. In contrast, using as much virtual beacon nodes as possible does not further increase localization accuracy.

Similar results have been found for HDLS based on WCL with traditional RSS based weighting, shown in Figure 8. It is shown that this approach performs the better, the more virtual beacon nodes are used. Futhermore, it outperformes

ISBN: 978-1-61208-144-1

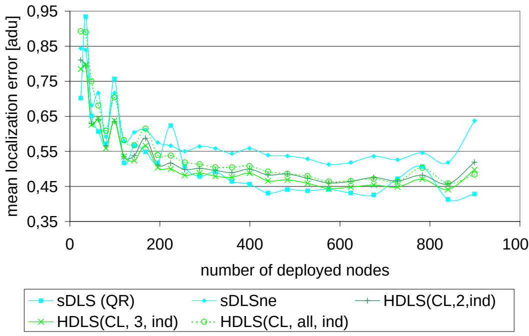


Figure 7. Mean localization error of CL based HDLS with independent distance estimation

sDLS and therefore it also outperforms the CL based approach.

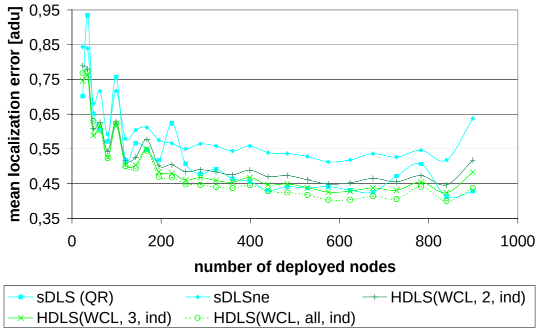


Figure 8. Mean localization error of WCL based HDLS with independent distance estimation

In Section III, SWCL has been introduced as an alternative approach of WCL, using similarity instead of signal strength. Both WCL based approaches are compared in Figure 9. On the one hand, the illustration shows that similar to the CL based approach, the SWCL approach performs best, when three virtual beacon nodes are used. On the other hand, it is shown that this approach is outperformed by the RSS based approach.

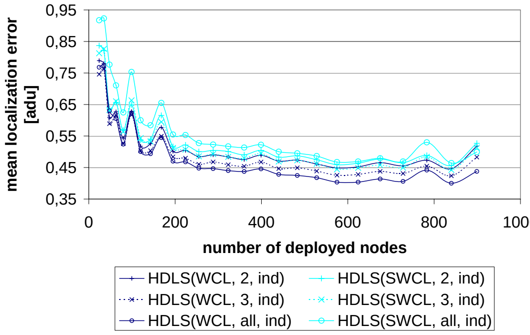


Figure 9. Mean localization error of HDLS based on WCL and SWCL with independent distance estimation

The third coarse grained technique, investigated to use with HDLS, is AWCL. The performance of AWCL depends on a reduction part, defined by AWCL. The best reduction part is said to be 55%. Therefore, this factor is also used for the results, given in Figure 10. The presented results show, that this approach also outperforms the costly sDLS approach and performs the better the more virtual beacon nodes are used. Further investigations, using different reduction factors showed that also in the given context a reduction factor of 55% performs best in most cases. Nevertheless, achieved accuracy is often influenced only marginal by the reduction factor.

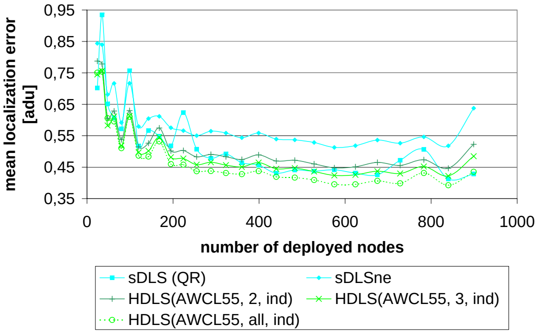


Figure 10. Mean localization error of AWCL based HDLS with independent distance estimation

As an intermediate result HDLS provides best accuracy, using as much virtual beacon nodes as possible, combined by AWCL with a reduction part of 55%. While the previous results used virtual beacon strategies with a fixed number of virtual beacon nodes, the following results investigate the range based virtual beacon strategy. The range within precalculations of beacon nodes are used to create virtual beacon nodes was varied from 125% to 250% of the distance between blind node and closest beacon node. The range based approach is combined with a fixed upper bound as presented before. Figure 11 shows the resulting localization accuracy for CL based HDLS, using various ranges and an upper bound of two, i.e., HDLS falls back into sDLS^{ne}, if the closest beacon node is significantly closer than all other beacon nodes. On the one hand, it is shown, that even a small range of 125% outperforms sDLS^{ne}. On the other hand, the graph shows, that only in few cases, this spatial limit outperforms the unlimited version. It also shows that in most cases a spatial limitation of 175% performs very close to the unlimited counterpart. Similar results have been found for the use of WCL, SWCL or AWCL.

As AWCL turned out as the most promising approach, it is selected to compare the range based strategy with various upper limits of virtual beacon nodes. Figure 12 shows the results for the previously introduced upper bounds in combiniation with the spatial limits of 125% and 250%. On the one hand, the results show that the range based strategy

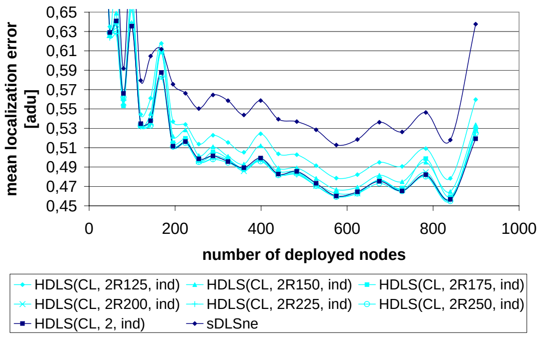


Figure 11. Mean localization error of CL based HDLS with independent distance estimation and range based virtual beacon strategy

also works for limits higher than two. On the other hand, it is shown that the higher the spatial limit, the lower the mean localization error. Although the unlimited approaches perform better than the corresponding limited approaches, it comes out that a spatial limit of 250% achieves good results.

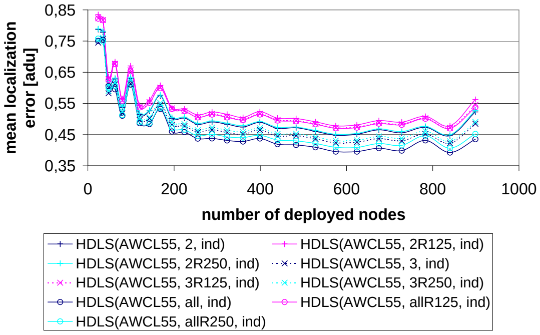


Figure 12. Mean localization error of AWCL based HDLS with independent distance estimation and range based virtual beacon strategy

To evaluate the range based strategy as an alternative to the before mentioned strategies of fixed limits, spatial limits have been figured out, which are equivalent to numerous limits. As illustrated in Figure 13, a spatial range of 150% can be put on a level with the upper bound of two virtual beacon nodes. A spatial limit of 200% instead can be equated with the strategy of using 3 virtual beacon nodes. Once again, as much beacon nodes as available is proved to provide lowest localization error. Nevertheless, a spatial limit of 250% provides also good results.

Using a spatial limitation instead of a fixed number of virtual beacon nodes can be only seen as alternative, if it is more cost efficient. Therefore, the number of arithmetic operations, used for the according localization approach, has been investigated. Figure 14 illustrates this cost for the HDLS approaches, presented in Figure 13. It clearly comes out that the two range based approaches, which have been pointed out as equivalents need slightly more computations than the corresponding approaches.

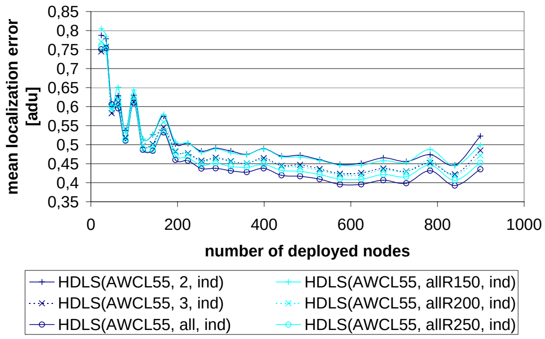


Figure 13. Range based equivalents of AWCL based HDLS approaches with independent distance estimation – mean localization error

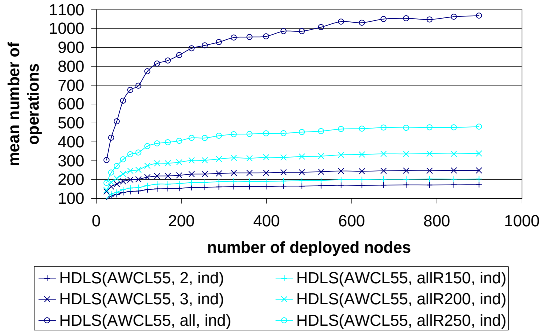


Figure 14. Range based equivalents of AWCL based HDLS approaches with independent distance estimation – mean number of operations

Up to this point, the presented results are based on independent approximation of distances towards inaccessible beacon nodes. The remaining part of this section presents the results, achieved by use of dependet approximation. As shown in Figure 15, use of this approximation significantly improves localization accuracy of CL based HDLS. It outperforms sDLS as well as the best CL approach with independent approximation, even if only two virtual beacon nodes are used. It is also shown that there is only a small gain, which distinguishes the all beacon strategy from the three beacon strategy. Similar results have been found using WCL, SWCL and AWCL. In all cases, each approach using dependent distance approximation outperforms the according HDLS approach based on independent distance approximation, using as much virtual beacon nodes as possible.

To sum up the before mentioned results and to figure out the best HDLS approach for each coarse grained technique the best performing approach is presented in Figure 16. Noticeable, but not surprising, best results are achieved using as much virtual beacons as possible. Furthermore, Figure 16 shows impressively that use of dependent distance approximation outperforms independent distance approximation. Using dependent approximation, the AWCL based approach performs best closely followed by WCL. The same

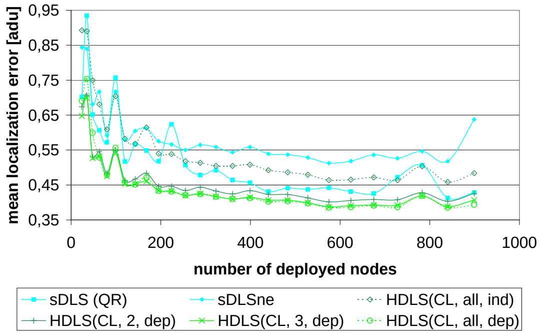


Figure 15. Mean localization error of CL based HDLS with dependent distance estimation

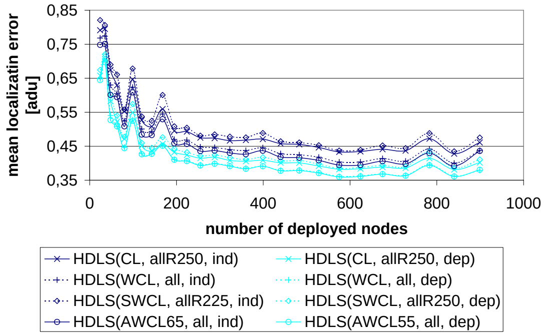


Figure 16. Mean localization error of best performing HDLS approaches, grouped by estimation technique and distance approximation

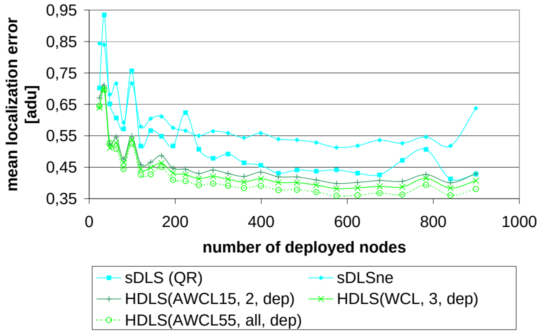


Figure 17. Mean localization error of best performing HDLS approaches for different limits of virtual beacon nodes

order is depicted for independent distance approximation. Furthermore, it is shown, that range based virtual beacon strategy is useful in combination with CL and SWCL. In all cases a high spatial limit is used.

Due to the obviously strong impact of the number of virtual beacon nodes, same analyses have been performed, taking only results with an upper limit of three or two virtual beacon nodes into account, respectively. In both cases dependent distance approximation outperforms independent distance approximation. Also the internal order of the presented approaches is similar to the one presented in Figure 16. For each upper limit of virtual beacon nodes the best HDLS approach is presented in Figure 17. As it is illustrated, two times AWCL based approaches provide best results, while in the third case WCL performs best. All given HDLS approaches perform better than original sDLS with costly update operations. Even though, using as much virtual beacon nodes as possible results in highest accuracy, high accuracy can be also achieved using two or three virtual beacon nodes.

The achieved improvements in localization accuracy are mainly caused by an increased number of beacon nodes, used for localization. Due to the fact, that different virtual beacons, based on different precalculations, use different sets of beacon nodes, cardinality of resulting unions is commonly higher than cardinality of the individual sets. Since the number of used beacon nodes only depends on the applied virtual beacon strategy, Figure 18 exemplary shows the number of used beacon nodes for the best cases, presented in Figure 17. It is shown, that using two virtual beacon nodes increases the number of beacon nodes used by about 40% compared to sDLS^{ne}. Using three virtual beacon nodes leads to an increase of about 67%, while using as much virtual beacon nodes as possible leads to an increase of 245% beacon nodes.

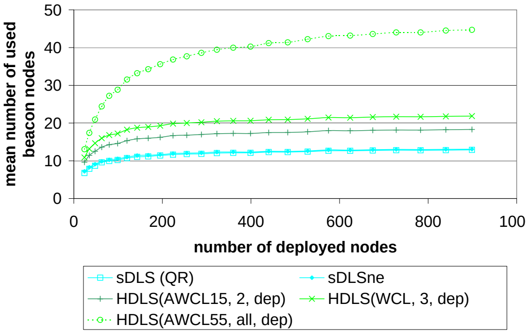


Figure 18. Mean number of used beacon nodes of best performing HDLS approaches for different limits of virtual beacon nodes

As a matter of course, using more beacon nodes, increasing localization accuracy, comes along with increased cost by means of computation. The mean number of operations, performed on each blind node, to perform one localization is given in Figure 19. The number of operations is mainly determined by the number of virtual beacon nodes or the number of individual precalculations, respectively. The most important result is that all presented approaches need less computations than the original sDLS with matrix updates, while all of these approaches, given in Figure 19, provide higher accuracy than sDLS. The additional cost for each additional virtual beacon node is about 80% of the cost of sDLS^{ne}.

VI. Conclusion

In this work, the efficient localization approach sDLS^{ne} has been combined with various coarse grained approache

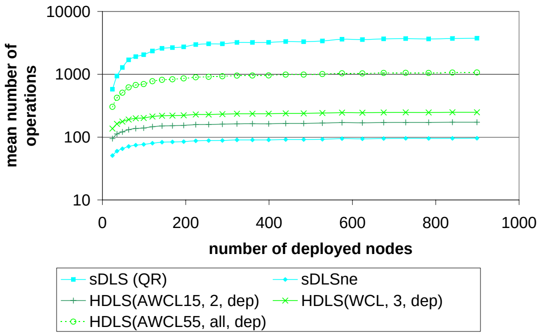


Figure 19. Mean number of operations of best performing HDLS approaches for different limits of virtual beacon nodes

to improve accuracy of localization. As shown in Figures 8, 10, 15, and 17, the new HDLS approach provides higher accuracy than sDLS^{ne} and even outperforms the initial sDLS approach. Using the newly introduced dependent distance approximation, even use of only two virtual beacon nodes, i.e., two precalculations, dramatically increases localization accuracy. Although HDLS needs more computations than sDLS^{ne}, it needs much less computations than sDLS. It further provides the possibility to chose between various variants with different cost. Using a small range, the presented range based virtual beacon strategy provides an very cost efficient way to improve sDLS^{ne}.

ACKNOWLEDGMENT

This work was supported by the German Research Foundation under grant number BI467/17-2 (keyword: Geosens2)

REFERENCES

[1] K. Akkaya and M. F. Younis, “A survey on routing protocols for wireless sensor networks,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[2] J. Al-Karaki and A. Kamal, “Routing techniques in wireless sensor networks: a survey,” *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 6–28, Dec. 2004.

[3] J. Salzmann, R. Behnke, M. Gag, and D. Timmermann, “4-MASCLE - Improved Coverage Aware Clustering with Self Healing Abilities,” *International Symposium on Multidisciplinary Autonomous Networks and Systems (MANS 2009)*, pp. 537–543, Jul. 2009.

[4] J. D. Gibson, *The Mobile Communications Handbook*. Boca Raton FL, USA: CRC Press, 1996.

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

[6] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, “Weighted Centroid Localization in Zigbee-based Sensor Networks,” in *IEEE International Symposium on Intelligent Signal Processing, WISP 2007*, Madrid, Oct. 2007.

[7] R. Behnke and D. Timmermann, “AWCL: Adaptive Weighted Centroid Localization as an efficient Improvement of Coarse Grained Localization,” *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, pp. 243–250, Mar. 2008.

[8] F. Reichenbach, A. Born, D. Timmermann, and R. Bill, “A distributed linear least squares method for precise localization with low complexity in wireless sensor networks,” *Distributed Computing in Sensor Systems*, pp. 514–528, 2006.

[9] R. Behnke, J. Salzmann, D. Lieckfeldt, and D. Timmermann, “sDLS - Distributed Least Squares Localization for Large Wireless Sensor Networks,” *International Workshop on Sensing and Acting in Ubiquitous Environments*, Oct. 2009.

[10] R. Behnke, J. Salzmann, and D. Timmermann, “sDLS^{ne} - Improved Scalable Distributed Least Squares Localization with minimized Communication,” *21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2010)*, Sep. 2010.

[11] W. S. Murphy and W. Hereman, “Determination of a position in three dimensions using trilateration and approximate distances,” Tech. Rep., 1999.

[12] Y. Zhang, M. Fromherz, and L. Kuhn, “Rmase: Routing modeling application simulation environment,” 2009, http://www2.parc.com/isl/groups/era/nest/Rmase/.

Tree-Based Organization for Very Large Scale Sensor Networks

Moran Feldman
Faculty of C.S. Technion
Israel Institute of Technology
Haifa, Israel
moranfe@technion.ac.il

Sharoni Feldman
ATC Consulting
Haifa, Israel
feldsh@netvision.net.il

***Abstract*— One of the major challenges in deploying large scale sensor networks is the ability of the sensors to weave dynamically and autonomously into a sensing plan. In this paper we present a novel algorithm with certain characteristics. It is applicable for thousands of sensors and uses a tree based organization to present an aggregation method that aggregates discrete events into compound ones. It presents a set of security levels to ensure that events are transmitted to the control center (sink) and also presents backup layers to ensure maximal connectivity. Furthermore, it is applicable for sensors with no GPS. The algorithm was successfully tested using the dedicated simulator on a terrain containing 10,000 sensors. Our results show that the sensors perform the process of weaving into a sensing plan, the task of identifying multiple intruders, reporting the events to the sink in a short time and comply with the other demands.**

Keywords- Large Scales Sensor network; Routing; Localization; Data aggregation.

I. INTRODUCTION

The evolution of microelectronics and communication technologies facilitates the manufacturing of miniature sensors comprising a small transmitter/receiver, a processor, memory components and a low-power battery [1], [2]. Most often, the sensor, or node, is a Boolean sensing device that detects an event within a given sensing range, and is able to inter-communicate using wireless protocols with adjacent nodes, creating a wireless sensor network (WSN).

There are a few common methods to categorize the network organization in a WSN according to the network structure [3]. The first category is flat routing, wherein all nodes have an identical role. The routing of events from the sensing nodes to the sinks can use any node in the network without any limitations and without interacting with any centric nodes [4][5][6][7][8][9]. In hierarchical routing protocols, part of the sensors possesses additional tasks. In this case, the sensors are grouped into clusters. One of the sensors in every cluster is designated as the cluster-head. If needed, it is possible to group cluster heads into new clusters [10][11] [12][13]. This organization method delegates routing responsibilities to the cluster heads rather than the regular nodes. Hierarchical routing is considered as an efficient way to reduce energy consumption by transferring the aggregation process to the cluster head. A third method is location-based routing. The sensors are aware of their position in the theater. This location information can be exploited in order to route data over the network more efficiently. The sensor location can be obtained from a GPS receiver installed in every sensor [14]. Another method uses relative coordinates that are based on information gathered from neighboring sensors. The distance between neighboring sensors can be estimated according to the strength of the incoming signals [15][16][17]. An interesting approach is presented in [18]. The traffic is divided into two types: high priority traffic and low-priority traffic. High priority data is routed using a dedicated congestion zone arranged as a spanning tree with the sink as a root and the low priority traffic is routed via other nodes and longer paths. This model enables every node to be connected to several trees according to the number of sinks.

In this paper, we present the Very Large Scale Sensor Network Algorithm (VLSSNA). The VLSSNA presents a novel routing technique for a very large network composed of 10,000 sensors. This number is significantly higher than the number presented in literature [19]. This algorithm creates a flat network. Moreover, the algorithm enables the network operators to assign an "importance level" to events and to transmit important events on parallel layers of the network. This feature increases significantly the probability that events will not get lost even if some intermediate nodes malfunction.

The sensors dissemination process is totally random without any prior definitions in the sensors. The organization of the network is an autonomous procedure without any external intervention. In our research, we assume a sensor density that prevents an object from crossing the sensing field without being detected [20]. Energy economization is achieved using a communication method that combines broadcasting and sensor-to-sensor communication and an aggregation process used to minimize the number of messages.

The reminder of this paper is organized as follows. Section 2 presents the theater and the network elements, Section 3 presents the sensors control and management, Section 4 deals with energy saving methods, Section 5 presents the simulator used for evaluation of the sensor network and Section 6 presents the simulations and results. Conclusions follow in Section 7.

ISBN: 978-1-61208-144-1

II. Theatre and Network Elements

A. The Theater

Fig. 1 presents a schematic view of a typical theater on which the sensors are dispersed. The sensors are dispersed so that their density ensures that every target will be detected by one or more sensors. At the edge of the theater, three base stations (BSs) are placed, arranged in an equilateral triangle. The BSs are connected by a high-speed communication link. At least one sink controls and monitors the events.

B. Network Elements

a)Base Stations

The BSs are identical, and they are controlled by the sink. Every BS is required to know its exact position in space (x, y, z) coordinates. This information can be obtained manually when the BS is installed or via a GPS. In addition, all BSs clocks are synchronized.

A BS is constructed of the following units: (a) A long-range downstream transmitter that covers the whole theatre. (b) A short-range downstream transmitter and upstream receiver that cover the adjacent sensors. (c) A high speed LAN that connects all BSs and the sink. This LAN is used to transfer synchronization data among the BSs and the sink, alarms received from the sensors to the sinks and messages from the sink to the BSs and the sensors.

b) Sensors

The sensors are scattered in the terrain. The distribution of the sensors in the theater is not required to be uniform. However, it is assumed that some connectivity between a sensor and its surrounding sensors can always be found. A sensor has a limited life expectancy, and as a dispensable device the whole network will continue to function with a certain decrease in the number of active sensors. The design of the network allows the operators to add periodically new sensors to the field to overcome the natural decrease in the number of sensors. A sensor is composed of the following major units: (a) A binary Omni sensing device like a microphone. The sensor is unable to detect the direction of the event. (b) Short range Transmitter/Receiver. We can assume that the transmission range is greater than the sensing range. Fig. 1 presents a sensors field. The dashed lines present the sensing range while the full lines present the transmission range. A sensor can receive messages transmitted from other sensors within their transmission range or from the BSs. (c) Processor and Battery.

III. Sensors Control and Management

A. First Network Activation

The network activation process starts after completing scattering the sensors in the field. The sensors in the field will not start to work until the activation process ends. Note that all sensors clocks will be synchronized during the activation process. The activation process has 3 steps and is coordinated by the BSs:

1. BS_1 sends a beacon that covers the whole field. This beacon carries the following data elements: (a) Time stamp used to synchronize the internal clock of every sensor and the other BSs. (b) The name and geographical location (XBS_1, YBS_1, ZBS_1) of BS_1 (c) Wait time (t_w) in milliseconds.
   Every sensor receiving this beacon will update its internal clock with the BS_1 time and its internal database with the coordinates of BS_1. The other BSs will update their internal clock according to BS_1 clock.

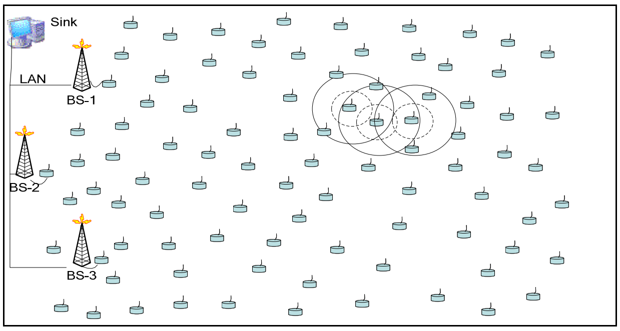


Figure 1: A typical sensor theater. Three base stations organized as an equilateral triangle and a single sink monitor the wireless sensor netwrok.

2. BS_2 waits predefined t_w milliseconds (a) value known to all network elements) before broadcasting a beacon. This beacon which covers the whole field carries the following parameters: (a) The name and geographical location (XBS_2, YBS_2, ZBS_2) of BS_2. (b)Time stamp.
   Every sensor receiving this beacon will update its internal database with the coordinates of BS_2.
3. BS_3 will wait t_w milliseconds after BS_2 transmission before broadcasting a beacon. This beacon parameters and the process are identical to these of BS_2.
   Every sensor receiving this beacon will update its internal storage with the coordinates of BS_3.

After receiving the 3 beacons, every sensor starts its localization algorithm (based on trilateration). The termination phase of this algorithm is a set of sensors spread in the field when every sensor has identified the BSs, the BSs geographical locations and its own location. In the current state, the sensors stay dormant and do not initiate any activity.

The network activation process runs periodically and is used to join new sensors that were added to the field or remap old sensors that were moved inside the field.

B. Messages and Data transfer

The communication among the network elements is performed by messages. The network elements exchange broadcast messages addressed by all listening nodes within the transmission range and directed messages that address a specific node within the transmission range. Another type of classification is based on the transmission range of the message originator. A BS can transmit short range and long range messages while a sensor is capable to transmit only short range messages. Tab. 1 summarizes combinations between the message types and message transmission range. While a BS can send directly a long distance message to every node in the terrain, a node which is required to send an event to the BS, is required to use intermediate nodes to bridge the distance. The process of transferring the information from the node to the BS is based on a “store and forward” mechanism. A node that received a message will forward the message to the next leg in the chain only after it was received completely.

Special attention was given to energy saving. As will be described in details later in this paper we implemented “energy saving” methods in critical and demanding procedures.

C. Trees formation processes

During the trees formation phase, every sensor builds its connections in the field. All BSs initiate stimulatingly the trees formation process although the processes are independent. It starts after the localization process and terminates when the sensors complete joining the spanning trees. After completion, every node is connected to *n* trees when *n* is the number of BSs.

Fig. 2 presents 2 trees in the theater. The tree of the root node BS_1 and the nodes addresses is presented in continuous lines in black and the tree of root node BS_3 is presented in dashed lines. The addresses of the nodes that belong to root BS_1 start with <1.> and the addresses of nodes that belong to BS_3 start with <3.>. We skip the tree formation process due to lack of space.

IV. Detailed Description – TFA algorithm

The trees formation algorithm (TFA) organizes the nodes in the field in rooted trees. Only nodes that belong to the same tree can transfer events to the BS which is also the tree root and the sink. To ensure the maximal connectivity, all nodes will try to organize themselves in a single tree. Every node in the field has a unique and fixed node-id and a virtual coordinate (of the type x.y.z..) that may change depending on the changes in the tree structure. Every tree is identified by a “tree name” which is the *id* of the root node, which is the BS. The metrical join (referred also as join) protocol should satisfy the following properties:

Table 1: Messages Definition and Types of Messages

MSG Range Type	Long Range Message	Short Range Message
Directed Message	A message sent by a BS, directed to a specific sensor in the theater. The addressed sensor is identified by unique sensor-id within the terrain.	A message sent by sensors or the short-range transmitter of the BS. This message is addressed a specific sensor identified by unique sensor-id within the transmission range.
Broadcast Message	A message transmitted only by a BS. This message is targeted to all sensors in the terrain	A message sent by sensors or the short-range transmitter of the BS. All receiving nodes within the short transmission range address this message.

Eventually all nodes within transmission area must fuse into a single tree. When two trees are being fused, most updates should be made to the nodes of the smaller tree (in the number of nodes). The protocol should maximize the number of nodes that joins the tree in every step (yielding a parallel fuse). Nodes periodically attempt to balance the tree by shortening their distance to the root of the tree by improving their position in the tree and joining higher-level nodes.

The protocol is fully distributive with no “central" control.

Running alarms will not be affected (i.e. will not break) in any way from the ongoing trees join process or rearrangement of an existing tress. The parallel process of creating the trees (with the roots BS_1, BS_2…) is fully independent. When all processes terminate, every node is a member in all trees.

The TFA algorithm runs when the stability of the network is violated. Every time this algorithm runs, it will bridge the “holes” created by faulty sensors and will add new sensors that were added to the terrain.

A. Events, Events aggregation and Delivery Verification

In our model, every sensor has a circular events detection area around itself. An intruder that enters the sensors field will stimulate every node whenever it enters its detection radius.

Every node that detects an intruder sends an event message to its father. Fig. 3 presents a section of the sensors field and the intruder path within it. The circle around each sensor presents the detection zone of the sensor. Our assumption is that every sensor is able to perform basic filtering of the detected noise, i.e. a sensor programmed to detect a noise of a car will not respond to a noise created by a walking animal or barking of a dog. In Fig. 3 the intruder triggers 4 sensors located on his path.

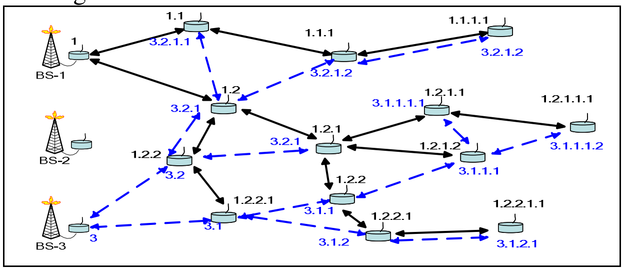


Figure 2: Trees in the terrain

The tree structure creates a natural organization of the nodes, which allows the network to “aggregate” events from each subtree in the subtree root and minimize the amount of events transferred towards the sink. For example, node <1.1> is able to aggregate the events detected by its children <1.1.X> and send it as a unified event to its father. A managing automaton runs in every node and enables the node to act in one of two possible ways: Detect an event or receive and handle event messages from its children and transfer them either transparently or with some updates toward the sink.

The automaton is composed of 3 states.

1) Idle. This is the stable state of the node. In this state, the node is waiting for an event created by the intruder or to a message from one of its children. When the node detects an intruder, it sends the message “event-report-message” with the event details to its father. The main event details are the geographical location of the node and the event timestamp. After detecting an event, the node will not report any additional event for a short time span. This prevents the node from creating a flood of messages toward the sink caused by a single short timed stimulation.

2) Store information – open timer. As soon as the node receives an “event-report-message” from one of its children, it assumes that his other children may detect this same event. The node stores the event data and waits for additional messages from the other children. A short wait timer is activated in order to limit the wait time. If the node receives an “event-report-message” from all its children, the timer is redundant, and it is cancelled.

3) Calculate direction. This state is activated once the node decides to stop waiting for additional “event-report-messages”. The node tries to calculate the local direction of the intruder, based on the messages it received. The results are then sent towards the root using the message “aggregated-event-message”. It is possible to calculate the direction only if two or more children reported the event. In case that only a single child detected the event, the aggregated message will carry the location of the detecting node. In the case that 2 or more nodes have detected the event, the message will carry the locations of the two most distant detecting nodes.

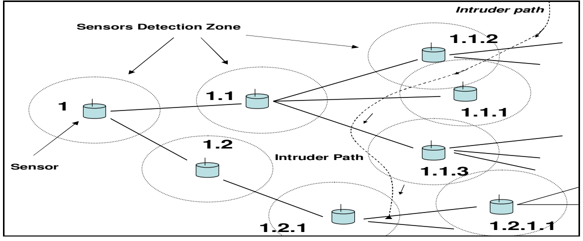


Figure 3: Sample Intruder Path

When a node receives a message “aggregate-event-message” from its subtree, it acts as a router and sends it to its father.

B. Data Aggregation

The purpose of the aggregation process is to reduce the amount of data transferred from the detecting nodes toward the root. However, as will be explained later, this process slows the speed in which events are transferred toward the sink. The aggregation process can be nested, for example, node <1> in Fig. 3 can aggregate the data from its children <1.1> and <1.2> into a single message. A *k-level* aggregation process performs the aggregation process in the *k* lowest levels of the tree (starting from the leaves). The aggregation level can be adjusted according to the required performance of the network and the type of expected intruders

The need to aggregate messages requires that inbound messages will be delayed in the node for a period of time. This delay enables other inbound event messages to arrive during the delay and to be aggregated into a single aggregated message.

C. Event Delivery Verification

The verification feature enables the network to ensure with high level of certainty that a reported event has been transferred successfully from the sensing nodes via the routing nodes to the BSs and the sink. The network architecture enables the use of two types of verification levels. The basic method is "Report & Forget" which does not require a BS to acknowledge the acceptance of the event report. The enhanced method "Report & Acknowledge" requires the receiving BS to send an acknowledgement to the reporting nodes. The acknowledgement does not use the sensors network to transfer the acknowledgement to the sender but broadcasts a "Shout" message over the air from the BS with the sensor-id and event details. In case that the event originator has not received the acknowledgment message within a predefined period, it will resend the event report.

Another capability of the architecture is to use two levels of energy saving delivery methods. The economized method uses a "*Load-Sharing*" mechanism where the reporting node selects cyclically one of the trees and sends the event report over the tree to a single BS. The "*Active-All*" mechanism is more energy consuming. Using this method, the node sends in parallel the event report over all possible trees by replicating the event report over all trees. The replicated events are propagated to the BSs and the sink. According to the events time and locations, the sink will fuse the replicated events into one single event.

V. ENERGY SAVING METHODS

The energy resources of the nodes are very limited. A critical factor in the design of the VLSSNA algorithm was to reduce the energy utilization as much as possible and to enable the network operators to select an operation mode that fits the needs and will meet the energy resources of the nodes. The following methods are used by the network. *(a)* In case that a node is able to control its transmission power, and it has more than a single candidate to become its father, it will select the most “economical” father that is not too close. *(b)* Optional Acknowledge (Ack) via “shouting” and not via the network. *(c)*Using adjustable replicated transmissions. A node can generate an Ack message that will be transmitted in parallel on one or more trees according to its importance. *(d)* The ability of the initiating sensor to select randomly a tree contributes also to an even utilization of energy in the network.

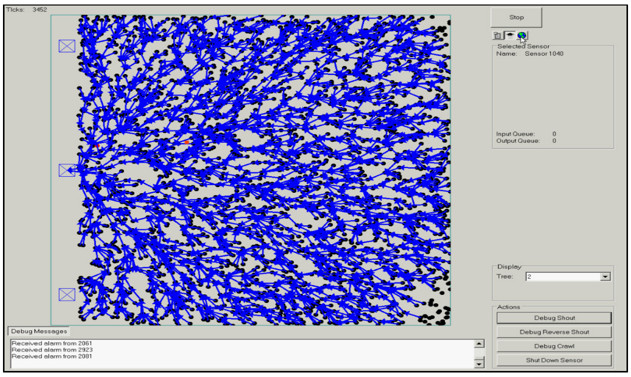
VI. THE INTERACTIVE FLEXIBLE AD HOC SIMULATOR

For testing and evaluating the protocols described in this research (VLSSNA and TFA) we used the interactive Flexible Ad-Hoc Simulator (IFAS) [7]. The IFAS simulator was originally developed for evaluating the performance of ad-hoc protocols; it was adapted for sensor networks and can handle successfully thousands of sensors. In this section, we shall describe the simulator and the simulation scenarios.

Special attention was given to the following aspects: (a) enhanced visualization tools that give a full visual of the theater, zooming of selected zones, node movements and voice channels in ad hoc networks, and specific node status including queue status; (b) tracing the formation of trees; (c) tracing the events transfer and sessions in real time; (d) configuration and simulation definition via online screens; (e) definition and tracking of intruders paths and pace; (f) support of logging, debugging and analysis tools.

The enhanced visualization capabilities, unique to this simulator, contributed to the understanding of the protocols behavior, as we were able to view the progress in the field and detect unexpected behavior.

The simulator enables the user to get detailed online reports. These capabilities set afloat disruptions in specific nodes behavior as a result of their location in the field. It is possible to "kill" nodes, zoom in and out selected areas and trace explicitly certain events. Fig. 4 presents one of the trees on a terrain with 3000 sensors (the black points) placed randomly in the field.



Ticks: 3452

Selected Sensor
Name: Sensor 1040

Input Queue: 0
Output Queue: 0

Display
Tree: 2

Debug Messages

```
Received alarm from 2061
Received alarm from 2923
Received alarm from 2081
```

Figure 4: Sample Tree with BS-2 as a root

The root tree is BS_2. The intruders which are not seen in this view are crossing the field and activating the sensors. In the bottom of the screen we see the events received by the BS.

Table 2: Quantities details

Parameter	Value
Number of sensors	3000 sensors (except for the scalability tests)
Field size	1Km x 1Km
Sensor transmission range	60 meters
Event detection range	10 meters

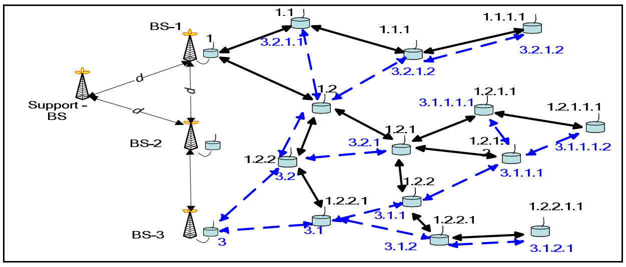


Figure 5: Base stations Organization

VII. SIMULATIONS AND RESULTS

The simulation environment creates the infrastructure to analyze the following directions: (a) the efficiency and scalability of the TFA algorithm. (b) The performance of the network in the following aspects: The contribution of the aggregation process to reduce the number of messages and the delay created by nodes as a result of the aggregation process. The tests were performed using the parameters presented in Tab. 2.

Fig. 5 presents the BSs organization. BS_1, BS_2 and BS_3 act as the trees roots and are able to broadcast "shout" messages that cover the whole terrain. In addition, the events are received via these BSs.

The localization algorithm requires using a minimum of three synchronized beacons that broadcast periodically a time event. The best organization for the beacons is in an equilateral triangle. To show an applicative possibility, we combined the functionality of two beacons with BS_1 and BS_2 and created a Support Basestation (SBS) that participates only on the localization process. The distance between BSs, BS_1 and BS_2 is *d*. Note that it is possible to use the SBS as a replacement for BS_3.

<u>Scalability and connectivity</u>

The basic requirement from the TFA algorithm is to connect all sensors in the field into one single network. This set of tests is comprised of two groups:

1) Scalability tests. We run the TFA algorithm on a field with a minimal population of 1000 sensors and a maximal population of 10,000 nodes. The trees creation process succeeded to fuse all sensors into a single tree without any measurable impact on the performance.

2) Connectivity tests. The connectivity tests included two groups of tests – the first group verified that all nodes dispersed in the tree are merged into a single tree. The second group of tests checked what happens if sub-tree nodes within the tree die. In this case, the tests show that the nods that belong to the subtree of the faulty node discover the fault, and declare themselves as standalone trees. This declaration initiated the fusion process that results in a new fully connected field.

<u>Average nodal delay</u>

Intuitively, two factors contribute to the efficiency of the aggregation process – the nodal delay time and the number

ISBN: 978-1-61208-144-1

of the tree levels that participate in the aggregation process. Fig. 6 presents the impact of the nodal delay time on the aggregation process. In this test, we measured the percentage of aggregated messages out of all event messages triggered by the intruder as a function of the nodal delay time. In this test, every inbound message received by a sensor on its way to the sink is delayed for a fixed period before it is outbounded to the next tree level. We expect the number of aggregated messages to grow as the internal delay grows. As presented, the number of aggregated messages grows significantly to 33% when the delay grows to 700ms. A small increase to 40% is achieved when the delay grows to 1200ms. Additional delay time greater than 1200ms does not contribute to the performance. Note that the increase in the performance costs a significant delay in the arrival of the alarm message to the sink.

VIII. CONCLUSIONS

The tree based connection between sensors presents a very efficient and practical way to connect between very large numbers of sensors in a sensor network. The method presented in this paper depicts also a replication tree mechanism that increases the redundancy of the network and ensures a very high level of connectivity.

The aggregation algorithm is targeted to reduce the number of events that are transferred from the network to the sinks. The main purpose of this algorithm is to save transmission energy. A major consideration in the decision of the quality of this algorithm is the usage of extra energy required in implementing more sophisticated algorithms, the extra requirements from the processing unit and the memory size of the sensor. In addition, transmitting more data that is required to improve the algorithm increases significantly the energy consumption. The savings should be balanced against the cost of transferring all raw events to the sinks.

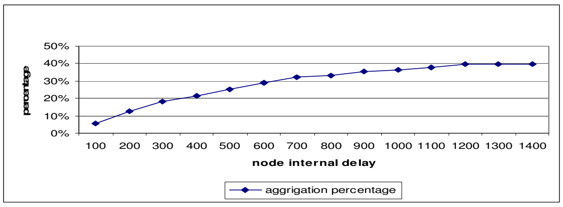


Figure 6: Average Aggregation percentage Vs. Nodal Delay

REFERENCES

[1] G. Pottie and W. Kaiser, "Wireless integrated network sensors," Communications of the ACM, vol. 43, May 2000, pp. 51-58.

[2] J. Kahn, R. Katz, and K. Pister, "Mobile networking for smart dust," In preceedings of The Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), 1999, pp. 271-278.

[3] J. N. Al-Karaki and A. E. Kamak, "Routing Techniques in Wireless Sensor Networks: A Survey," IEEE Wireless Communications Vol. 11 Issue 6, 2004, pp. 6-28.

[4] Y. Ben-Asher, M. Feldman, S. Feldman, and P. Gurfil. IFAS: Interactive flexible ad hoc simulator. Simulation Modeling Practice and Theory, 15(7), 2007, pp. 817–830.

[5] Y. Fan, A. Chen, L. Songwu, and Z. Lixia, "A scalable solution to minimum cost forwarding in large sensor networks". Proceedings of Tenth International Conference on Computer Communications and Networks, 2001, pp. 304-309.

[6] C. Intanagonwiwat, R Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks", Proceedings of the ACM MobiCom, Boston, Ma, 2000, pp. 56-67.

[7] J. Kulik, W. R. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks", ACM/IEEE Int. Conf. on Mobile Computing and Networking, Seattle, WA, Aug. 1999, pp. 174-175.

[8] D. Braginsky and D. Estrin", "Rumor Routing Algorithm for Sensor Networks", in proceedings of the First Workshop on Sensor and Applications (WSNA), Atlanta, GA, Oct. 2002, pp. 1-12.

[9] J. Kulik, W. Rabiner, and Hari Balakrishnan", "Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks", Wireless Networks, vol. 8, Num. 2-3, 2002, pp. 169-185.

[10] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy Efficient Communication Protocol for Wireless Mi-crosensor Networks," Proceedings of the Hawaii International Conference on System Sciences (HICSS '00), Jan. 2000, pp 323-327.

[11] L. Subramanian and R. H. Katz, "An Architecture for Building Self Configurable Systems", in the Proceedings of the IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, Boston, MA, August 2000, pp. 63-73.

[12] Q. Fang, F. Zhao, and L. Guibas, "Lightweight Sensing and Communication Protocols for Target Enumeration and Aggregation", Proceedings of the ACM international symposium on Mobile ad hoc networking and computing (MobiHoc), 2003, pp. 165-176.

[13] F. Ye, H. Luo, J. Cheng, and S. Lu, L. Zhang, "A Two-tier data dissemination model for large-scale wireless sensor networks", proceedings of the annual ACM/IEEE MobiCom, 2002, pp 148-159.

[14] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad-hoc Routing," In Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking, 2001, pp. 70-84.

[15] A. Savvides, C-C Han, and M. Srivastava, "Dynamic fine-grained localization in Ad-Hoc networks of sensors," Proceedings of the Seventh ACM Annual International Conference on Mobile Computing and Networking (MobiCom), July 2001, pp. 166-179.

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

[17] S. Capkun, M. Hamdi, and J. Hubaux,"GPS-free positioning in mobile ad-hoc networks", Proceedings of the 34th Annual Hawaii International Conference on System Sciences, 2001 pp. 3481-3490.

[18] R. Kumar, R. Hosam, and C. Guohong, A. Farooq, Y. Aylin and T. La Porta, "Conestion Aware Routing in Sensor Networks", Technical Report, http://nsrc.cse.psu.edu/tech_report/NAS-TR-0036-2006.pdf, 2006.

[19] F. Y. Alvin Chen, S. Lu, and L. Zhang, "A scalable Solution to Minimum, Cost Forwarding in Large Sensor Networks," in Proceedings of the 10th international conference on Computer Communications and Networks, , 2001, pp 304-309.

[20] B. Liu and D. Towsley, " A Study of the Coverage of Large-scale Sensor Networks," in IEEE International conference on Mobile Ad-Hoc and Sensor Systems, , 2004, pp. 475-483.

HyPAERLoc: Plausible Hybrid Localization for Wireless Sensor Networks

Jakob Salzmann, Ralf Behnke, Philipp Gorski, Dirk Timmermann
Institute of Applied Microelectronics and Computer Engineering
University of Rostock
Rostock, Germany
{jakob.salzmann, ralf.behnke, philipp.gorski, dirk.timmermann}@uni-rostock.de

***Abstract*— Position estimation is one of the major challenges of sensor nodes in wireless sensor networks. By utilizing the information of messages of some pre-deployed location aware nodes, called beacons, and signal strength based distance estimation, a location unaware node is able to estimate its position. In the recent years, several localization methodologies have been developed. Due to imprecise distance estimations via received signal strength utilization, the localization accuracies of these algorithms differ, depending on the scenario conditions. In the proposed work, we developed an algorithmic approach to improve the localization accuracy of a least squares approach via two strategies. On the one hand, we tackle the imprecise distance measurements with a preceding plausibility check. On the other hand, we evaluate the achieved accuracy and improve the result by weaving the result of adaptive weighted centroid localization into a hybrid localization. The achieved localization accuracy beats the performance of the preceding approaches in all investigated scenarios, particularly for low beacon densities.**

Keywords- Wireless Sensor Networks, Localization, Log-normal Fading.

I. Introduction

The ongoing miniaturization of technical devices allows to combine microcontroller, sensors and radio technology to a single tiny and battery driven device, called sensor node. Due to their radio technology, numbers of the nodes can compose themselves together to a wireless sensor network (WSN), which ranges from few nodes in one-hop distance to each other up to large networks with hundreds of nodes and multi-hop dimension. Such WSNs can be deployed in an area to observe, using their sensor abilities to detect phenomena in various scenarios. Examples for the operation of sensor networks are disaster control, environmental observation, tracking of moving objects [1]. In most of the scenarios, e.g. object tracking, the location of a detected phenomenon is as important as the properties of the phenomenon itself. As result, the detecting nodes have to be aware of their position within a reference system. Due to the WSN operation in inaccessible or indoor scenarios and the large amount of nodes, neither a specific deployment of all nodes nor a global localization system, e.g. GPS, are always feasible for the localization of sensor nodes. A feasible solution is that a fraction of nodes is location aware and used as reference nodes, called beacons [2-6]. These beacon nodes get their position either due to a specific deployment, e.g. deployed by a robot or are additionally equipped with a GPS-receiver. All remaining nodes, called unknowns may be deployed randomly in the monitored area and estimate their position with the help of the beacon nodes. Each beacon broadcasts a message, and each receiving node stores the contained information about the beacons' position as well as the received signal strength or the derived distance.

The algorithms, which utilize the collected information from all nearby beacon nodes to estimate the position of an unknown node, differ in complexity and achieved estimation accuracy. Interestingly, a high complexity is not mandatory for a competitive accuracy, especially if the distance estimations are inaccurate due to shadowing effects.

The contribution of the paper improves the localization accuracy in two ways. On the one hand, a plausibility check avoids applying impossible distance estimations, on the other hand the uncorrelated localization accuracy of a different algorithm family is utilized. The result is described as an algorithm called HyPAERLoc (Hybrid Plausible Approach for Error Reduced Localization). We compared HyPAERLoc with its preceding localization approaches in different scenarios with a log-normal fading radio channel model. With HyPAERLoc, the localization accuracy increases significantly compared to the preceding algorithms. Additionally, the algorithm avoids outliers.

The remainder of the paper is structured as follows: Section II describes the related work, Section III explains our simulation environment. Section IV analyzes the strengths and weaknesses of the researched localization approaches. In Section V, we describe and evaluate our check for implausible distances, Section VI describes and evaluates the hybrid localization approach. Section VII gives the conclusion and an outlook.

II. Related Work

Beacon based localization algorithms can generally be divided into coarse-grained and fine-grained localization. The actual section describes common representatives of both groups of algorithms.

1. *Coarse-Grained Localization*

Coarse-grained localization algorithms represent heuristic methodologies to estimate the position of an unknown node. These classes of algorithms are characterized by the disadvantage, that even with exact distance measurements, they are not able to estimate the exact position of the unknown due to their simplification. In the following, three coarse-grained localization algorithms are described and evaluated.

Centroid Localization

By the idea of Centroid Localization (CL), an unknown node is located at the centroid of all received beacons [2]. If *$P_i(x,y)$* represents the position of an unknown node *i*, *n* represents the number of received beacons and *$B_j(x,y)$* represents the known position of a beacon node *j* in range, each unknown node *i* can perform the algorithm as given in (1).

$$P_i(x,y) = \frac{1}{n}\sum_{j=1}^{n} B_j(x,y) \qquad (1)$$

It can be seen, that the algorithm is easily to perform and abstain from distance estimations.

Weighted Centroid Localization

An advanced approach is Weighted Centroid Localization (WCL) [3]. In contrast to CL, WCL uses additional information to calculate the centroid. Usually, a sensor node has the ability to measure the received signal strength of a message. The result of this measurement can be utilized to estimate the distance to the beacon node which sent this message. The estimated distance can be utilized to improve the position estimation by weighting the centroid calculation, as done by WCL. A practical approach is to describe the impact of a beacon B_j with the weight w_{ij} as reciprocal of the estimated distance d_{ij} between unknown *i* and beacon *j*, as done in equation (2).

$$w_{ij} = (d_{ij})^{-1} \qquad (2)$$

With this weight, the calculation of the centroid changes as given in (3).

$$\mathrm{P_i(x,y)} = \frac{\sum_{j=1}^{n}\left(w_{ij} * B_j(x,y)\right)}{\sum_{\mathrm{j=1}}^{\mathrm{n}} \mathrm{w_{ij}}} \qquad (3)$$

As result, the position of the unknown node is not estimated as real centroid of all surrounding beacons, but nearer beacons pull the position of the unknown in their direction.

Adaptive Weighted Centroid Localization

The goal of Adaptive Weighted Centroid Localization (AWCL) was the improvement of the weighting [4]. If all distances between one unknown and its received beacons are similar to each other, the effect of the weighting becomes relatively low and the result is comparable to CL. To overcome this problem, AWCL adapts the resulting weights by applying the following steps:

1.) Determination of the smallest weight $w_{i,min}$
2.) Calculation of the reduction part *q*. Simulation results in [4] provide to calculate a feasible reduction part as done in (4).

$$q = w_{i,min} * 0.55 \qquad (4)$$

3.) Applying the centroid estimation with reduced weighting, as done in (5).

$$\mathrm{P_i(x,y)} = \frac{\sum_{j=1}^{n}\left(\left(w_{ij} - q\right) * B_j(x,y)\right)}{\sum_{\mathrm{j=1}}^{\mathrm{n}}\left(w_{ij} - q\right)} \qquad (5)$$

As result, AWCL is able to apply a more influential weighting, which directly impact the localization accuracy in most situations.

2. Fine-Grained Localization

The idea of fine-grained localization approaches is to determine the exact position of an unknown node. The obvious disadvantage of the fine-grained localization approaches is the costly computational effort. In the following, the method of least squares as most common representative and one successor are explained.

Linear Least Squares

The method of linear least squares, also called atomic multilateration in [5], is an approach to approximate the solution of a linear over determined equation system. In least squares, the sum of the squares of the residua of the solved equation system is minimized. Applied to the localization problem, each unknown node *i* has to determine two unknown variables, x_i and y_i in a 2-dimensional sensor network. Additionally, each received beacon *j* allows setting up an equation as given in (6).

$$\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 = d_{ij}^2 \qquad (6)$$

This equation can be transformed as given in (7).

$$x_i^2 + y_i^2 + x_j^2 + y_j^2 - 2(x_i x_j + y_i y_j) = d_{ij}^2 \qquad (7)$$

Here, d_{ij} is the known distance between unknown node *i* and beacon *j*, x_i and y_i are the unknown coordinates of the unknown node and x_j and y_j are the known coordinates of the beacon in a common reference system. For applying linear least squares, the quadratic terms of the unknown have to be removed. This can be done by subtraction of the equation of beacon *k*, as done in (8).

$$x_j^2 + y_j^2 - x_k^2 - y_k^2 - 2\left(x_i x_j + y_i y_j - x_i x_k - y_i y_k\right) = d_{ij}^2 - d_{ik}^2 \qquad (8)$$

This linearization can be done for each couple *l* of received beacons. For *n* received beacons, the maximum number of different linear equations *m* is given by equation (9).

$$m = 0.5\left((n-1) * n\right) \qquad (9)$$

After that, the present equation can be transformed into a linear equation $h_i = g_i x_i + y_i$. Here, g_i and h_i are absolute terms and represents single points for the unknown node *i* in the

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

final equation system, in which a straight line with offset y_i and slope x_i can be found.

For each couple of beacons l, values for g and h are calculated as given in (10) and (11) for the example beacons j and k.

$$h_{i,l} = \frac{(d_{ij}^2 - d_{ik}^2 - x_j^2 - y_j^2 + x_k^2 + y_k^2)}{2 * (y_k - y_j)} \tag{10}$$

$$g_{i,l} = \frac{(x_k - x_j)}{(y_k - y_j)} \tag{11}$$

After transforming the equations for each couple of beacons into a system of linear equations, the values for x_i and y_i can be computed as given in equation (12) and (13), whereby $\overline{g_\iota}$ and $\overline{h_\iota}$ are the average values of all available terms of g_i and h_i.

$$x_i = \frac{\sum_{l=1}^{m} (g_{i,l} - \overline{g_\iota})(h_{i,l} - \overline{h_\iota})}{\sum_{l=1}^{m} (g_{i,l} - \overline{g_\iota})^2} \tag{12}$$

$$y_i = \overline{h_\iota} - x_i * \overline{g_\iota} \tag{13}$$

As result, linear least squares is able to calculate the exact position of an unknown node, if the distances between the unknown node and the beacons are correctly estimated. In this case, usually only three beacon nodes and hence two independent couples of beacons are sufficient to determine the exact position.

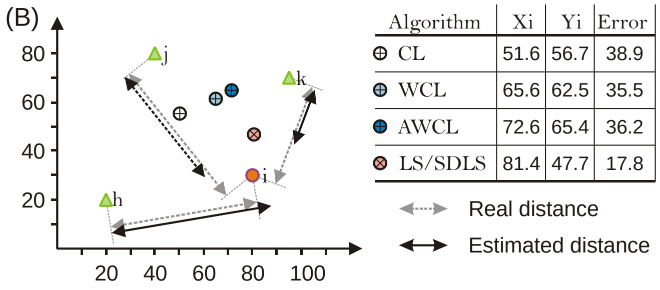
Scalable Distributed Least Squares

Linear least squares have two major drawbacks. The first one is the cost intensive computation of the final position, the second one is the inaccuracy, if the measured distances are inexact. Both drawbacks were tackled with Scalable Distributed Least squares (SDLS) in [6]. While the splitting of the calculation and the involved challenges are not in the focus of this paper, the improvement of the accuracy is significant. Instead of choosing all available or a subset of couples of beacons for calculating $g_{i,l}$ and $h_{i,l}$, an unknown node i in SDLS selects the nearest beacon node as linearizer and subtract the equation of the linearizer from each other node. Hence, the number m of resulting beacon couples for n beacons is given by m=n-1, and each couple contains a part of the linearizer. The reason for selecting the closest beacon as linearizer is given by the fact, that smaller distances could be measured more exactly due to the greater absolute difference in the signal strength.

(A)

Node	X	Y	Real distance to unknown node i	Estimated distance to unknown node i
Unknown node ● i	80	30	0	0
Beacon node ▲ h	20	20	60.8	67
Beacon node ▲ j	40	80	64	52
Beacon node ▲ k	95	70	42.7	25

(B)



Algorithm	Xi	Yi	Error
⊕ CL	51.6	56.7	38.9
⊕ WCL	65.6	62.5	35.5
⊕ AWCL	72.6	65.4	36.2
⊗ LS/SDLS	81.4	47.7	17.8

Figure 1. Example Localization (A) Example setup with erroneous distance estimations, (B) Visualization and results. All values in meters.

To get an impression of the work of all localization algorithms with inaccurate distance estimations, example localization is given in Figure 1. LS and SDLS estimate the same position of the unknown node due to the limited number of beacons. Additionally it is recognizable that all coarse-grained algorithms localize the node in a triangle with the beacons as corners, which is always given by the heuristic of the algorithms.

III. Simulation environment

The described localization algorithms share one major drawback: Until now, they were only partly analyzed and not optimized to deal with realistic fading situations. Due to the node deployment near to the ground, their motionlessness and the possibly heterogeneous environment conditions, the log-normal shadow fading offers a well performing model for the real-world behavior of communicating sensor nodes [7,8]. Major part of the model is transmission equation of Friis, as given in equation (14).

$$P_r = P_t * G_t * G_r * \left(\frac{c}{4\pi f}\right)^2 * \left(\frac{1}{d}\right)^2 \tag{14}$$

Here, P_r is the received Power, P_t the transmitted Power, G_t and G_r the antenna gains, c the speed of light, f the transmission frequency and d the distance between transmitter and receiver. To adapt the transmission equation to log-normal shadow fading, an environment depending path loss exponent N and a Gaussian distributed random variable X_σ with mean value 0 and standard deviation σ is included into this equation, as given in (15), transformed to dB.

$$\begin{aligned} P_r[dB] = P_t[dB] + 20log_{10}\left(\frac{c}{4\pi f}\right) - N * 10log_{10}(d) \\ + 10log_{10}(G_t * G_r) + X_\sigma \end{aligned} \tag{15}$$

To analyze how the in Section II proposed localization algorithms perform in different environments, realistic values for the path loss exponent N and the standard deviation σ are required. In [8], the authors performed a widespread analysis of the log-normal shadow fading with a for sensor network feasible frequency band of 900 MHz. For our analysis, we selected two scenarios, from this paper.

TABLE I
SIMULATION SETUP

Property	Value
Sensor node properties	
Frequency f	900 MHz
Transmission power	0 dBm
Receiver sensitivity	98 dBm
Receiving Antenna Gain G_r	1
Transmitting Antenna Gain G_t	1
Scenario “Sandy Flat Beach”	
Path Loss exponent N	4.2
Standard deviation σ	2
Scenario “Dry Tall Underbrush”	
Path Loss exponent	3.6
Standard deviation σ	2.9
Simulation environment	
Surrounding area	300m x 300m
Internal area	100m x 100m
Beacon density [10^{-4}/m²]	2.5;5;7.5;…;22.5;25
Beacon arrangements per density	200
Unknown nodes/beacon arrangement	50

The first one is the “Sandy Flat Beach” scenario. This scenario represents a sensor network which is deployed on a beach, a desert or another sandy and relatively flat area, for example to detect vehicles. The second selected scenario is the “Dry Tall Underbrush” scenario, which emulates a sensor network deployed in a forest, e.g. to detect or prevent forest fires. To compare the algorithms with each other in the appropriate scenarios, we set up a simulation environment as given in Table 1 in Prowler [9] with minimized edge effects by creating an internal area for unknown nodes and beacons and a surrounding are only for beacon nodes. As precondition for the distance estimation, each node has knowledge about the path loss exponent *N* of its environment and is able to compute a distance to a beacon on the basis of the received signal strength by assuming an undisturbed channel and applying equation (16).

$$d[m] = 10^{\left(\frac{P_t[dB]-P_r[dB]+20log_{10}\left(\frac{c}{4\pi f}\right)+10log_{10}(G_t*G_r)}{10*N}\right)} \qquad (16)$$

It is assumed, that the path loss exponent is estimated by all communicating beacons and provided in the network.

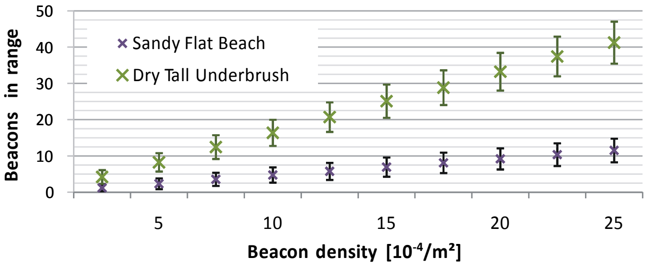


Figure 2. Beacons in range versus beacon density

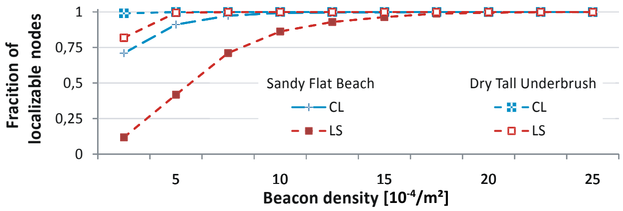


Figure 3. Fraction of localizable nodes versus beacon density. Coarse-grained algorithms require at least 1 received beacon, fine-grained algorithms at least 3 received beacons

To get a statement about the number of received beacons versus the achieved accuracy, we varied the beacon density. For each selected density, we created 200 beacon arrangements and in each arrangement 50 unknown nodes localized their position with all described algorithms, if possible. The increased beacon density correlates with a certain number of beacons in range, as shown in Figure 2, and for low density, a fraction of nodes did not received enough beacons to localize themselves, as shown in Figure 3. After applying the simulation with all beacon densities, the overall number of nodes with certain beacons in range differed extremely, as shown in Figure 4.

For statistical significance, we decided that a beacon number had to be used at least 1000 times for localization. Hence, we were able to analyze the accuracy with up to 15 received beacon nodes in the Sandy Flat Beach scenario and up to 44 received beacon nodes in the Dry Tall Underbrush scenario.

IV. PERFORMANCE ANALYSIS

To estimate the performance of the algorithms, we noted the localization error as distance of the origin position of a node and computed position of each algorithm. The simulation for the Sandy Flat Beach scenario is shown in Figure 5 and for the Dry Tall Underbrush scenario in Figure 6 as Boxplot-diagram. We selected this kind of presentation for a better visualization of the dispersion of the achieved results. The result shows that each algorithm performs better if more messages of different beacons are received. This is not surprising and covers our expectations.

The simulations allow comparing the coarse-grained and later the fine-grained algorithms among each other. In both

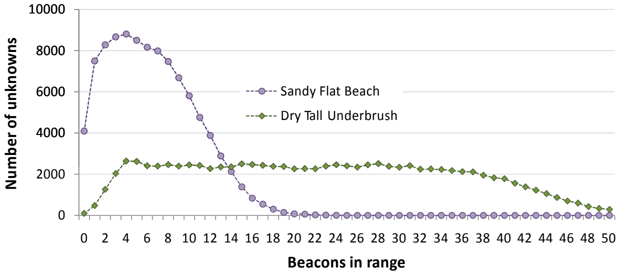


Figure 4. Overall number of simulated unknowns with certain beacons in range

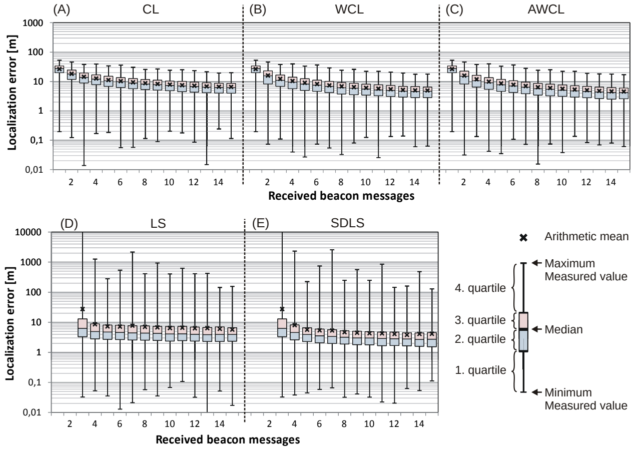


Figure 5. Accuracy of the investigated localization algorithms in the scenario "Sandy Flat Beach"

scenarios, AWCL outperforms CL and WCL by a reduced median and arithmetic mean error. The reason is given by the improved weighting of the different distances to the beacons. The outliers of all three algorithms are similar and there is no predication about the best performance possible.

In the fine-grained algorithms, the arithmetic mean and the median of SDLS outperform the linear least squares algorithm with random choice of beacon couples. The reason is the careful choice of the linearizer in SDLS, which is always one of the closest beacons. Due to the channel model, the average distance estimation error is reduced if the distance is shorter. Hence, the impact of the more accurate linearizer distance allows a performance increase compared to randomly selected beacon couples.

Furthermore, the simulations allow a comparison between the fine-grained and the coarse-grained algorithms. It is recognizable that the fine-grained algorithms perform only marginal better or even worse than the coarse-grained algorithms. Additionally, both fine-grained algorithms are characterized by a number of extreme outliers, which achieve errors in the range of hundreds up to several thousand meters. There are two reasons for such outliers. The first one is an inauspicious beacon arrangement where the received beacons do not surround the unknown node, but are (nearly) arranged in a line. The second one is erroneous distance measurements to the beacon nodes. Due to the algorithm, which does not optimize until the smallest distance error is found, but optimize the terms as given in (10) and (11), the final positions are not forced to reflect real possible positions. In contrast, the localization results of the coarse-grained algorithms localize the unknown node always anywhere between the beacons. On the one hand, this limits the maximum possible error, on the other hand, this limits the maximum achievable localization accuracy. As last part of our analysis, we investigated the correlation between errors in coarse-grained and fine-grained localization by selecting randomly 100 localized nodes with each received 3 beacon messages, and compared the localization error AWCL with WCL and SDLS of each localized node in a scatter plot, as shown in Figure 7. It is recognizable that there is a strong correlation between the two coarse-grained localization algorithms, while there is nearly no correlation between AWCL and SDLS. This different behavior is later utilized by HyPAERLoc.

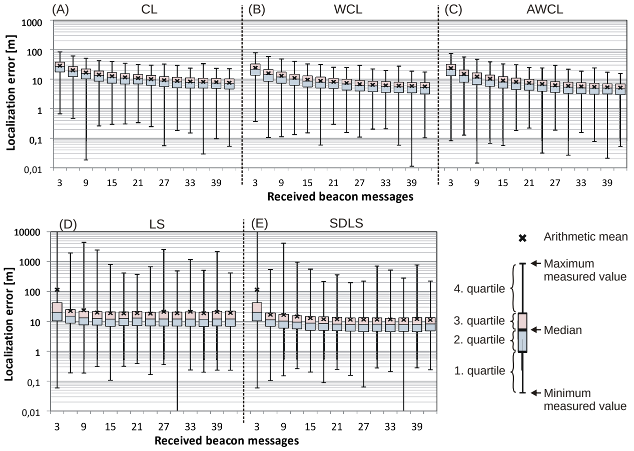


Figure 6. Accuracy of the investigated localization algorithms in the scenario "Dry Tall Underbrush"

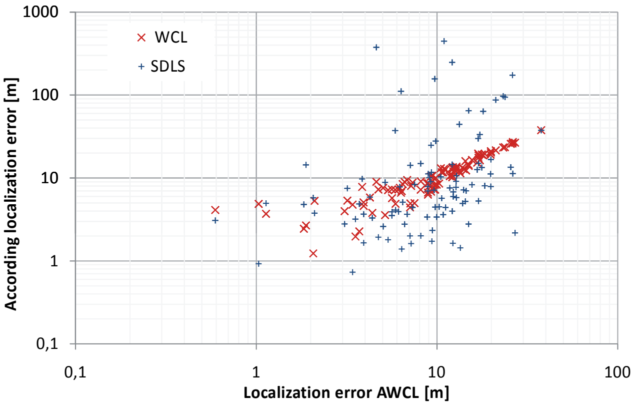


Figure 7. Correlation between sorted localization errors of SDLS-measurements and according localization errors of AWCL

Concluding, both algorithm families perform similar in the terms of accuracy in average, but with different accuracy in single beacon arrangements and distance estimations. Due to the higher accuracy potential of the fine-grained algorithms, our developed algorithm HyPAERLoc is based on the idea of SDLS, which accuracy is improved in two steps.

V. DETECTION OF IMPLAUSIBLE DISTANCE MEASUREMENTS

The major idea to improve the distance measurement is given by a plausibility check, which benefits from the known positions of each couple of beacons j and k, whose distances to the unknown i are used to create a common linear equation. With the knowledge of the beacons' positions and the estimated distances to and between them, an unknown node is able to recognize implausible gaps as result of erroneous distance measurements. A node can

TABLE II
ADAPTATION OF THE DISTANCE ESTIMATIONS

Condition	Adaptation of d_{ij}	Adaptiaton of d_{ik}
$Gap_1 > 0$	$d'_{ij} = d_{ij}\left(1 + \frac{Gap_1}{d_{ij} + d_{ik}}\right)$	$d'_{ik} = d_{ik}\left(1 + \frac{Gap_1}{d_{ij} + d_{ik}}\right)$
$Gap_2 > 0$	$d'_{ij} = d_{ij}\left(1 + \frac{Gap_2}{d_{ij} + d_{ik}}\right)$	$d'_{ik} = d_{ik}\left(1 - \frac{Gap_2}{d_{ij} + d_{ik}}\right)$
$Gap_3 > 0$	$d'_{ij} = d_{ij}\left(1 - \frac{Gap_3}{d_{ij} + d_{ik}}\right)$	$d'_{ik} = d_{ik}\left(1 + \frac{Gap_3}{d_{ij} + d_{ik}}\right)$

apply the plausibility check by solving equations (17), (18) and (19).

If one of the resulting gaps are greater than 0, at least one distance estimation is erroneous, even if the unknown and the beacons are deployed in a line, as shown in Figure 8.

$$Gap_1 = d_{jk} - (d_{ik} + d_{ij}) \tag{17}$$

$$Gap_2 = d_{ik} - \left(d_{jk} + d_{ij}\right) \tag{18}$$

$$Gap_3 = d_{ij} - \left(d_{jk} + d_{ik}\right) \tag{19}$$

Our solution to deal with this knowledge about the implausible gaps is to adapt the estimated distances between the node and the beacons, until the conditions are not longer fulfilled, as given in Table II. As result, a more realistic distance estimation is performed, which can be applied to any fine-grained localization algorithm. Applied to SDLS, the increased accuracy is shown in Figure 9. Although the dispersion is not reduced, the plausibility check increases the arithmetic mean of the algorithm in both scenarios by up to 20%..

VI. HYBRID LOCALIZATION

Unfortunately, the plausibility check is not able to tackle the outliers due to inauspicious beacon arrangements or unrecognized erroneous distance estimations. For a further improvement of the localization accuracy, we tackle extreme outliers by a comparison of the plausibility of the fine-grained algorithm result with the result of a robust coarse-grained algorithm. This hybrid approach completes our HyPAERLoc algorithm and is performed in 3 steps:

1.) Estimation of the position of an unknown node with SDLS with preceding plausibility check and with AWCL.

2.) Rating of the accuracy of the estimated position of the unknown by comparing the estimated distances to all beacons $RATE_{SDLS}$ and $RATE_{AWCL}$ with equation (20).

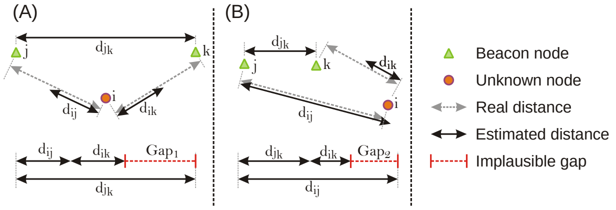


Figure 8. Implausible distance estimations (A) Condition 1, (B) Condition 2. For condition 3, exchange *j* and *k* in (B)

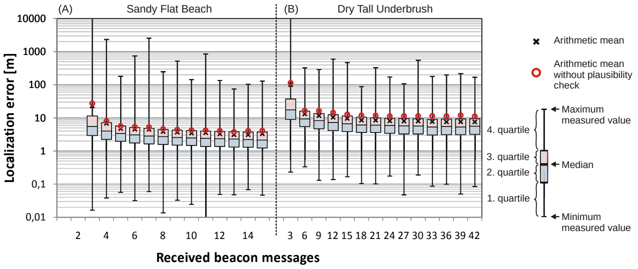


Figure 9. Accuracy of SDLS with plausibility check in both investigated scenarios

$$RATE = \sum_{j=1}^{n}\left(\sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} - d_{ij}\right)^2 \tag{20}$$

Here, x_i and y_i are the estimated position of an unknown i, x_j and y_j are the position of the beacon j, d_{ij} the estimated distance between beacon j and node i and n the number of beacons in range. As a result, a high rate correlates with bad position estimation, because the resulting position does not correlate with the estimated distances to the beacons.

3.) Computation of the final position for HyPAERLoc $x_{i,h}$ and $y_{i,h}$ with equations (21) and (22).

$$x_{i,h} = x_{i,SDLS} + \frac{RATE_{SDLS}}{RATE_{SDLS} + RATE_{AWCL}} * \left(x_{i,AWCL} - x_{i,SDLS}\right) \tag{21}$$

$$y_{i,h} = y_{i,SDLS} + \frac{RATE_{SDLS}}{RATE_{SDLS} + RATE_{AWCL}} * \left(y_{i,AWCL} - y_{i,SDLS}\right) \tag{22}$$

After applying equations (20) and (21), the position of the node is estimated between the position of SDLS and AWCL and the node is located nearer to the position with the lower rate. The result for the position estimation of HyPAERLoc as boxplot diagram is given in Figure 10. One can see that the outliers of SDLS are completely eliminated and also the accuracy of the median is increased. For a further comparison, the resulting mean averages of SDLS, AWCL and HyPAERLoc compared to the beacon density are given in Figure 11 for both scenarios. One can see that HyPAERLoc always outperform its preceding algorithms with round about 50% accuracy increase in average.

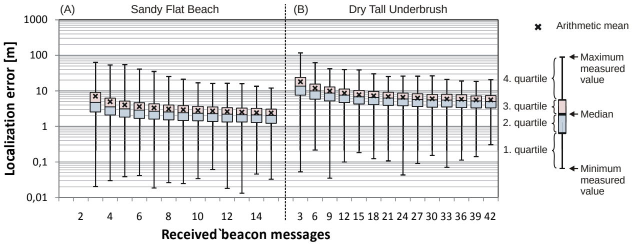


Figure 10. Accuracy of HyPAERLoc in both investigated scenarios

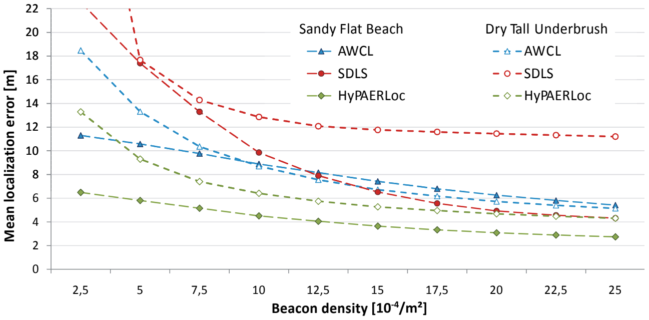


Figure 11. Average mean localization error versus beacon density. Only nodes with at least 3 beacons in range were considered

VII. Conclusion and Outlook

This paper presented HyPAERLoc as an enhanced localization algorithm of SDLS for inaccurate distance measurements. In two steps major problems of SDLS are tackled. The first one is a plausibility check, which allows detecting a fraction of erroneous distance estimations. The second one avoids outliers by evaluating the achieved positions and comparing them with the robust AWCL, which eliminates all strong outliers and improves the overall performance.

The algorithm comes along with additional computational cost, but the strong accuracy increase should compensate this drawback. Furthermore, there is no additional knowledge necessary compared to the involved localization algorithms. The algorithm is easily extendable to 3D-scenarios, which allow the application in more wireless sensor network scenarios.

Although the algorithm was tested in two scenarios with assumed log-normal fading channel model, an application onto real nodes would additionally strength the result and identifies additional challenges, e.g. erroneous RSSI measurements.

Essential questions, which are still left open in the paper are how to figure out the path loss exponent for each node and how a less random beacon deployment would impact the algorithms performance and the number of required beacon nodes. A comparison to alternative localization algorithms, e.g. MDS-map [10], is also intended in the near future.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, *A Survey on Sensor Networks*, IEEE Communication Magazine, vol. 40, no. 8, pages 102-116, Aug. 2002

[2] N. Bulusu, J. Heidemann, and D.Estrin. *GPS-less low cost outdoor localization for very small devices*. IEEE Personal Communications Magazine, 7(5):28–34, Oct. 2000.

[3] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann. *Weighted Centroid Localization in Zigbee-based Sensor Networks*, In IEEE International Symposium on Intelligent Signal Processing, WISP 2007, Madrid, Oct. 2007

[4] R. Behnke and D. Timmermann, *AWCL: Adaptive Weighted Centroid Localization as an efficient improvement of Coarse Grained Localization*, 5th Workshop on Positioning, Navigation and Communication 2008, Hannover, Deutschland, Mar. 2008

[5] A. Savvides, C. Han, and M. B. Strivastava, *Dynamic fine grained localization in ad-hoc networks of sensors*, Proceedings of the 5th Internationl Conference on Mibole Computing and Networking, Rome, Italy, July 2001

[6] R. Behnke, J. Salzmann, and Dirk Timmermann, *sDLSne - Improved Scalable Distributed Least Squares Localization with minimized Communication*, 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2010), Istanbul, Turkey, Sep. 2010

[7] J J.M. Molina-Garcia-Pardo, A. Martinez-Sala, M.V. Bueno-Delgado, E. Egea-Lopez, L. Juan-Llacer, and J. García-Haro, *Channel Model at 868 MHz for Wireless Sensor Networks in Outdoor Scenarios*, International Workshop on Wireless Ad Hoc Networks, London, May 2005

[8] K. Sohrabi, B. Manriquez, G.J. Pottie, *Near ground wideband channel measurement in 800-1000 MHz*, 49th IEEE Vehicular Conference, Houston, USA, July 1999

[9] http://www.isis.vanderbilt.edu/Projects/nest/prowler/, May 2011

[10] Y. Shang, W. Ruml, and Y. Zhang, *Localization from connectivity in sensor networks*, IEEE Transactions on Parallel Distributed Systems 15, no. 11, 961–974, 2004

MPT-MAC: A Multiple Packets Transmission MAC Protocol for Wireless Sensor Networks

Tang Hong-wei, Sun Cai-xia, Lu Kai and Liu Yong-peng
College of Computer Science
National University of Defense Technology
Changsha, Hunan, P.R. China, 410073
Email: {hwtang, cxsun}@nudt.edu.cn, dilen.lu@gmail.com and liuyp@nudt.edu.cn

***Abstract*—Event detection is a major application in wireless sensor networks (WSNs). Current Medium Access Control (MAC) protocols for WSNs are mainly optimized for the situation that an event generates only one data packet on a single node and the event occurrence rate is low. When an event generates multiple data packets or the event occurrence rate is relatively high, packet delivery latency and delivery ratio are degraded rapidly. In this paper, we present a new MAC protocol called Multiple Packets Transmission MAC (MPT-MAC) for event-based WSNs. MPT-MAC schedules multiple data packets generated by an event on a single node to be forwarded over multiple hops in an operational cycle. By this means, MPT-MAC can achieve low delivery latency and high delivery ratio under heavy traffic loads. We use event delivery latency (EDL) and event delivery ratio (EDR) to measure the event detection capability of MPT-MAC protocol. We show the performance of MPT-MAC through detailed ns-2 simulation. Compared to S-MAC-AL, R-MAC and DW-MAC, MPT-MAC can achieve lower EDL and higher EDR without more energy consumption. Furthermore, MPT-MAC can obtain lower duty cycle than DW-MAC when satisfying the latency requirement of the applications.**

***Index Terms*—wireless sensor networks, medium control access, event detection, duty cycle, event delivery latency, event delivery ratio**

I. Introduction

With the development of wireless communication, embedded computation and sensor technology, WSNs are used widely in applications including military, industry, agriculture and environmental monitoring, and have been an active research area in the past few years.

Medium Access Control (MAC) protocols control how the wireless devices access the sharing wireless channel, being fundamental protocols and key techniques in wireless sensor networks. In the wireless MAC protocol used by wireless ad hoc networks, such as IEEE 802.11[1], the wireless devices must listen to the wireless channel in order not to miss incoming packets, even when no packets are transmitted or received. Idle listening consumes significant energy[2]. Therefore, traditional MAC protocols are not suitable for WSNs in which sensor nodes are generally battery-powered.

To reduce energy consumption of idle listening, duty cycling mechanism[3][4] has been introduced in wireless sensor network MAC protocols. In duty cycling, each sensor node follows a periodic active/sleeping schedule and the percentage of time in the active state is called duty cycle. When a node is active, it turns on its radio to transmit or receive data packets. However, when sleeping, it turns off its radio to save energy. Most of existing MAC protocols for WSNs adopt duty cycling mechanism, such as S-MAC[4][5], T-MAC[6], R-MAC[7], DW-MAC[8], B-MAC[3], X-MAC[9], RI-MAC[10] and so on.

Event detection is among the major applications in wireless sensor networks. Sometimes we need the long message to describe the event. However, the cost of re-transmitting the long message is very high. So when a node detects an event, more than one small data packet may be generated to describe the event. In addition, with more and more sensor nodes deployed, multiple nodes may detect the events and transmit data packets simultaneously. Current MAC protocols for WSNs are mainly optimized for the situation that an event generates only one data packets on a single node and the event occurrence rate is low. Under such heavy traffic loads, the performance of existing MAC protocols degrades obviously.

In this paper, we present a new MAC protocol for event-based WSNs, called MPT-MAC. MPT-MAC is a synchronous duty cycle MAC protocol. It allows nodes to wake up to communicate in sleep period and to continuously transmit multiple data packets generated by an event. MPT-MAC can achieve low delivery latency and high delivery ratio under heavy traffic loads. Furthermore, with the requirement of the applications satisfied, MPT-MAC can obtain low duty cycle to save energy and prolong the network lifetime. The contributions of this work are as follows:

- Presenting a new MAC protocol MPT-MAC, that schedules sensor nodes to continuously transmit multiple data packets generated by an event.
- Analyzing the possibility of low duty cycle in MPT-MAC when satisfying the requirements of applications.
- Using event delivery latency and event delivery ratio to measure MPT-MAC.
- Evaluating MPT-MAC protocol using NS2 simulator and comparing it with existing MAC protocols.

The rest of the paper is organized as follows. In Section II, we discuss related work and analyze some synchronous MAC protocols. Section III details the design of MPT-MAC. In Section IV, we show results from our simulation-based evaluation of MPT-MAC, including a comparison with existing

MAC protocols. Finally, Section V presents the conclusions.

II. RELATED WORK

Duty cycle MAC protocols can be classified into two categories: synchronous and asynchronous. In synchronous MAC protocols, all nodes need to be synchronized and wake up simultaneously to transmit data packets, for example, S-MAC, T-MAC, R-MAC, DW-MAC, and so on.

However, in asynchronous MAC protocols, all nodes decide wake-up time according to their own schedules and need not to be synchronized. Asynchronous protocols mainly include B-MAC, X-MAC, RI-MAC, and so on. MPT-MAC presented in this work is a synchronous duty cycle MAC protocol, so we only discuss synchronous MAC protocols in this section.

S-MAC[4] was one of original synchronous duty cycle MAC protocols for WSNs. Figure 1 shows an overview of S-MAC. A cycle of S-MAC is composed of three periods: SYNC, DATA and SLEEP. At the beginning of SYNC period, the node wakes up to broadcast a SYNC packet to synchronize neighbor nodes. In DATA period, if node A wants to send a data packet to node B, they use RTS/CTS/DATA/ACK to complete data transmission. After transmitting the packet, node A and B turn to sleep. Node C without data communication will turn off its radio to sleep at the beginning of SLEEP period.

In S-MAC, nodes periodically alternate between being active and sleeping to reduce energy consumption of idle listening. But in one operational cycle, a data packet can be forwarded only one hop, so multi-hop transmission latency will be greatly increased. Wei Ye et al. proposed S-MAC with adaptive listening(S-MAC-AL)[5] to reduce data transmission latency. As shown in Figure 1, if node C overhears CTS packet from B to A, it will adaptively wake up after the transmission between A and B is done. After node B received a data packet, it will send an RTS to C. If C is the next hop of the data packet, node B can immediately forward the data packet to C and needs not to wait the next cycle. By adaptive listening, a data packet can be delivered up to two hops in one operational cycle.

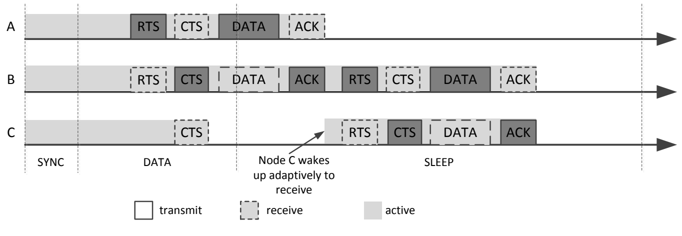


Fig. 1: Schedule of S-MAC and S-MAC-AL

The duration time of DATA period in S-MAC and S-MAC-AL is fixed. Even though nodes have no data packets to communicate in the current cycle, they wait to sleep until DATA period is ended. Nodes in WSNs have no data communications in most of time, so idle listening of DATA period will waste significant energy. The fixed DATA period is not suitable for light traffic load. T-MAC[6] is primarily designed to shorten the DATA period when no traffic is around the nodes, so that nodes can preserve more energy. Its principle is that nodes go to sleep if they cannot detect any specified events in TA. TA is the minimum idle listening time of nodes in a cycle. Although T-MAC can preserve more energy than S-MAC when there is no traffic, it also only delivers a packet up to two hops within one operational cycle and cannot reduce multi-hop delivery latency of data packets.

Some approaches are proposed to reduce deliver latency. However, they make some specific assumptions on the communication pattern. For example, D-MAC[11] reduces data delivery latency only for data gathering tree. The streamlined wakeup optimization proposed by Cao et al.[12] addresses only the case in which each sensor node sends data to a sink node. Lu et al.[13] discusses how to minimize end-to-end delivery latency for a tree or a ring network.

R-MAC[7] introduces a new cross-layer approach to reduce packet delivery latency in multi-hop forwarding. Figure 2 shows the schedule of R-MAC. RTS/CTS in S-MAC are replaced by the pioneer frame (PION) of R-MAC. In an operational cycle, PION is forwarded over multiple hops during DATA period to inform nodes B and C when to wake up to receive or transit the data packets during SLEEP period. According to the number of hops carried in PION, nodes that are on the data forwarding path calculate their wake-up time during the SLEEP period using the equation (1).

$$T_{wakeup}(i) = (i{-}1)\cdot(durDATA{+}SIFS{+}durACK{+}SIFS) \quad (1)$$

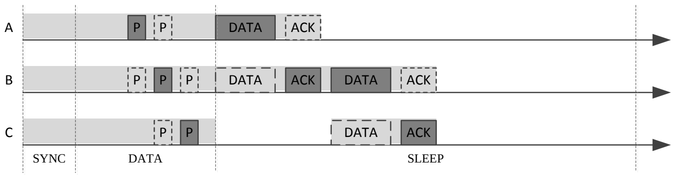


Fig. 2: Multi-hop forwarding of R-MAC

The process of PION forwarding goes on till the DATA period is over, so the number of hops over which R-MAC can forward a data packet in a cycle is limited by the duration of the DATA period. However, a source node (e.g., node A in Figure 2) always starts transmitting a data packet at the beginning of the SLEEP period, two hidden terminal nodes that have succeeded in contending the channel in the DATA period will cause collision at the following SLEEP period.

In order to resolve the collision between the hidden source nodes at the SLEEP period, DW-MAC[8] uses one-to-one mapping to schedule nodes to wake up. Figure 3 gives the overview of the scheduling approach in DW-MAC. In this example, node A wants to transmit a data packet to node B. A firstly contends the channel and then transmits a SCH control frame during the DATA period. Supposed that transmission of SCH starts at T_1 units after the beginning of the DATA period and the duration of transmission is T_3. Based on T_1, T_3, the ratio between T_{Sleep} and T_{Data} and the equation (2), we can calculate the wake-up time T_2 from the beginning of the

SLEEP period of node A and B, and the maximum wake-up duration T_4. By one-to-one mapping function, data transmission during the SLEEP period will not collide. Furthermore, DW-MAC uses cross-layer approach like R-MAC to reduce multi-hop delivery latency.

$$\frac{T_2}{T_1} = \frac{T_4}{T_3} = \frac{T_{Sleep}}{T_{Data}} \tag{2}$$

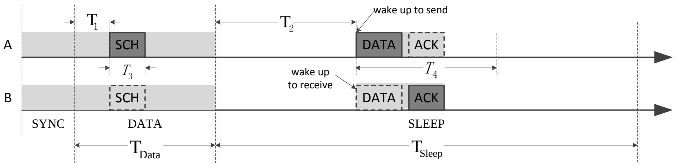


Fig. 3: Overview of the schedule in DW-MAC

Although DW-MAC resolves the problem that the hidden source nodes collide at the beginning of the SLEEP period, it only schedules one data packet to forward during the SLEEP period. If multiple data packets are generated by an event on a single node, DW-MAC has to schedule nodes to transmit data packets to the sink node in multiple operational cycles. This increases the data packets delivery latency. In addition, DW-MAC transmits SCH control frame for each data packet and multiple SCHs waste more energy.

These MAC protocols propose several approaches to reduce packet delivery latency. However, they are optimized for one data packet of an event and low event occurrence rate. MPT-MAC proposed in this paper can schedule multiple data packets generated by an event on a single node to be forwarded over multiple hops in an operational cycle, so it can work well under heavy traffic loads, such as when an event generates multiple data packets on a single node and the event occurrence rate is very high.

III. MPT-MAC DESIGN

A. Overview

MPT-MAC is a synchronous duty-cycle MAC protocol. Each operational cycle of MPT-MAC is also divided into three periods: SYNC, DATA and SLEEP period. We denote the duration of each period by T_{Sync}, T_{Data} and T_{Sleep} respectively. Similar to prior works, MPT-MAC must use synchronizing mechanisms[14][15] to resolve the clock drift and ensure to synchronize the clock in sensor nodes.

The principle of MPT-MAC is that nodes are scheduled to wake up during SLEEP period and to deliver multiple data packets over multiple hops in an operational cycle. In order to deliver multiple data packets, the receiver node that is scheduled to wake up in the SLEEP period waits for a little duration T_{wait} after receiving a data packet. If the node does not receive any data packet during T_{wait}, it will go to sleep.

Figure 4 shows the example of multiple data packets transmission in MPT-MAC. In this example, node A detects an event and generates two data packets for this event. These two packets need to be transmitted to node B. According to the scheduling algorithm of MPT-MAC, node A and B wake up to transmit the data packet at T_1. Unlike DW-MAC, node B will keep listening to the channel. If there are other data packets in the A's queue, Node A can transmit the data packet D2 to B SIFS delay after receiving ACK for D1 from B. Once A receives ACK for D2, A goes to sleep. However, node B will wait for a little duration T_{wait} after receiving D2. If B doesn't receive anything during T_{wait}, it will go to sleep. As described in the example, node A only needs one operational cycle and one SCH frame to transmit two data packets to B in MPT-MAC, but DW-MAC needs two operational cycles and two SCH frames. With the number of the data packets generated by an event on a node increasing, the number of the operational cycles and the SCH frames needed by DW-MAC will increase accordingly.

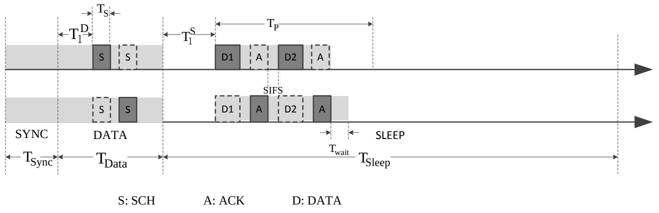


Fig. 4: Multiple packets scheduling of MPT-MAC

B. Wakeup Scheduling

MPT-MAC uses one-to-one mapping function to schedule nodes to wake up intelligently, just like DW-MAC. Node A that wants to transmit a data packet to node B contends the wireless channel using CSMA/CA protocol in IEEE 802.11. Once succeeding in contending the channel, node A will transmit a special SCH frame (SCH includes all fields of RTS/CTS and has the same function with RTS/CTS) that replaces RTS control frame. Node B replies with a SCH frame as CTS. Node A and B calculate their wake-up time T_i in the SLEEP period using the equation (3) respectively.

$$T_i^S = SDTR \cdot T_i^D \tag{3}$$

In the equation (3), we denote by $SDTR = \frac{T_{Sleep}}{T_{Data}}$ the ratio between the duration of the SLEEP period and the DATA period and by T_i^D the time difference between nodes transmitting/receiving SCH frame and the beginning of the DATA period.

As shown in Figure 5, in order to reduce multi-hop delivery latency of data packets transmission, MPT-MAC uses cross-layer approach to schedule multiple data packets to forward over multiple hops in a cycle. In the example, node B will send its own SCH frame after receiving a SCH frame from the up hop node A. The SCH frame transmitted by node B plays two roles: firstly, it is the ACK of node A's SCH frame, secondly, the next hop node C receiving B's SCH frame uses the mapping function in the equation (3) to calculate wake-up time to receive data packets, so that multiple data packets can be forwarded over multiple hops in an operational cycle.

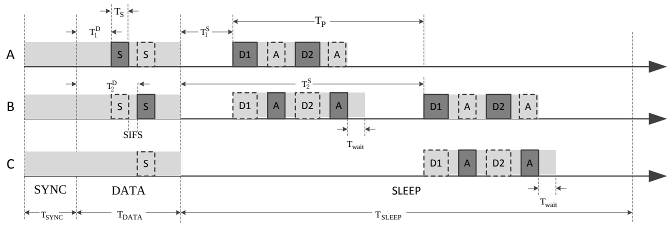


Fig. 5: Multiple hops optimization in MPT-MAC

C. Multiple Packets Transmission

We denote by T_P the maximum time of a node occupying the channel during the SLEEP period, if it succeeds transmitting the SCH frame during the DATA period. According to the equation (3) and Figure 5, the time of node A using the channel without collision in the SLEEP period can be calculated by the following equation:

$$\begin{aligned} T_P &= T_2^S - T_1^S \\ &= SDTR \cdot T_2^D - SDTR \cdot T_1^D \\ &= SDTR \cdot (T_2^D - T_1^D) \\ &= SDTR \cdot (T_S + SIFS) \qquad (4) \end{aligned}$$

In DW-MAC, node A will go to sleep immediately after transmitting data packet D1 in the time interval T_P. However, the communication latency u between two nodes is less than T_P in common. We denote u as the following equation (5):

$$u = durDATA + SIFS + durACK + SIFS \qquad (5)$$

Therefore, MPT-MAC can transmit multiple data packets in the time T_P. In order to avoid collision between A's transmission and B's, according to the equations (4) and (5), the maximum number of data packets transmitted by node A is $N_{max} = \lceil \frac{SDTR \cdot (T_S + SIFS)}{u} \rceil$. Each node must maintain a transmitting/receiving counter. The counter is added by 1 when the node transmits or receives a data packet.

When one of the following three situations happens, the node does not transmit the data packets in the queue or receive data packets any more, and goes to sleep:

1. The value of counter is equal with N_{max};
2. Nodes find that the remain time of T_P is less than u;
3. The receiver node cannot receive any data packet in T_{wait}, it turn to sleep. Because the sender node transmits the next data packet SIFS delay after receiving ACK of the previous data packet, we denote $T_{wait} = SIFS + T_{MAX\_PRO\_DLY}$, where $T_{MAX\_PRO\_DLY}$ presents the maximum propagation delay.

D. Low Duty Cycle

Duty cycle is denoted by the ratio between the active time of a node and the cycle:

$$duty\_cycle = \frac{T_{Sync} + T_{Data}}{T_{Sync} + T_{Data} + T_{Sleep}}$$

If a node has lower duty cycle, it will consume less energy and the lifetime of the node is longer. However, low duty cycle increases the sleep latency during the data packets forwarded. The major challenge of MAC protocol design for WSNs is how to tradeoff between low latency and energy consumption.

Compared with DW-MAC, MPT-MAC can schedule multiple data packets to deliver multiple hops without collision in an operational cycle, so it is possible for MPT-MAC to achieve lower duty cycle than DW-MAC. We analyze the relationship between the duty cycle and the data packets delivery latency, when a node transmits two data packets in one hop in DW-MAC and MPT-MAC. We give some following assumptions in DW-MAC and MPT-MAC to simplify the analysis:

1. Nodes always generate data packets when waking up.
2. Nodes always succeed in contending the channel at the same time T_D.
3. Nodes have the same T_{Sync} and T_{Data}, so $T_{Listen} = T_{Sync} + T_{Data}$.
4. The duration of one RTS/CTS handshake is $t = 2 \cdot durCtrl + SIFS$. We use $durCtrl$ to present the duration of control packet transmission.
5. DW-MAC's duty cycle is d, and MPT-MAC's duty cycle is $k \cdot d$, where k is a constant.

According to the wake-up scheduling mechanism described in subsection III-B, we can calculate the two data packets deliver latency in one hop in DW-MAC and MPT-MAC respectively:

$$\begin{aligned} Delay_{DW} &= T_{Cycle}^{DW} + SDTR_{DW} \cdot T^D + u \\ &= T_{Cycle}^{DW} + \frac{T_{Listen} \cdot (1-d)}{T_{Data} \cdot d} \cdot T^D + u \\ &= T_{Cycle}^{DW} + T_{Cycle}^{DW} \cdot \frac{T^D}{T_{Data}} \cdot (1-d) + u \qquad (6) \end{aligned}$$

$$\begin{aligned} Delay_{MPT} &= SDTR_{MPT} \cdot T^D + 2 \cdot u \\ &= \frac{T_{Listen} \cdot (1-kd)}{T_{Data} \cdot kd} \cdot T^D + 2 \cdot u \\ &= T_{Cycle}^{DW} \cdot \frac{T^D}{T_{Data}} \cdot \frac{1-kd}{k} + 2 \cdot u \qquad (7) \end{aligned}$$

The difference between $Delay_{DW}$ and $Delay_{MPT}$ is given by the following equation (8) according to the equations (6) and (7):

$$\begin{aligned} Delay_{MPT} - Delay_{DW} &= T_{Cycle}^{DW} - u + T_{Cycle}^{DW} \cdot \frac{T^D}{T_{Data}} \\ &\quad - \frac{1}{k} \cdot T_{Cycle}^{DW} \cdot \frac{T^D}{T_{Data}} \qquad (8) \end{aligned}$$

From the equation (8), as long as k satisfies the condition of $k \geq \frac{T_{Cycle}^{DW} \cdot T^D}{(T_{Cycle}^{DW} - u) \cdot T_{Data} + T_{Cycle}^{DW} \cdot T^D}$, the delivery latency of DW-MAC will be greater than that of MPT-MAC, $Delay_{DW} \geq Delay_{MPT}$. Obviously, T_{Cycle}^{DW} is much greater than u, so we can draw a conclusion $\frac{T_{Cycle}^{DW} \cdot T^D}{(T_{Cycle}^{DW} - u) \cdot T_{Data} + T_{Cycle}^{DW} \cdot T^D} < 1$. Therefore, when k is greater than a number less than 1, which means the duty cycle of MPT-MAC is less than that of DW-MAC, MPT-MAC will achieve lower data packets delivery latency than DW-MAC.

ISBN: 978-1-61208-144-1

IV. SIMULATION AND EVALUATION

A. Measure Metrics

For the event detection applications, packet delivery latency (PDL) and packet delivery ratio (PDR) cannot reflect well the capability of event detection in WSNs. Therefore we introduce event delivery latency (EDL) and event delivery ratio (EDR)[16].

- Event Delivery Latency. Supposed that Node S detects an event at T_0 and generates N data packets to describe the event. The sink node R receives all data packets of the event at T_1, we denote $EDL = T_1 - T_0$.
- Event Delivery Ratio. EDR is the ratio between the events succeeded in receiving by the sink node and the number of the events detected by source nodes. Only if the sink node receives all data packets of an event, we call that the sink node succeeds in detecting this event.

To some extent, EDL and EDR can also reflect the network's PDL and PDR. EDL and EDR are more suitable for event-based WSNs, because they reflect well the capability of event detection.

B. Simulation Environment

We evaluate MPT-MAC using version 2.29 of the NS2 simulator and compare it with S-MAC-AL, R-MAC and DW-MAC.

Table I lists the key network parameters used in our simulations. These parameters are the default values in the S-MAC-AL module distributed with NS-2 package. They are used also in the simulations of R-MAC and DW-MAC. We ignore the state transition power and energy consumed by other modules such as CPU and memory[17].

TABLE I: Network parameters

Bandwidth	20Kbps	Tx Range	250m
Tx Power	0.5 W	Carrier Sensing Range	550 m
Rx Power	0.5 W	Contention Window	64 ms
Idle Power	0.45 W	Size of RTS/CTS/ACK	10 B
Sleep Power	0.05 W	Size of SCH	14 B
SIFS	5 ms	Size of Data	50 B
DIFS	10 ms	Channel Encoding Ratio	2
		Slot Time	1 m

Traffic loads are generated by constant bit rate (CBR) flows. CBR can generate variable size data packets (50 bytes in common). So we can simulate the situation that an event detected by a node generates multiple packets by setting UDP's packet size to 50 bytes. For example, if the data generated by CBR is 100 bytes, UDP will send two data packets, which presents a node generates two data packets when detecting an event. Intermediate replying nodes do not aggregate or compress data. We also assume that data processing at any node can be finished within a SIFS duration, so data processing will not introduce extra latency. The transmission latency of all types of packets can be calculated by the equation (9), where we choose 5 bytes for the preamble size p and 2 for channel encoding ratio Encode_Ratio in our simulations.

$$durPkt = \frac{Size_{Pkt} \cdot Encode\_Ratio + p}{Bandwidth} + 1ms \tag{9}$$

Table II lists the transmission latency of all types of packets.

TABLE II: Transmission latency of packets

Type of Packet	Size of Packet(B)	Latency(ms)
RTS/CTS/ACK	10	11
SCH	14	14.2
DATA	50	43

In order to evaluate the MPT-MAC's performance under lower duty cycle, we adopt variable duty cycle in MPT-MAC. However, we keep the same duty cycle of 5% for other MAC protocols. The duration of SYNC, DATA, SLEEP and duty cycle are shown in Table III.

We use two types of scenarios for our simulations: chain and grid network.

Figure 6 give an example of a chain scenario. All nodes are equally spaced in a straight line and neighbor nodes are placed 200 m apart to compose a chain. A CBR that generates the event periodically is connected with node 0 as the source node, and node n is the sink node. In our simulations, we use 21 nodes to compose the chain, so from the source node to the sink node is 20 hops.

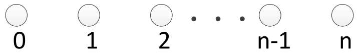


Fig. 6: n-1 hops chain

In the grid network scenario, the 7x7 grid network is composed of 49 nodes. As shown in Figure 7, the x coordinate and the y coordinate of each node are 200m apart. The sink node locates the center of the grid network, and its coordinate is (600, 600).

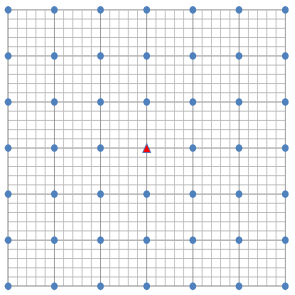


Fig. 7: 7x7 grid network

Based on a correlated-event workload[18], we use a Random Correlated-Event (RCE)[8] to simulate random events. RCE randomly selects a coordinate (x, y) and generate an event there. If the sensing radius of a node is R, only the nodes can detect the event within the circle centered at (x, y) with radius R. With R increasing, more nodes will detect the event and the traffic loads become heavier. Table IV shows the average

TABLE III: Duty cycle configuration

MAC	T_{Sync}(ms)	T_{Data}(ms)	T_{Sleep}(ms)	T_{Cycle}(ms)	Duty Cycle
S-MAC-AL	55.2	104.0	3025.8	3185.0	5%
R-MAC	55.2	168.0	4241.8	4465.0	5%
DW-MAC	55.2	168.0	4241.8	4465.0	5%
MPT-MAC	55.2	168.0	variable	variable	variable

number of nodes detecting the event with different R. In our simulations, we can adjust the sensing radius of nodes and the number of data packets generated by a node detecting an event to simulate the different types of scenario. In the chain and grid network scenarios, we simulate that a node generates multiple data packets when it detects an event by adjusting the size of data packet in UDP, so that we can evaluate the event detection capability of MPT-MAC.

TABLE IV: Number of nodes detecting the event with different sensing radius

Range	100	150	200	250	300	350	400	450	500
Nodes	0.8	1.8	3.1	4.7	6.5	8.6	10.9	13.3	15.8

C. Event Delivery Latency Evaluation

In this subsection, we evaluate the EDL of MPT-MAC. A node can generate N data packets when detecting an event. EDL is the interval from the source node detecting the event to the sink node receiving all data packets of the event. The average EDL is the average value of EDL of all events.

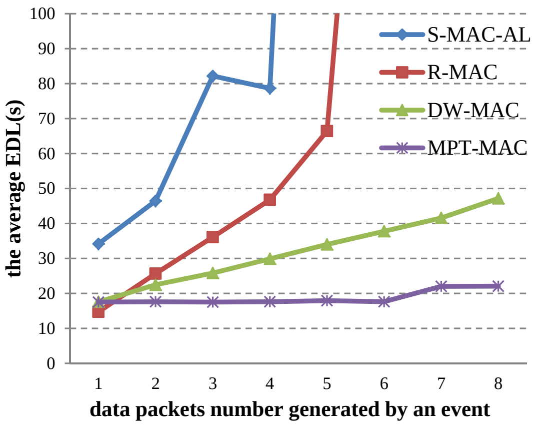


Fig. 8: Average EDL in the 20-hops chain

For the 20-hops chain scenario, we evaluate the average EDL when the number of data packets generated by an event is from 1 to 8. In our simulations, the source node generates an event per 50s. From Figure 8, we find that the average EDL of S-MAC-AL increases to 82.19197s when $N = 4$, and the average EDL of R-MAC increases to 66.42565s when $N = 5$. However, MPT-MAC's average EDL doesn't increase nearly when $N \leq 6$, and it only increases 20% when $N \geq 7$. Because MPT-MAC can schedule multiple data packets to deliver over multi-hop in one operational cycle, when $N \geq 2$, the average EDL of MPT-MAC outperforms DW-MAC. Furthermore, in the case of $N = 8$, the average EDL of MPT-MAC is 22.1s, but the average of EDL of DW-MAC is 47.2s. MPT-MAC reduces the average EDL over DW-MAC by 46.8%.

Figure 9 shows the results of our EDL evaluation for grid network. In our simulations, we keep the sensing radius of nodes 200m, RCE generates an event per 200s and the number of data packets generated by the event is 1 to 8.

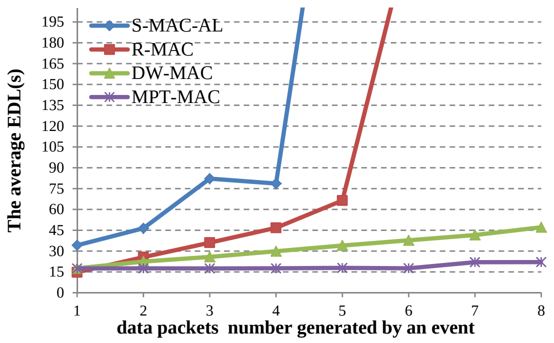


Fig. 9: Average EDL in the 7x7 grid network

From Figure 9 we can see that the average EDLs of S-MAC-AL, R-MAC and DW-MAC all increase with the number of data packets increasing. However, MPT-MAC's average EDL is about 7.9s when $N \leq 6$. When $N = 8$, the average EDL of MPT-MAC is 16.13227s, and DW-MAC is 73.59882s. The average EDL of MPT-MAC is only 22% of DW-MAC.

D. Event Delivery Ratio Evaluation

In this subsection, we evaluate the event delivery ratio of MPT-MAC. The number of data packets generated by an event is 6 in the event deliver ratio evaluation.

In the chain scenario, only node 0 can generate events. Consequently, in order to evaluate the EDR of MPT-MAC, we adjust the event generating rate from an event per 50s to an event per 15s.

Figure 10 shows the EDR of MAC protocols under the different event generating rate. We can find the EDRs of S-MAC-AL and R-MAC are always less than 1, and when the event generating rate reaches an event per 15s, the EDR of R-MAC and S-MAC-AL significantly reduces to 0.13 and 0.0945 respectively. The EDR of DW-MAC keeps 1 until the event generating rate increases to an event per 25s. When the event generating rate increases to an event per 20s, the EDR of DW-MAC reduces to 0.8. And the EDR of DW-MAC is only 0.236 when generating an event per 15s. However, the EDR of MPT-MAC always remains 1 until an event per 15s.

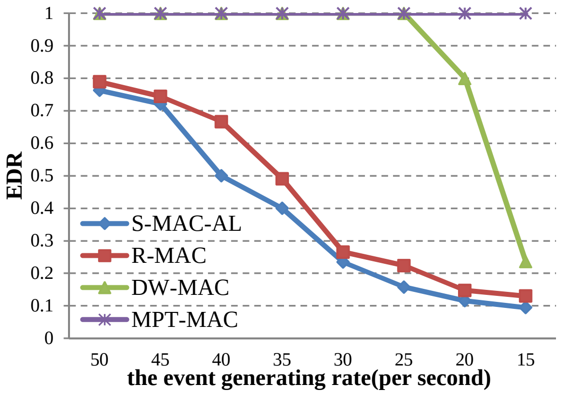


Fig. 10: EDR in the 20-hops chain

For the grid network, there are more nodes that detect the event with the increase of the node's sensing radius, so the traffic loads of the network is increased. Figure 11 shows the results of the EDR evaluation for 7x7 grid network under the different sensing range when RCE generates an event per 200s. We find that the EDR of S-MAC-AL and R-MAC is 0.083 and 0.156 respectively when the node's sensing radius increases to 500m. When the node's sensing radius is 500m, the EDR of DW-MAC is 0.711, and the EDR of MPT-MAC still is about 0.9, which is 25% higher than that of DW-MAC.

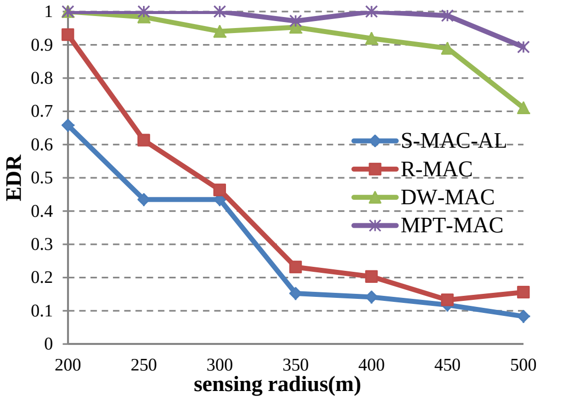


Fig. 11: EDR in the 7x7 grid network

E. Energy Consumption Evaluation

In this subsection, we evaluate the energy efficiency of MPT-MAC. We vary the number of data packets generated by an event on a single node from 1 to 8 in the chain and grid network scenarios, and observe the average energy consumption during the entire simulation.

Figure 12 shows the average energy consumption in the chain scenario. When the number of data packets is increased, the average energy consumption of S-MAC-AL and R-MAC both increase. However, the average energy consumption of DW-MAC and MPT-MAC increase slowly. Because less SCH frames are transmitted in MPT-MAC than in DW-MAC, the average energy consumption of MPT-MAC is always little lower than DW-MAC. When an event generates 8 data packets in a single node, the average energy consumption of MPT-MAC is 5% less than DW-MAC.

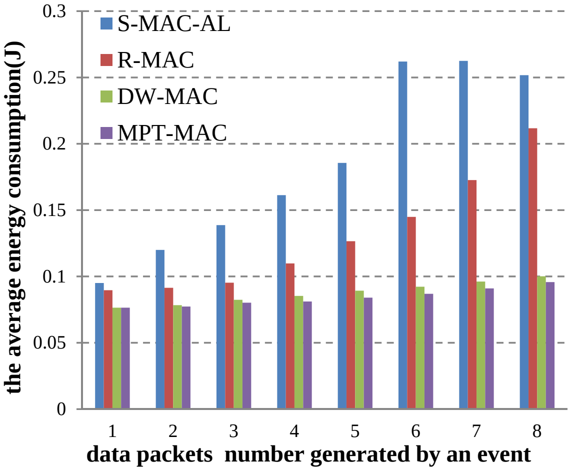


Fig. 12: Average energy consumption in the 20-hops chain

Figure 13 shows the average energy consumption for 7x7 grid network. In this simulation, we set the sensing radius as 200m and RCE generates an event per 200s. We find that the energy efficient of MPT-MAC is as much as that of DM-MAC. However, Figure 9 shows the average EDL of MPT-MAC is much less than that of DW-MAC under this condition.

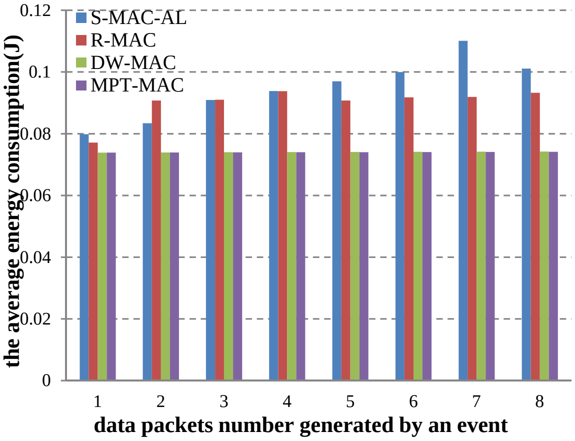


Fig. 13: Average energy consumption in the 7x7 grid network

F. Duty Cycle Evaluation

According to the analysis in the subsection III-D, MPT-MAC can achieve comparable or better EDL than DW-MAC with 5% duty cycle. In this subsection, we evaluate the EDL and the average energy consumption of MPT-MAC when it adopts variable duty cycle from 2% to 5%, but the duty cycle of DW-MAC keeps 5%. We only use 7x7 grid network to evaluate the duty cycle of MPT-MAC. In our simulation, we keep the sensing radius of 200m for MPT-MAC and DW-MAC. RCE generates an event per 200s and each event generates 6 data packets.

The average EDL of MPT-MAC with different duty cycle is shown in Figure 14. We find that the average EDL of MPT-MAC remains about 10s when the duty cycle is less than 3.5%, and the average EDL of MPT-MAC increase obviously when the duty cycle is less than 3%. The average EDL of DW-MAC is about 51.04s. Even though the duty cycle of MPT-MAC reduces to 2%, the average EDL of MPT-MAC is 22.3s, that is still about 50

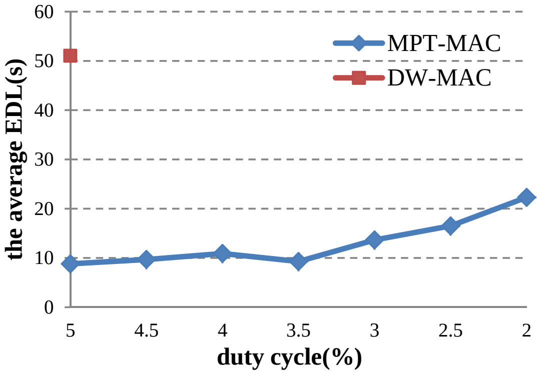


Fig. 14: Average EDL of MPT-MAC with different duty cycle

From the results shown by Figure 14, we draw a conclusion that the average EDL of MPT-MAC is much lower than that of DW-MAC even when the duty cycle of MPT-MAC is only 2%. So it is possible for MPT-MAC to achieve higher energy efficient. Figure 15 shows that the average energy consumption of MPT-MAC with 2% duty cycle is 8% less than that of DW-MAC with 5% duty cycle. During the entire simulation, in most of time nodes have data communications. When the traffic loads are ultra-light or even zero, lower duty cycle gains higher energy efficient.

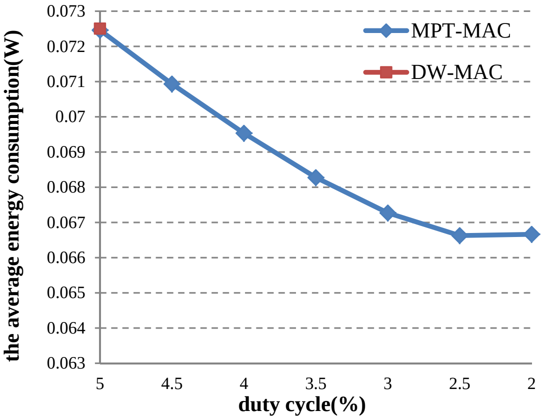


Fig. 15: Average EDL of MPT-MAC with different duty cycle

V. Conclusion

In this paper, we presented MPT-MAC, a new synchronous duty cycle MAC protocol for event-based WSNs to reduce event delivery latency and to increase event delivery ratio under heavy traffic loads. With the number of data packets generated by an event on a single node and the event generation ratio increasing, MPT-MAC achieved lower EDL and higher EDR than the existing MAC protocols without more energy consumption. Furthermore, MPT-MAC with 2% duty cycle achieved lower EDL than DW-MAC with 5% duty cycle, which means that MPT-MAC can preserve more energy and prolong the lifetime of the WSNs.

Acknowledgment

We thank for Jiang Jie for his help during our simulation. We also thank the reviewers for their valuable feedbacks that helped to improve this paper. This work is supported by the State Key Laboratory of High-End Server & Storage Technology under the grant No. 2009HSSA04.

References

[1] LAN MAN Standards Committee of the IEEE Computer Society, ”Wireless LAN medium access control (MAC) and physical layer (PHY) specification,” IEEE, New York, NY, USA, IEEE Std 802.11-1997 edition, 1997.
[2] M. Stemm and R. H. Katz, ”Measuring and reducing energy consumption of network interfaces in hand-held devices,” IEICE Transactions on Communications, vol. vol. E80-B, no. 8, pp. 1125-1131, Aug. 1997.
[3] J. Polastre, J. Hill, and D. Culler, ”Versatile Low Power Media Access for Wireless Sensor Networks,” in In Proceedings of the Second International Conference On Embedded Networked Sensor Systems (SenSys 2004), Nov. 2004, pp. 95-107.
[4] W. Ye, J. S. Heidemann, and D. Estrin, ”An Energy-Efficient MAC Protocol for Wireless Sensor Networks,” in In Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), Jun. 2002, pp. 1567-1576.
[5] W. Ye, J. Heidemann, and D. Estrin, ”Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks,” IEEE/ACM Transactions on Networking, vol. 12(3), pp. 493-506, 2004.
[6] T. v. Dam and K. Langendoen, ”An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks,” in In Proceedings of the First International Conference On Embedded Networked Sensor Systems (SenSys 2003), Nov. 2003, pp. 171-180.
[7] S. Du, A. K. Saha, and D. B. Johnson, ”RMAC: Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks,” in In Proceedings of the 26th Annual IEEE Conference on Computer Communications (INFOCOM 2007), May 2007, pp. 1478-1486.
[8] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, ”DW-MAC: A Low Latency, Energy Efficient Demand-Wakeup MAC Protocol forWireless Sensor Networks,” in In Proceedings of The 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2008), May 2008, pp. 53-62.
[9] M. Buettner, G. V. Yee, E. Anderson, and R. Han, ”X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks,” in In Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys 2006), 2006, pp. 307-320.
[10] Y. Sun, O. Gurewitz, and D. B. Johnson, ”RI-MAC: A Receiver-initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks,” in Proceedings of the 6th ACM conference on Embedded network sensor systems (SenSys 2008), 2008, pp. 1-14.
[11] G. Lu, B. Krishnamachari, and C. S. Raghavendra, ”An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks,” in In Proceedings of the 18th International Parallel and Distributed Processing Symposium, Apr. 2004.
[12] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, ”Towards Optimal Sleep Scheduling in Sensor Networks for Rare-Event Detection” in In Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN 2005), Apr. 2005, pp. 20-27.
[13] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, ”Delay Efficient Sleep Scheduling in Wireless Sensor Networks,” in In Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005), Mar. 2005, pp. 2470-2481.

[14] J. Elson, L. Girod, and D. Estrin, ”Fine-Grained Network Time Synchronization using Reference Broadcasts,” in In Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI 2002), Dec. 2002, pp. 147-163.

[15] S. Ganeriwal, R. Kumar, and M. B. Srivastava, ”Timing-Sync Protocol for Sensor Networks,” in In Proceedings of the 1st International Conference On Embedded Networked Sensor Systems (SenSys 2003), Nov. 2003, pp. 138-149.

[16] C. Cano, B. Bellalta, J. Barcel, and A. Sfairopoulou, ”A Novel MAC Protocol for Event-Based Wireless Sensor Networks: Improving the Collective QoS,” in Proceedings of the 7th International Conference on Wired/Wireless Internet Communications (WWIC 2009), 2009, pp. 1-12.

[17] V. Shnayder, M. Hempstead, B.-r. Chen, G. W. Allen, and M. Welsh, ”Simulating the Power Consumption of Large-Scale Sensor Network Applications,” in In Proceedings of the 2nd International Conference On Embedded Networked Sensor Systems (SenSys 2004), Nov. 2003, pp. 188-200.

[18] B. Hull, K. Jamieson, and H. Balakrishnan, ”Mitigating Congestion in Wireless Sensor Networks,” in In Proceedings of the Second International Conference On Embedded Networked Sensor Systems (SenSys 2004), Nov. 2004, pp. 134-147.

On Energy-Aware Forwarding Schemes in Wireless Sensor Networks

Adrian Fr. Kacsó
Computer Science Department
University of Siegen
57068 Siegen, Germany
Email: adrian.kacso@uni-siegen.de

***Abstract*—The paper discusses two forwarding strategies in wireless sensor networks (WSNs) involving the residual energy of nodes and provides the corresponding distributed algorithms; the routing is achieved without additional routing messages. To reduce excessive broadcasts at the setup phase we consider the impact of the delay time. Simulation results are obtained using our sensor network simulator (SNF).**

***Keywords*-WSN, energy-awareness, routing, simulation.**

I. Introduction

A WSN is a self-organizing communication network consisting of a large number of sensor nodes that are randomly (and densely) deployed in a region, to monitor the environment or some physical phenomenon. Therefore, the distributed nodes individually sense the environment and collaboratively preprocess and communicate their captured information to interested clients (sinks). In request-driven WSNs a sink sends a request (interest) in a data-centric manner, where the destination is specified by tuples of attribute-value pairs of the data carried inside the message. Routing protocols determine routes how messages (interest and data) are forwarded between the sink and sources (nodes able to deliver the requested data) using data-centric approaches. Due to energy constraints, source nodes usually cannot sent the data to sink(s) directly. The data is forwarded by intermediate nodes until it reaches the intended sink(s). The limited energy, the restricted communication and computation abilities (capabilities) of battery powered sensor nodes require energy-efficient routing protocols. The data gathered in a sensor network is highly correlated, due to a spatial and temporal correlation between successive measurements. Exploiting the data-centricity and the spatial-temporal correlation characteristics allows to apply effective in-network techniques, which further improve the energy-efficiency of the communication in WSN. Aggregation can eliminate the inherent redundancy of the raw data collected and, additionally, it reduces the traffic in the network, avoiding in this way congestions and induced collisions.

Routing means to find the right route between the many routes from source(s) to sink(s) by defining a path metrics. The strategy to select the next hop employs various metrics which allows to find different paths, e.g., energy-efficient, shorter, rapid, reliable paths according to the application goals. Some metrics combines the power consumption and latency, whereas others focus on prolonging the lifetime of the network by considering the node's residual energy.

In the present paper, we focus on strategies to design energy-aware routing protocols using metrics that aim to prolong the network lifetime and we illustrate the performance of these distributed algorithms using our SNF simulator. We assume that the network is randomly deployed, each node can be a sink, the sources and traffic are not known apriori and the routing is achieved without additional routing messages. Collisions and retransmissions are also taken into account and the energy model computes automatically the energy consumed according to the state of the radio.

The paper is structured as follows. Section II presents the state-of-art and the motivation behind simulating routing protocols for WSNs; Section III describes two energy-aware routing metrics and the corresponding distributed algorithms. Section IV illustrates some simulation results and unexpected behavior. Section V concludes the paper.

II. Related work and objectives

Many energy-aware routing metrics have been proposed, e.g., [1][2][3][4][5][6][7], to minimize the energy consumption and to prolong the lifetime of the network. Several routing algorithms use distance-based forwarding, where the number of hops serves as a distance metrics. Establishing reverse paths is a very used scheme [2][8][9][10]. Shortest-path routing improves the overall energy consumption since the energy needed to transmit a message from source to intended destination is correlated to the path length.

Unfortunately, shortest path (or minimum energy path) will heavily load nodes on the path and these nodes die sooner, thus creating holes or leading to disconnected networks. Various techniques to balance the load among all forwarding nodes are proposed. Some of them consider the node's residual energy to prevent nodes from choosing the same route often [6], other minimize the variance of the remaining energy between different routes [2][4][5] or use traffic spreading and aggregation as GBR [11]. The robustness to different types of failures (unreliable and asymmetric links, node failures) can be improved by multipath routing [2][9][10], where multiple paths to sink are established. In such cases the routing must incorporate packet delivery rate and link quality metrics.

Energy-efficient routing needs joint optimization with the link layer, since the only way to save energy is to *switch off* the transceiver; thus the MAC must use an active-sleep regime with a low duty cycle [12][13]. In order to study and optimize routing protocols, we use our SNF based on a cross-layer design. The simulator and the modular network architecture of a sensor node were described in [14][15].

III. Energy-aware routing

We discuss two distributed forwarding algorithms for randomly deployed WSN, involving the residual energy of nodes. Each node can be a sink and the sources and the traffic are not known apriori. The routing is achieved without employing additional routing messages and without having global knowledge about the network's topology. We assume that the sensor nodes and the communication links are reliable enough to relay the data packet along one path from source to the sink. The routing protocol consists of three phases:

P1: *Interest propagation and cost establishment* phase, where the sink broadcasts the interest. This phase includes the maintenance phase, with periodical refreshes.

P2: *Data transmission* phase, where sources send data packets, which are routed in a multihop fashion (along intermediate nodes) to the sink.

P3: *Reconfiguration* in case of transient failures.

P1) During the *interest propagation* phase **some cost** information (to the sink) must be established. Each node keeps in a gradient table one-hop candidate neighbors in the sink direction. Each network packet has a routing header consisting of several fields including the source node, the destination node, a sequence number, hop count, some energy fields and the initiator node (optional). When the request packet leaves the sink the cost field is set to 0, the energy level is set to the node's residual energy, the source and initiator addresses are set to the sink ID, the destination is set to a broadcast address and the sequence number is set to a unique number.

When a node receives the first time a request, it reads the relevant cost information and rebroadcasts a copy of the packet with the updated cost metrics. The node changes the source address and the energy level field to its node ID and its residual energy, respectively. Additionally, the node stores the sender ID and its residual energy in the gradient table (if this does not already exist). Whenever a node receives a copy of the packet leading to a smaller cost metrics, it resets its cost metrics and broadcasts again.

A node recognizes copies of the same packet by using a unique sequence number, which can be a combination of the initiator identifier and the current time (or sequence number). Each interest packet is discarded as soon as it is known (same sequence number) and does not bring any new information.

Finally, each node has determined its minimum cost to the sink and depending on the size of its gradient table, it knows a subset or all of its neighbors and their cost.

P2) The *data transmission* phase starts when a node concludes that it can deliver data matching the interest attributes. This node, referred as source, sends a data reply, which is routed to its best neighbor (the gradient) and so on hop by hop until the data reaches the sink. Each node is able to address the data packet (radio unicast) to next receiver, and only this receiver forwards the data further. The routing algorithm in each node is now able to adaptively select among several possible candidates, the one having the best cost. The routing header of the data does not change and is used in a similar way as for the interest. The initiator field is set to the source ID in order to inform the sink where the data comes from.

To let an intermediate node conclude that the data packet sent has reached the receiver, we use an implicit acknowledgment scheme. It is based on the omnidirectional radio signal property and exploits the fact that, when the intermediate node receives a packet and forwards it, the sender node can overhear the transmission and concludes (by inspecting only the header) that the transmission succeeded. If the sender node didn't hear the forwarded packet, it means the packet is possibly lost. Then the sender node either retransmits the packet up to a maximum number of retries or it sends the packet to another candidate intermediate node. In the latter case the node stores in the gradient entry of the failed intermediate information about the quality of the link, which can be used in the routing decision.

Some remarks considering both phases:

• Update routing information: To spread the routing information the (routing) protocol does not send extra routing messages but uses instead the interest to set-up routing information about the way back (return path) to sink. The routing information must be updated periodically to reflect the network state. Updates are also required to discover topology changes (i.e., new/depleted nodes) and ensure that the interest reaches all the nodes and was not lost. Therefore, the sink periodically broadcasts a copy of the original interest, referred as an interest refresh packet. Such a refresh updates cost information due to propagation failures (collision), node's energy reserve depletion or the addition of a new node. Moreover, the broadcast nature of the transmission medium allows to update some other information (e.g., node's residual energy, quality of the link) also during the data transmission phase (using the piggyback principle). The frequency of such updates depends on the sensor network application, where factors such as the numbers of sinks, the network's data traffic, and the dynamics of topology change play a decisive role.

• Adaptivity and alternative intermediate nodes: The routing protocol stores in its gradient table *several* candidate neighbors. This is advisable since in WSN, (transient) node and link failures are common; when nodes along the default route fail, providing alternative relay nodes (for each intermediate node) saves a lot of overhead, since a new route establishment process is not necessary. This enables each node to self-adapt its routing behavior to current network conditions.

Moreover, if the application requires that some critical data must reach the sink under any circumstances one can use redundancy by sending the critical data packet on more than one path (tradeoff between energy efficiency and reliability).

• Cost establishment during interest propagation: The interest and its periodical refreshes are used to spread and update the routing information into the network. Each intermediate node rebroadcasts these packets many times. If a node receives

ISBN: 978-1-61208-144-1

several copies of the interest (or refresh) consecutively, each of them leading to a smaller cost metrics, the node rebroadcasts the interest several times. Moreover, each interest copy at a node induces further updates and copies for next relay nodes. Thus, each relay node consumes more energy and excessive retransmissions can lead to congestion in the network.

The reason that a node broadcasts more than once is that it rebroadcasts immediately after receiving a lower cost, without knowing if this cost is minimal, so it may rebroadcast too early. A useful approach to this problem is to postpone the broadcast of each node by a delay time T_w in order to gain time for further cost messages. This delay time can be set to a constant value or chosen directly proportional to the cost at the node; in the latter case, a node with high cost will wait longer in order to be able to receive better costs.

- In-network processing: Aggregation is very useful to reduce the energy consumption of the nodes on the path and, additionally, it can reduce the traffic in the network avoiding in this way congestions and induced collisions.

P3) Recovery mechanisms: Since the interest is refreshed periodically (by broadcast) each node must receive a message in a given interval from its neighbors, since each neighbor rebroadcasts it at least once. But due to unreliable transmissions and collisions not all the messages reach their destination. Therefore, the interval until a message from a neighbor is expected is set larger. In our simulations, this interval was configured to three refresh rounds. Each time a message from a given neighbor arrives the timestamp of the entry is updated. If within the predefined interval the node does not receive any message (incl. SYNC frames at MAC) from a neighbor, it removes the neighbor from the cache tables.

Similarly, when a node is trying to transmit a message to a neighbor that does not react to it (acknowledgment or overhear its retransmission), the node increments a retries counter. If after a predefined number of tries the neighbor does not responds, the node concludes that it is damaged or depleted and removes it from cache. In such a case, the routing algorithm selects from its gradient table a next candidate node where to forward the data message.
New nodes are automatically inserted in the tables as soon as they rebroadcast an interest refresh.

Special mechanisms are necessary when some source becomes disconnected from the sink. In such situations the lack of the refresh, for a predefined number of rounds, stops or decreases the data generation rate at a source. Only when a new refresh is received the data generation is restarted. A similar approach can be used at the sink.
We discuss next two energy-aware strategies that we proposed in [15] and provide the corresponding distributed algorithms.

A. *The* hcE *routing strategy*

To get path information we consider a combination of the hop count and the residual energy of each node on the entire path. For that each node j computes the metrics $M(j) = \min\{M(i) + 1/E(j) | i \in Neighbor(j)\}$, where $E(j)$ is the residual energy of node j and $M(i)$ is the summation of the costs on the path from the sink up to and including node i.

Listing 1. Distributed cost establishment algorithm using the hcE metrics. Note: one can use the hop count also directly in the routing decision.

```
1  Timer timer = Timer( "Tw", BCAST_DELAY_TIMER);
2  ...
3  if ( amISink() ) { hc=0;  M=0; // metrics
4    broadcastPkt( pkt= (nodeId, hc, M) );
5  } else { hc=INFINITY; M = INFINITY; }
6  ...
7  void handleEvent( Network *pkt)
8  {
9   case RECEPTION_EVENT: // received packet from i
10   receivePkt( i, hc(i), M(i), E);
11   // Store neighbor information
12   storeUpdateNeighborInCacheTables( i, M(i), hc(i));
13   if ( hc > hc(i) + 1 ) hc = hc(i) + 1;
14   if ( M > (M(i) + cost) ) {  // Compute better M
15    M = M(i) + 1/E; nextHop = i;  // record i as relay node
16    // create new packet and schedule its transmission
17    npkt = createPkt( nodeId, hc, M, E, ...);
18    if ( timer->isScheduled()) cancelTimer( timer);
19    scheduleTimerAt( crtTime() + TWAIT * 1/E, timer); // Tw
20   }
21  case BCAST_DELAY_TIMER:  broadcastPkt( npkt ); ...
22 }
```

This additive metrics represents a quantitative characterization for the goodness of the entire route and balances the energy consumption of the network by redistributing the traffic load more uniformly on the nodes. The distributed cost field establishment algorithm is given in Listing 1.

To alleviate the problem of excessive broadcasts during flooding, caused by the fact that a node broadcasts instantly after receiving a lower cost without knowing whether this cost is minimal we introduce (in Listing 1, line 19) the waiting time T_w proportional to the cost between the receiver and sender (along the path the waiting time of each intermediate node will sum up, so at a node j T_w is proportional to $M(j)$).

The metrics used by hcE captures path information, but due to the summation it can still lead to special cases when a chosen path with small cost goes through a node with very low residual energy.

B. *The* hccE *routing strategy*

In order to avoid a situation as above we considered the hccE strategy, which uses a combined metrics involving both the hop count and the critical energy on the entire path. The intuition behind this strategy is that the bootleneck node's energy is propagated along, to be able to skip it if there are better paths (even longer ones).

Each node computes and forwards the pair: [distance to sink (in hops); critical energy on path], as $[hc(j); cE(j)] = [hc(i); cE(i)] \oplus [1; E(j)]$, where $[hc(i); cE(i)]$ is the hop count and critical energy pair corresponding to node $v = arg \min\{hc(i)/cE(i) | i \in Neighbor(j)\}$ and the operator $\oplus$ is defined for each term as $hc(j) = hc(i) + 1$ and $cE(j) = \min\{cE(i), E(j)\}$. The cost establishment algorithm is given in Listing 2.

Listing 2. Distributed cost establishment algorithm using the hccE metrics.

```
pair M = [INFINITY; E];     // E the node's initial energy
if ( amISink() ) {  M = [0; E];
 broadcastPkt( pkt= (nodeId, M, E));}
...
void handleEvent( Network *pkt ) {
  case RECEPTION_EVENT:  // received packet from node i
   receivePkt( i, M(i)=[hc(i);cE(i)], E(i) );
   storeUpdateNeighborInCacheTables( i, M(i), E(i) );
   Mcrt = add( M(i), [ 1; E] );

   if ( isNewCostBetter( M, Mcrt ) {
    M = Mcrt; nextHop = i;  // record i as relay node
    npkt = createPkt( nodeId, M, E, ...);
    if ( timer->isScheduled() )  cancelTimer( timer);
    scheduleTimerAt( crtTime()+ TWAIT*M.hc/M.cE, timer);//Tw
   }
  case BCAST_DELAY_TIMER:  broadcastPkt( npkt ); ...
 }
 pair add( pair M1, pair M2) {  // Addition of two costs
  pair Res; Res.hc=M1.hc + M2.hc; Res.cE= min(M1.cE, M2.cE);
  return Res;
 }
 bool isCostBetter( M1, M2) { // Comparison of costs
  return (M1.hc/M1.cE > M2.hc/M2.cE ); }
```

As can be seen, at each reception of a message from a node i carrying the pair $[hc(i); cE(i)]$ the receiver node records the pair in its cache, computes the new cost using the `add` method and compares it with its previous cost using the `isCostBetter` method. Hereby, we compare the ratios hc/cE for the receiver and sender node, since it is naturally to promote shorter paths (having the hop count in numerator) and paths with higher critical energy (having cE in denominator). After the comparison the (receiver) node propagates the better cost as a pair and the node that leads to the better cost is the preferred next hop. The algorithm is illustrated in Figure 1.

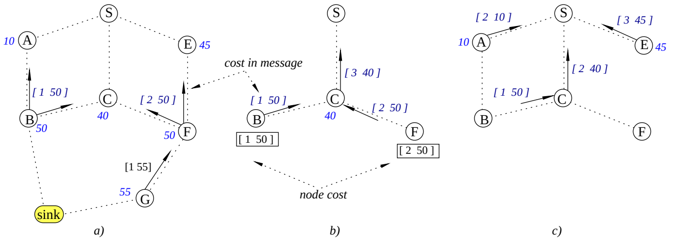


Fig. 1. Cost establishment process: a) Nodes G, B, and F send their cost; b) C receives first the packet from F and it broadcasts the cost $[3,\ 40]$; c) C receives a smaller cost from B and broadcasts again the cost $[2,\ 40]$. Nodes A and E send their cost. The source S knows three paths (along nodes A, C, E) and selects the path with the smaller cost (along C).

A node will wait for a time T_w, which is chosen directly proportional with the ratio hc/cE (computed from the pair $[hc; cE]$ of the sender an its local energy). During this period, the node computes from all received packets the minimal cost field and, if this is better than its own, it updates its local cost and resets the timer. When the timer expires, the node broadcasts the packet with its local cost. Finally some remarks:

- The strategies `hcE` and `hccE` capture path information inside their metrics. The `hcE` strategy uses an additive path cost function. Having a minimal path, it can be split (divided) in intermediate paths and all are minimal. The second metrics is not additive, but is a strictly monotonically increasing function.
- Setting the waiting time T_w depends on the underlying MAC protocol, especially if it has an adaptive duty-cycle such as the T-MAC. For the computation of T_w see §IV-B.

IV. Simulations results

We analyse the following performance parameters: the energy consumption and throughput.

We use the following general setting for the simulations. As MAC protocol we use our variant of T-MAC [16] with a listen time of $30ms$ and a frame time of $600ms$ and overhearing avoidance flag enabled (nodes in the NAV-state turn off their radio to save energy). The interest is refreshed each 5s and the simulation time is 180s. To configure the radio we used the CC2420 transmitter [17] with the following parameters:

current [mA]			**power** [mW]			
SL(sleep)	RX(receive)	TX(transmit)	SL	RX	TX	Switch
0.02	24	14	0.04	48	28	30

switching time [μs]					
$SL \rightarrow RX$	$SL \rightarrow TX$	$RX \rightarrow SL$	$TX \rightarrow SL$	$RX \rightarrow TX$	$TX \rightarrow RX$
580	580	10	10	580	580

The energy consumed according to the node's different states was multiplied with a factor of 10 to make the results sooner visible and to reduce the simulation running time.

We start with a network scenario where three sinks send a request for different network's zones and wait for data to be reported by one or several sources.

A. Impact of the routing strategy on the depletion time

We illustrate comparatively the impact of strategies on the time when the first three nodes run out of energy.
Settings: The three sinks are node 21, 24 and 41 (see Fig.2). The first two of them are placed in the bottom right corner and gather data from two zones placed on the left side, the upper one (the red rectangle) and the bottom one (the green rectangle) with three sources, respectively. The third sink, node 41, is placed on the buttom left side and gathers data from the opposite upper-right corner (the purple rectangle). Node 21 and 24 request data at each 400ms and node 41 at each 800ms (aggregation disabled) respectively. The interest is referеshed at 1s for the first sink and at 4s for the left two.

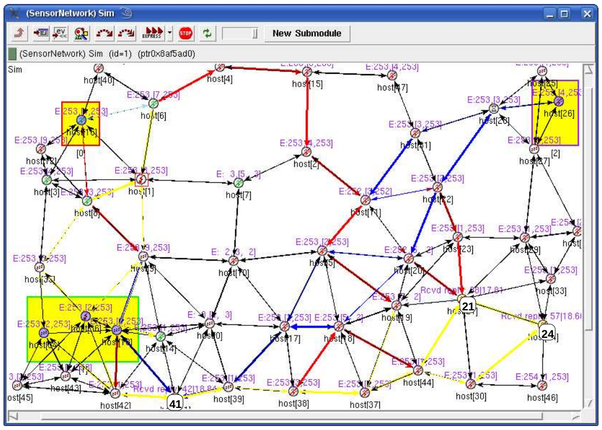


Fig. 2. Snapshot for the `hccE` strategy.

Since the routes from sources to sinks cross themselves, we set a very low initial energy for several nodes in the middle of the network: 700mJ (3eU) for node 10, and 1000mJ (5eU) for nodes 0 and 7. The energy of a node is converted in a scale between 0-255, which are called *energy units* (eU).

For space constraints we illustrate comparatively only the impact of the `hc` and `hccE` strategies on the nodes' depletion time. The `hc` strategy uses the shortest path between source and sink, meaning that the routes are along the low energy nodes (0,7 and 10). As explained in section §III-A, the `hccE`

ISBN: 978-1-61208-144-1

strategy avoids the low energy zone of the 3 nodes; the routes are either upper or lower as illustrated in Figure 2. To illustrate the routes that each data packet follows, we animated with a different color the links on which data packets are forwarded. In the figure the data packets for sink 21 travel on red paths, for sink 24 on yellow paths and for sink 41 on blue paths, respectively. Even though the three colors overwrite themselves when two data packets follow the same line (sometimes in opposite direction, e.g., node 1, 9, etc.) it is easy to identify during simulation the route that a packet follows to reach the corresponding sink.

Depletion time [s]	Nodes		
Strategy	Node 10	Node 0	Node 7
hc	42.62	53.56	73.84
hccE	54.64	63.60	85.20

Table I. Impact of high traffic and strategy on depletion time.

The depletion time result are given in Table I. The low energy nodes are sooner completely discharged in the case of hc since they are participating in forwarding the data. In the case of hccE the low energy reserve is consumed by the active phase of active-sleep regime of T-MAC. The time procentage gain is between 15% (node 7) and 28% (node 10).

We observed during simulation (with different seeds) that the hccE can occasionally route for short time a packet along a "bottleneck" node even though there are other paths. More about the causes of such a behavior in §IV-C.

B. The backoff waiting time

The metric establishment process takes place in the first phase and periodically at the rate of refreshes sent by the sink. A large number of rebroadcasts (especially when the refresh rate is high) impacts on the active time of a sensor node and the network traffic leading to a higher energy consumption. In order to overcome this problem we analyse the impact of various waiting times by using different strategies in a particular network topology (see Figure 3) with sensor nodes using an active-sleep regime.

Settings: The sink is node 21 and the rectangle zone contains one source, node 16, which generates data at each $300ms$. We set a very low initial energy for several nodes: 0.7mJ (3eU) for node 19 and 1J (5eU) for node 17. We have in this scenario paths of different length and due to low energy nodes we have various metrics depending on the chosen strategy. The simulation time is 180s and the interest refresh rate is 5s. That means that each node should broadcast at least 36 times, i.e., for 30 nodes this gives a total of 1080 times. Since the nodes 17 and 19 have very few energy, the total number of rebroadcasts is reduced to 1050 broadcasts (optimum), as these two nodes are broadcasting together no more than 30 times.

Rebroadcasts	Broadcast delay (T_w) [ms]			
Strategy	0	40	40*M	600*M
hc	1043 (5)	1044 (2)	1037 (2)	1036 (1)
hcE	1044 (4)	1043 (4)	1033 (1)	1023 (0)
hccE	1277 (3)	1252 (3)	1091 (4)	1050 (0)

Table II. Impact of T_w on the number of rebroadcasts.

The total number of rebroadcasts is given in Table II, including the number of missed refreshes (in parenthesis). To avoid side effects the source does not generate any data packets. We consider two fixed values for 0 and 40ms and two variable ones (chosen in accordance with the 30 ms listen time of T-MAC). The variable delay is achieved by multiplying the fixed delay with the metric computed by the current node as explained in section §III-A.

Note that for the hc and hcE strategies a fixed T_w has a low influence on the number of broadcasts. In the case of hccE strategy the number of broadcasts for fixed T_w is about 20% higher than for hc, as nodes rebroadcast often since here the metrics changes faster. Therefore, a fixed T_w is not recommended here and we use a variable one. When T_w is chosen proportional to the metrics (40*M, 600*M in Table II) the number of rebroadcasts improves considerable for the hccE strategy and reaches its optimum.

The same measurements with data traffic are given in Table III. Due to collisions not all rebroadcasts are received by all nodes, but the routing is not affected since all data packets reach the sink.

Rebroadcasts	Broadcast delay (T_w) [ms]			
Strategy	0	40	40*M	600*M
hc	1013 (4)	1020 (1)	1010 (4)	1010 (3)
hcE	1014 (3)	1011 (0)	1010 (2)	1011 (4)
hccE	1179 (7)	1111 (9)	1055 (7)	1043 (6)

Table III. Number of rebroadcasts with data traffic.

Thus, for a fixed broadcast delay the number of rebroadcasts is high and therefore by using hccE strategy a variable T_w larger than 40*M is recommended.

We illustrate in Table IV the impact of the broadcast delay (T_w) on the total energy consumption.

Energy [J]	without data			with data	
Strategy	0	40*M	600*M	40*M	600*M
hc	42.7	42.0	47.7	42.0	43.7
hc/E	43.8	42.7	44.2	42.9	43.4
hccE	45.1	43.5	45.3	43.8	44.5

Table IV. Energy consumption with data traffic.

As expected, for $T_w = 0$ the energy consumption for the hc strategy is a bit lower than for the hcE and hccE strategies (2.5% and 5.6%, respectively). This situation can change when using a $T_w > 0$. Although introducing an adjustable waiting time reduces the number of broadcasts, its impact in the energy consumption is not necessarily as expected in theory. This is due to the fact that nodes are spending more time in idle state, which leads to higher energy consumption.

For the hccE strategy the energy consumption decreases with a variable delay. For hc with a smaller adjustable waiting time the energy consumption decreases, but it increases for larger delay due to the T-MAC's aggressive time-out policy. Since T-MAC extends its listen period at each send/receive event, the total time the node is in idle state is longer for a 600ms delay than for a 40ms delay. This can be seen by means of our SNF when examining the transceiver states of the involved nodes (for place reason we omit the graphs here).

C. Behavior anomalies

We discuss next an example of unexpected behavior. We consider the special network scenario given in Figure 3, with the settings given in §IV-B. The depletion time of of nodes 17 and 19 are 78.04s and 70.83s for hc and 84.07s and 62s for hccE. It is expected that in the case of hccE strategy, paths along low energy node are avoided. The results show a gain of about 8% at node 17, but a lose of more than 4% at node 19. To explain this behavior we take a closer look to the simulation.

In the case of the hc the source 16 selects the minimal hop count route, thus a four hops paths, with next hop either 17 or 23 or 25. All routes are along low energy nodes (either 17 or 19). After a while both nodes are depleted, but due to different causes. Node 17 is depleted at 78.04s being a relay node in forwarding all data packets. Node 19 loses all its energy at 70.83 being passive, by following its active-sleep schedule imposed by T-MAC. After both nodes are down, the source selects the next shortest path namely 23-24-18-20-21.

In case of hccE strategy, by deferring the broadcast with a time proportional to the received metrics the optimal path 23-24-18-20-21 is not chosen. A random contention time in T-MAC (maximal 9ms) may cause that a broadcast propagates faster on a longer path than on a shorter one. Moreover, if a node goes to sleep, the broadcast is additionally delayed with the time of the sleep period. Different velocity of propagation of the broadcast, collisions, short term disconnection and transient failures are common in WSNs. The cumulative impact of all these factors is very difficult to be predicted, but using our simulation framework we identified three cases.

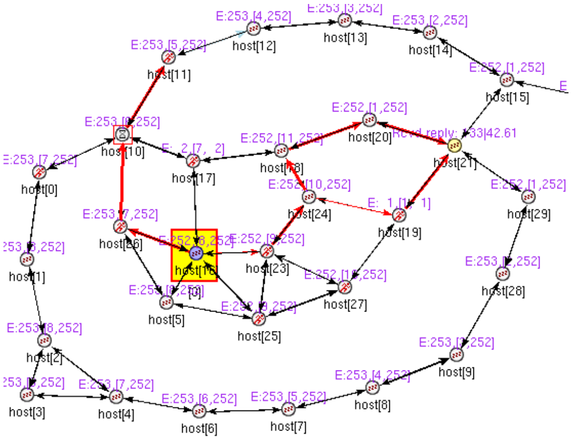


Fig. 3. Data forwarding on the path 26-10-11-12-...(lightblue arrow).

In the ideal case the minimal cost broadcast arrives in the required time or later. In the latter case the node should rebroadcast again, but the forwarding path is the optimal one.

In the suboptimal case the broadcast does not reach some relevant nodes on the optimal path between sink and source, e.g., 24 failed to receive from 18. A low energy node (19) acts as a relay node to forward the data; thus its energy decreases sooner. The source may receive a better routing cost (broadcast in the same round) from a node further from sink than the optimal one, that improves the routing information. In our scenario this happens when the broadcast on the upper path, along nodes 15-14-...-10-26, arrives and the source infers that the cost [7;252] through 26 is better than [3;2] along 17. In this way, the hccE strategy selects a suboptimal path 26-10-11-12-13-14-15-21 to forwards the data and avoids the nodes 17 and 19. For longer simulation time the lower route is selected, as soon as the metric on the upper path deteriorates.

The worst case happens when a suboptimal broadcast (of the same round) is unable to correct an already suboptimal routing information, e.g., paths along 26 are invisible since 26 failed to receive the broadcast from 10.

Even though such cases are rare, the hccE strategy cannot always avoid them and the simulation framework gives insights to find the causes for an initially unexplained behavior.

V. Conclusion

In this paper we supplemented our research [15] on energy-aware strategies randomly deployed WSNs, where multiple sinks are allowed, each node can be a sink and the sources and the traffic are not known apriori. We provided distributed routing algorithms that compute the next hop without employing additional routing messages. We further investigated the effects of introducing a broadcast delay on the number of messages and on the energy consumption. We presented simulation results that also give insights for unexpected behavior.

References

[1] W. Liang and Y. Liu, "On-line disjoint path routing for network capacity maximization in energy-constrained ad hoc networks," *Ad Hoc Networks Journal*, vol. 5, no. 2, pp. 272–285, 2007.

[2] M. Busse, T. Hänselmann, and W. Effelsberg, "Energy-efficient forwarding schemes for wireless sensor networks," in *Proc. Int. Symp. on WoWMoM*. New York, USA, June 2006, pp. 125–133.

[3] H. Hassanein and J. Luo, "Reliable energy aware routing in wsns," in *Proc. 2nd IEEE Workshop on DSSNS*, 2006, pp. 54–64.

[4] H. Nurul, M. Hossain, S. Yamada, E. Kamioka, and O.-S. Chae, "Cost-effective lifetime prediction based routing protocol for manet," in *Proc. of ICOIN*. Springer-Verlag Berlin, 2005, pp. 170–177.

[5] M. Maleki, K. Dantu, and M. Pedram, "Lifetime prediction routing in mobile ad-hoc networks," in *Proc. IEEE WCNC*. New Orleans, LA, USA, March 2003, pp. 1185–1190 (vol.2).

[6] J. Aslam, Q. Li, and D. Rus, "Three power-aware routing algorithms for sensor networks," *Wireless Comm. and Mob. Computing*, vol. 3, no. 2, pp. 187–208, 2003.

[7] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annual HICSS*. Washington, USA, 2000, pp. 3005–3014.

[8] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion a scalable and robust communication paradigm for sensor networks," in *Proc. ACM MobiCom*. Boston, MA, USA,, 2000, pp. 56–67.

[9] F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient Broadcast: A robust data delivery protocol for large scale sensor networks," *Wireless Networks, Springer, The Netherlands*, vol. 11, no. 2, pp. 285–298, 2005.

[10] F. Ye, A. Chen, S. Lu, and L. Zhang, "A scalable solution to minimum cost forwarding in large sensor networks," in *Proc. 10th Int. Conf. on Comp. Comm. and Networks*. Scottsdale, USA, 2001, pp. 304–309.

[11] C. Schurgers and M. Srivastava, "Energy efficient routing in wireless sensor networks," in *Proc. MILCOM on Comm. for Network-Centric Operations: Creating the Inform. Force*. Virginia, 2001, pp. 357–361.

[12] G. Halkes, T. Dam, and K. Langendoen, "Comparing energy-saving mac protocols for wireless sensor networks," *Mob. Netw. Appl.*, vol. 10, no. 5, pp. 783–791, 2005.

[13] T. Dam and K.Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. 1st Int. Conf. on Embedded Networked SenSys*. LA, California, USA, 2003, pp. 171–180.

[14] A. Kacsó and R. Wismüller, "A simulation framework for energy-aware wireless sensor network protocols," in *Proc. 18th Int. Conf. on Comp. Comm. and Networks*. San Francisco,CA,USA, August 2009, pp. 1–7.

[15] A. Kacsó, "Simulation of multihop energy-aware routing protocols in wireless sensor networks," *Int. Journal On Advances in Internet Technology*, vol. 3, no. 1&2, pp. 88–103, 2010.

[16] A. Kacsó and U. Schipper, "Receiver-based routing service for T-Mac protocol," in *Proc. 4th Int. Conf. on Sensor Technologies and Appl. (SensorComm)*. Venice, Italy, July 2010, pp. 489–494.

[17] *CC2420.pdf*, www-inst.eecs.berkeley.edu/∼cs150/Documents, 7.1.2011.

ISBN: 978-1-61208-144-1

Self-adaptive Localization using Signal Strength Measurements

Michal Marks*[†], Ewa Niewiadomska-Szynkiewicz*[†]
* *Institute of Control and Computation Engineering*
Warsaw University of Technology
ul. Nowowiejska 15/19
00-665 Warsaw, Poland
e-mail: ens@ia.pw.edu.pl, mmarks@ia.pw.edu.pl
[†] *Research and Academic Computer Network (NASK)*
ul. Wawozowa 18
02-796 Warsaw, Poland
e-mail: ewan@nask.pl, mmarks@nask.pl

***Abstract*—The paper treats the problem of localization in Wireless Sensor Network (WSN). In our work, we present and evaluate the localization system that can be used to calculate the geographical positions of network nodes. The search for the accurate positions of nodes is performed using a signal strength measurements and known positions of a set of selected sensors equipped with GPS system. Our scheme uses node to node distance estimates calculated based on RSSI (Received Signal Strength Indicator). The proposed solution is self-adaptive, since the transformation of RSSI measurements into distances is done automatically using information about strength of signals received by nodes equipped with GPS. We focus on the performance of our approach to localization, and discuss the accuracy of position calculation for various methods of inter-node distances estimation. The use and efficiency of the proposed localization system is illustrated by numerical examples performed in our WSN Localization Simulator.**

***Keywords*-wireless sensor networks; localization; Received Signal Strength Indicator; RSSI; signal measurements; optimization; simulated annealing; simulator**

I. Introduction

The goal of localization is to assign geographic coordinates to each node in the sensor network in the deployment area. Wireless sensor network localization is a complex problem that can be solved in different ways [1]. A number of research and commercial location systems for WSNs have been developed. They differ in their assumptions about the network configuration, distribution of calculation processes, mobility and finally the hardware's capabilities, [2], [3], [4].

Recently proposed localization techniques consist in identification of approximate location of nodes based on merely partial information on the location of the set of nodes in a sensor network. An anchor is defined as a node that is aware of its own location, either through GPS or manual pre-programming during deployment. Identification of the location of other nodes is up to an algorithm locating non-anchors. Considering hardware's capabilities of network nodes we can distinguish two classes of methods:

- range based (distance-based) methods,
- range free (connectivity based) methods.

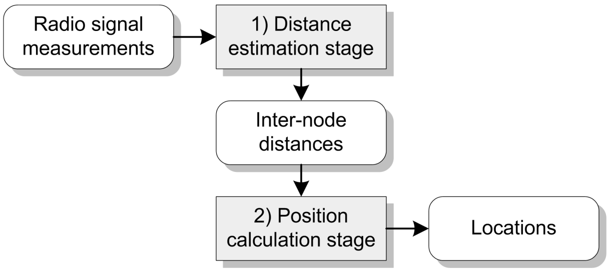

Radio signal measurements
1) Distance estimation stage
Inter-node distances
2) Position calculation stage
Locations

Figure 1. The scheme of localization process

The former is defined by protocols that use absolute point to point distance estimates (ranges) or angle estimates in location calculation. The latter makes no assumption about the availability or validity of such information, and use only connectivity information to locate the entire sensor network. The popular range free solutions are hop-counting techniques. Distance-based methods require the additional equipment but through that much better resolution can be reached than in case of connectivity based ones.

In general, to solve the distance-based localization problem it is necessary to combine two techniques: signal processing and algorithms transforming measurements into the coordinates of the nodes in the network. Hence, distance-based localization schemes operate in two stages, as shown in Fig. 1:

- *Distance estimation stage* – estimation of inter-node distances based on inter-node transmissions.
- *Position calculation stage* – calculation of geographic coordinates of nodes forming the network.

The paper is structured as follows: We formulate the localization problem in Section II. In Section III, we provide a short overview of popular radio signal measurement techniques and discuss the signal propagation modeling. In Section IV, the localization process using our localization system is described. The results of numerical experiments are summarized in Section V. In Section VI, we present conclusions.

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

II. Distance-based localization techniques

We are concerned with the distance-based approach to localization. Let us consider a WSN formed by M sensors (anchor nodes) with known position expressed as l-dimensional coordinates $a_k \in \mathbf{R}^l,\ k = 1, \ldots, M$ and N sensors (non-anchor nodes) $x_i \in \mathbf{R}^l,\ i = 1, \ldots, N$ with unknown locations. Our goal is to estimate the coordinates of non-anchor nodes. We can formulate the optimization problem with the performance measure J considering estimated Euclidean distances of all neighbor nodes

$$\begin{aligned}\min_{\hat{x}} \Big\{ J = \sum_{k=1}^{M} \sum_{j \in N_k} (||a_k - \hat{x}_j||_2 - \tilde{d}_{kj})^2 \\ + \sum_{i=1}^{N} \sum_{j \in N_i} (||\hat{x}_i - \hat{x}_j||_2 - \tilde{d}_{ij})^2 \Big\},\end{aligned} \qquad (1)$$

where $\hat{x}_i$ and $\hat{x}_j$ denote estimated positions of nodes i and j, $\tilde{d}_{kj}$ and $\tilde{d}_{ij}$ distances between pairs of nodes (k, j) and (i, j) calculated based on radio signal measurements, $N_k = \{(k, j) : d_{kj} \leq r\}$, $N_i = \{(i, j) : d_{ij} \leq r\}$ sets of neighbors of anchor and non-anchor nodes $(j = 1 \ldots, N)$, and r maximal transmission range.

The stochastic optimization algorithms can be used to solve the problem (1). Kannan, Mao and Vucetic in [5] present the results of location calculation for simulated annealing method. We propose the hybrid technique that uses a combination of the trilateration method, along with simulated annealing (TSA: Trilateration & Simulated Annealing). TSA was described in details in [6]. It operates in two phases:

- *Phase 1* – the auxiliary solution (localization) is provided using the geometry of triangles.
- *Phase 2* – the solution of the phase 1 is improved by applying stochastic optimization.

III. Range estimation

As it was mentioned in Section I using range based methods we can reach much better resolution than in case of range free ones. However in order to do that the additional equipment is usually required. Each of popular techniques – widely described in literature [2], [1] – such as Angle of Arrival (AoA), Time of Arrival (ToA), Time Difference of Arrival (TDoA) needs an additional stuff such as antennas or accurately synchronized clocks. The only exception from these requirements is a Received Signal Strength Indicator (RSSI) technique.

RSSI is considered as the simplest and cheapest method amongst the wireless distance estimation techniques, since it does not require additional hardware for distance measurements and is unlikely to significantly impact local power consumption, sensor size and thus cost. Main disadvantage of using RSSI is low accuracy. In respect to wireless channel models (Section III-A) received power should be a function of distance. However, the RSSI values have a high variability and they cannot be treated as a good distance estimates [7], [8]. On the other hand some authors indicate that new radio transceivers can give RSSI measurements good enough to be a reasonable link estimator [9], [10].

A. The radio signal propagation modeling

Propagation models are generally focused on predicting the average received signal strength at a given distance from the transmitter, as well as the variability of the signal strength in close spatial proximity to a particular location. Propagation models that predict the mean signal strength for an arbitrary transmitter-receiver separation distance are useful in estimating the radio coverage area of a transmitter and are called *large-scale* propagation models, since they characterize signal strength over large distances (hundreds or thousands of meters). On the other hand, propagation models that characterize the rapid fluctuations of the received signal strength over very short travel distances or short time durations are called *small-scale* models [11].

In this paper we concentrate on the stationary networks and do not consider small fluctuations of the signal strength in time. Hence the large-scale model is used further. Both theoretical and measurement based propagation models indicate that average received signal power decreases logarithmically with distance, whether in outdoor or indoor radio channels [11]. The mean large-scale path loss can be expressed as a function of distance:

$$PL(d)[dB] = PL(d_0)[dB] + 10nlog\left(\frac{d}{d_0}\right), \qquad (2)$$

where d is the transmitter-receiver distance, d_0 is a reference distance (for IEEE 802.15.4 radio typically the value of d_0 is taken to be 1 m) and n is the path loss exponent (rate at which signal decays). The value of n depends on the specific propagation environment and should be obtained through curve fitting of empirical data. An empirical experiment is also the best way to select an appropriate path loss for the reference distance d_0 [12].

The received signal strength P^r at a distance d is:

$$P^r(d)[dBm] = P^t[dBm] - PL(d)[dB], \qquad (3)$$

where P^t denotes the power of transmitter.

IV. Localization process

As it was mentioned in Section I our localization system operates in two stages: the distance estimation stage and the position calculation stage.

A. Distance estimation stage

The signal propagation model outlined in Section III-A allows us to estimate the distance if we know the power of received signal. Hence, in our research we used RSSI measurements. The objective of the distance estimation stage is to tune parameters of propagation model (2-3) wrt a given network technology and deployment area. Calibration procedure achieves this goal automatically – by exploiting information (pair o values: RSSI and true physical distance) obtained for the links connecting anchor to anchor node. Therefore the localization can be called self-adaptive since the algorithm is capable of calibrating own parameters without additional information about the environment.

Consider WSN with M anchor nodes with known coordinates $a_k \in \mathbf{R}^l,\ k = 1, \ldots, M$ as defined in Section II. For each pair (i, j) of anchors which is in transmission range we can measure received signal strength P^r_{ij} . The set of such pairs is as follows:

$$\Psi = \{(P^r_{ij}, d_{ij}) : ||a_i - a_j||_2 < r\}, \qquad (4)$$

where d_{ij} is known true physical distance between anchors i and j, and r transmission range.

Using (2) and (3) we can estimate the average distance between nodes i and j as a function of received signal strength P^r_{ij}:

$$\tilde{d}_{ij} = d_0 \cdot 10^{\frac{P^t - \overline{PL(d_0)}}{10n}} \cdot 10^{-\frac{1}{10n}P^r_{ij}}, \qquad (5)$$

where d_0 denotes the reference distance, $\overline{PL(d_0)}$ the path loss at the reference distance, n the path loss exponent and P^t output power of the transmitter. It should be pointed that the goal of the calibration procedure is only to predict a value of the distance d for known value of P^r_{ij}, not to find the exact value of the parameters $n, P^t, d_0, \overline{PL(d_0)}$. Hence, we can simplify the equation (5) introducing parameters α and β:

$$\tilde{d}_{ij} = \alpha \cdot 10^{\beta \cdot P^r_{ij}}, \qquad (6)$$

where $\alpha = d_0 \cdot 10^{\frac{P^t - \overline{PL(d_0)}}{10n}}$ and $\beta = -\frac{1}{10n}$. It seems to be reasonable to fit the RSSI-distance curve based on two parameters not four.

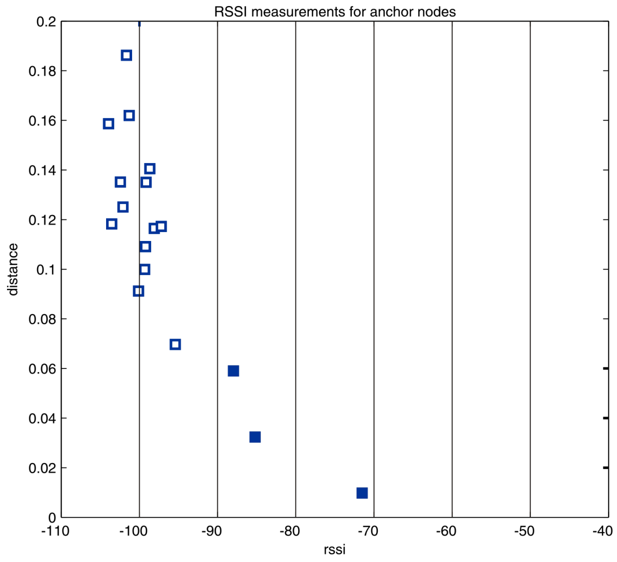
It is obvious that this average distance differs vastly from the true physical distance between selected nodes, but there is no chance to fit the curve describing signal propagation to all samples from Ψ. An ordinary least square (OLS) method can be used to calculate values of parameters α and β that minimize the error between the true physical and estimated distances:

$$\min_{\alpha_{ols},\beta_{ols}} \sum_{(P^r_{ij},d_{ij})\in\Psi} \left(\alpha_{ols} \cdot 10^{\beta_{ols}\cdot P^r_{ij}} - d_{ij}\right)^2. \qquad (7)$$

The set Ψ contains distances and RSSI measurements for anchor to anchor connections. It should be pointed here that errors caused by the signal diffraction, reflection and scattering are very high and increase with distance. For anchors distributed randomly it is very probably that they are not very close to each other and because of errors RSSI measurements are similar for different distances, see Fig. 2. In (7) all samples have the same significance (both empty and filled boxes in Fig. 2).

In order to overcome this property and improve the significance of "outliers" the weighted least square (WLS) approach was incorporated. The RSSI values scope was divided into a few ranges (indicated by a vertical lines in Fig. 2), which have the same impact on the minimized performance function. The set of anchor to anchor measurements can be given by a sum of separate subsets:

$$\Psi = \Psi_1 \cup ...\Psi_k ... \cup \Psi_n. \qquad (8)$$



$\overline{\Psi}_1$	$\overline{\Psi}_2$	$\overline{\Psi}_3$	$\overline{\Psi}_4$	$\overline{\Psi}_5$	$\overline{\Psi}_6$	$\overline{\Psi}_7$
7	7	2	1	0	0	0

Figure 2. Samples from the set Ψ for random topology.

The optimization problem for WLS approach is formulated as follows:

$$\min_{\alpha_{wls},\beta_{wls}} \sum_{\Psi_k\in\Psi} \left[\frac{1}{|\Psi_k|} \sum_{(P^r_{ij},d_{ij})\in\Psi_k} \left(\alpha_{wls} \cdot 10^{\beta_{wls}\cdot P^r_{ij}} - d_{ij}\right)^2\right]. \qquad (9)$$

Finally, in order to make calibration stage more robust the geometric combined least square method (GCLS) is proposed. In this approach parameters α_{gcls} and β_{gcls} are expressed as:

$$\alpha_{gcls} = \sqrt{\alpha_{ols} \cdot \alpha_{wls}}, \quad \beta_{gcls} = \frac{\beta_{ols} + \beta_{wls}}{2}. \qquad (10)$$

The parameters α and β obtained as a solution of optimization problems OLS (7), WLS (9) and GCLS (10) are used to transform the RSSI measurements characterizing the whole network into the matrix of appropriate distances, which is used in the next stage.

B. Position calculation stage

In the position calculation stage the measurements of inter-node distances are used to estimate the coordinates of non-anchor nodes in the network. We propose two-phase method – TSA to solve the optimization problem (1). We describe its performance in case of WSN placed on a plane.

1) Phase 1 (trilateration): In the first phase the initial localization is provided using the geometry of triangles. To determine the relative location of a non-anchor on a 2D plane using trilateration alone, generally at least three neighbors with known positions are needed. Hence, all nodes are divided into two groups: group **A** of nodes with known location (in the beginning only M anchor nodes) and group **B** of nodes with unknown location (in the beginning N non-anchor nodes). In each step of the algorithm node i from the group **B** is chosen. Next, three nodes from the group **A** that are within node i radio range are randomly selected. If such nodes exist the location of node i is calculated, node i is moved to the group **A**. Otherwise, another node from the group **B** is selected and the operation is repeated. The first phase stops when there are no more nodes that can be localized based on the available information about all nodes localization. It switches to the second phase.

2) Phase 2 (stochastic optimization): Due to the distance measurement uncertainty the coordinates calculated in the first phase are estimated with non-zero errors. In addition the position of nodes that have less than three localized neighbors can not be estimated. Hence, the solution of the first phase is modified by applying stochastic optimization method. A Simulated Annealing (SA) was considered in our research [13].

From the numerical experiments it was observed that the increased value of the location error is usually driven by incorrect location estimates calculated for a few nodes. The additional functionality (correction) was introduced to remove incorrect solutions involved by the distances measurement errors. The additional constraints were introduced to the optimization problem. The detail description of the correction algorithm can be found in [6].

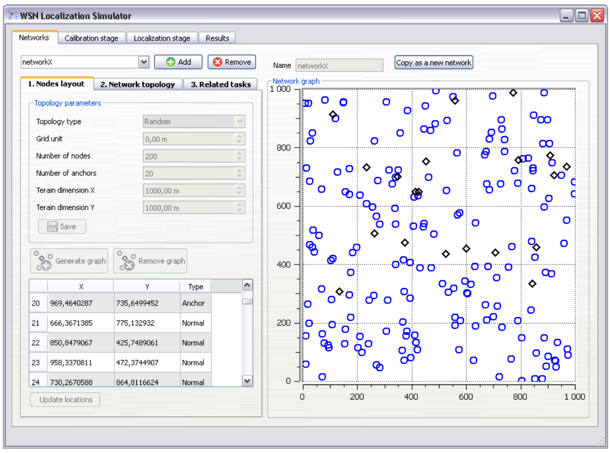
V. Experimental Results

In [6], we presented performance evaluation of TSA method in case of simplified model for distances approximation. We estimated the nodes' locations for known values of distance measurements, and focused only on localization phase. It is obvious that in real application inter-node distance have to be calculated based on radio signals measurements. Therefore, in this work we focus on self-adaptive distance calculation based on signal strengths and calibration of propagation models.

In order to evaluate our two-phase method extended by the calibration procedure many numerical tests were performed using our new software tool – *WSN Localization Simulator* (Fig. 3). All the calculations were carried out on the machine Intel Core2 Duo E6600 – 2.4GHz, 2GB RAM. The average results obtained during five runs of each localization task are presented in tables and figures. In this paper we present the results for the centralized TSA algorithm (each sensor node gather the measurements of distances between its and all the neighbors and pass them to a central station for analysis, after which the computed positions are transported back into the network).

A. Network topology generation

Network models considered in this work were created using generator built in our simulator, which is based on *Link Layer Model for MATLAB* provided by M. Zuniga and B. Krishnamachari [14]. This tool allows to generate link layer models for wireless sensor networks. In our software we focus on wireless channel modeling. No radio modulation and encoding were considered. The sample network topology – presented in Fig. 4 – was generated using parameters collected in Table I. The sensor network



networkX

Name: networkX

Topology type	Random
Grid unit	0,00 m
Number of nodes	200
Number of anchors	20
Terain dimension X	1000,00 m
Terain dimension Y	1000,00 m

	X	Y	Type
20	969,4640287	735,6499452	Anchor
21	666,3671385	775,132932	Normal
22	850,8479067	425,7489061	Normal
23	958,3370811	472,3744907	Normal
24	730,2670588	864,8116624	Normal

Figure 3. WSN Localization Simulator

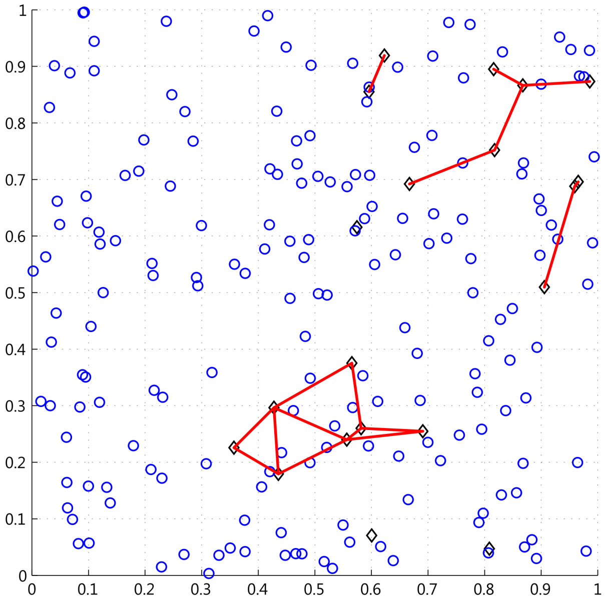


Figure 4. Sample network topology. Anchor nodes marked with diamonds and non-anchor nodes marked with circles. Connections between anchor nodes used for model calibration are marked by lines.

consisting of 200 nodes – 20 anchor nodes (marked with diamonds) and 180 non-anchor nodes (marked with circles) was considered.

B. Distance estimation stage evaluation

The comparative study of different methods of inter-node distances estimation was performed. Fig. 5 depicts the relationship between the RSSI measurements and the true physical distances for the considered task. In Fig. 5a the RSSI measurements only for anchors are presented – these data are used for signal propagation model calibration. The Fig. 5b depicts the measurements for all connections. The fitting curves obtained during calibration process are marked with the red line for the ordinary least square method, with the green line for the weighted least square method and with the blue line for the geometric combined least square method. As we can see the propagation model for the ordinary least square method

Table I
LINK LAYER MODEL PARAMETERS USED IN EXPERIMENTS

Parameter	Value
PATH_LOSS_EXPONENT	3.2
SHADOWING_STANDARD_DEVIATION	2.8
PL_D0	35.0 dB
D0	1.0 m
OUTPUT_POWER	0.0 dBm
NOISE_FLOOR	-105.0 dBm
ASYMMETRY	0 (NO)
TERRAIN_DIMENSIONS_X	1000.0 m
TERRAIN_DIMENSIONS_Y	1000.0 m
NUMBER_OF_NODES	200
TOPOLOGY	3 (RANDOM)

Table II
COMPARISON OF DISTANCE AND LOCALIZATION ERRORS FOR OLS, WLS AND GCLS METHODS USED IN CALIBRATION STAGE

Method	DE	LE
Ordinary least square (OLS)	19.22%	3.34 (0.65*)
Weighted least square (WLS)	17.65%	4.32 (0.97*)
Geometric combined least square (GCLS)	18.21%	3.95 (0.48*)

* the variance of results obtained from five runs of each task.

is prone to overestimating calculated distances between the nodes.

The differences between the true physical distances and those obtained from propagation model are presented in Fig. 6. The error is defined as $DE = |\frac{d-\hat{d}}{d}|$. It can be observed that WLS and GCLS methods effects in smaller maximum error. The differences in results of considered method increase in case of low density networks with big distances between anchor nodes.

C. Position calculation stage evaluation

The distance estimation stage produces inputs to the localization process. The estimates of inter-node distances are used to compute the coordinates of non-anchor nodes in the network. Due to the measurement uncertainty it is difficult to find a good metric to compare the obtained results. To evaluate the performance of tested algorithms we have used the mean error between the estimated and the true physical location of the non-anchor nodes in the network, defined as follows:

$$LE = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{(||\hat{x}_i - x_i||)^2}{r_i^2} \cdot 100\%, \qquad (11)$$

where LE denotes a localization error, x_i the true position of the sensor node i in the network, $\hat{x_i}$ estimated location of the sensor node i and r the radio transmission range. The localization error LE is expressed as a percentage error. It is normalized with respect to the radio range to allow comparison of results obtained for different size and range networks. The results of the localization stage are presented in Table II.

It should be underlined that the localization accuracy is more than satisfactory. The localization errors are below 5% for different calibration methods, while the mean error in distance measurements is about 20%. It means that our approach allows to obtain accurate location estimates even in case of very inaccurate distance measurements.

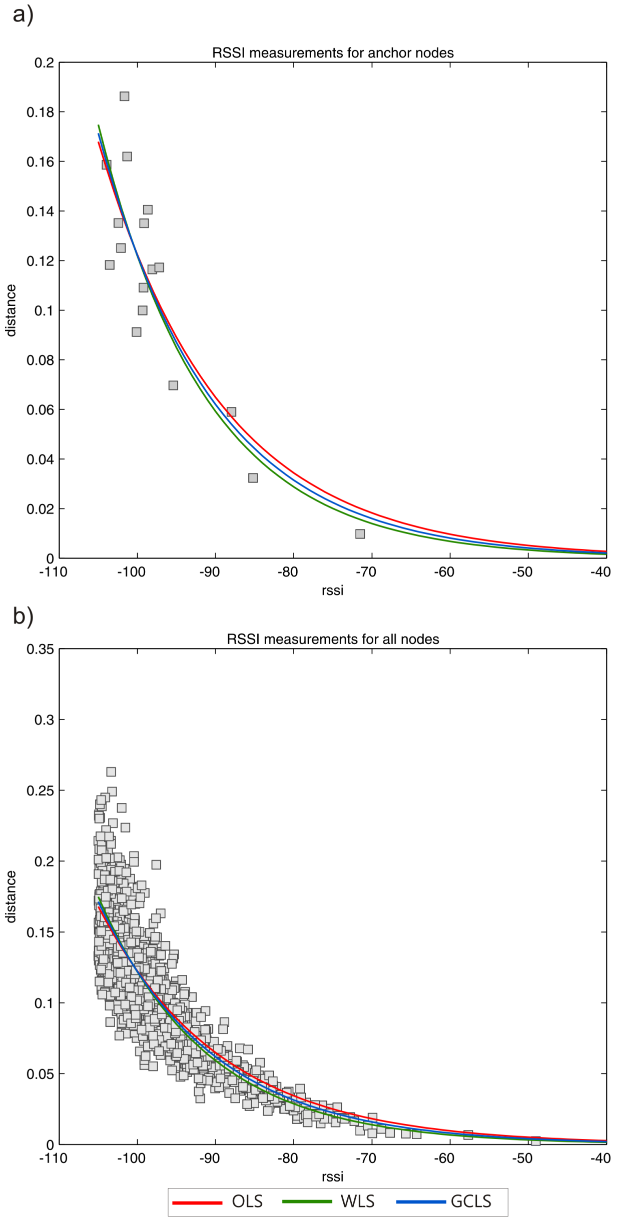


Figure 5. Distance as a function of RSSI measurements.

VI. SUMMARY AND CONCLUSIONS

We have presented the design and evaluation of our hybrid two-phase scheme for estimating the locations of nodes with unknown positions in WSN system. Emphasis was placed on the inter-node distances estimation based on received signal strength indicator. We have evaluated our algorithm, through analysis and simulation. Our evaluation demonstrates that the TSA method provides quite accurate location estimates. In our current research, we apply our algorithm to the testbed network of sensors in the laboratory.

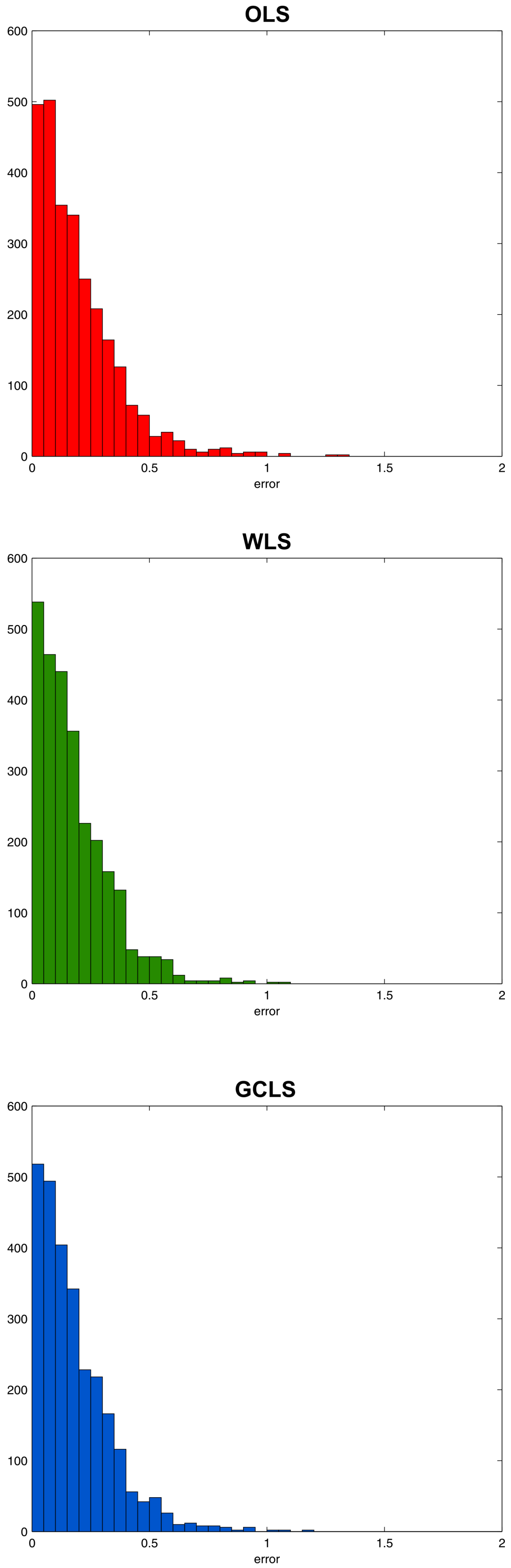


Figure 6. Distance errors histograms.

Acknowledgment

This work was partially supported by National Science Centre grant N N514 672940.

References

[1] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. Wiley, 2005.

[2] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 51, no. 10, pp. 2529–2553, 2007.

[3] A. Awad, T. Frunzke, and F. Dressler, "Adaptive distance estimation and localization in wsn using rssi measures," in *Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on*, 2007, pp. 471 –478.

[4] X. Zhang, Y. Wu, and X. Wei, "Localization algorithms in wireless sensor networks using nonmetric multidimensional scaling with rssi for precision agriculture," in *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, vol. 5, 2010, pp. 556 –559.

[5] A. A. Kannan, G. Mao, and B. Vucetic, "Simulated annealing based localization in wireless sensor network," in *LCN '05: Proceedings of the The IEEE Conference on Local Computer Networks 30th Anniversary*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 513–514.

[6] M. Marks and E. Niewiadomska-Szynkiewicz, "Two-phase stochastic optimization to sensor network localization," in *SENSORCOMM 2007: Proceedings of the international conference on Sensor Technologies and Applications*. IEEE Computer Society, 2007, pp. 134–139.

[7] K. Benkic, M. Malajner, P. Planinsic, and Z. Cucej, "Using rssi value for distance estimation in wireless sensor networks based on zigbee," in *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*, 2008, pp. 303 –306.

[8] V. Ramadurai and M. L. Sichitiu, "Localization in wireless sensor networks: A probabilistic approach," in *Proceedings of International Conference on Wireless Networks (ICWN 2003)*, Las Vegas, June 2003, pp. 300–305.

[9] P. Barsocchi, S. Lenzi, S. Chessa, and G. Giunta, "Virtual calibration for rssi-based indoor localization with ieee 802.15.4," in *Communications, 2009. ICC '09. IEEE International Conference on*, 2009, pp. 1 –5.

[10] K. Srinivasan and P. Levis, "Rssi is under appreciated," in *In Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets*, 2006.

[11] T. Rappapport, *Wireless communications: principles and practice*, second edition ed., ser. Communications Engineering and Emerging Technologies Series. Prentice Hall, 2002.

[12] J. Gibson, *The mobile communications handbook*, second edition ed., ser. Electrical Engineering Handbook Series. CRC Press, 1999.

[13] E. Niewiadomska-Szynkiewicz and M. Marks, "Optimization schemes for wireless sensor network localization," *International Journal of Applied Mathematics and Computer Science*, vol. 19, no. 2, 2009.

[14] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *In First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON*, Santa Clara, 2004, pp. 517–526.

Time Synchronization for Resource-Constrained Multi-Hop Wireless Sensor Networks based on Hop Delay Estimation

Ville Kaseva, Timo D. Hämäläinen, Marko Hännikäinen
Tampere University of Technology, Department of Computer Systems
P.O.Box 553, FI-33101 Tampere, Finland
{ville.a.kaseva, timo.d.hamalainen, marko.hannikainen}@tut.fi

***Abstract*—Ultra low-power Wireless Sensor Networks (WSNs) use duty cycled medium access protocols and dynamic routing for multi-hop communication. Transmitted packets can have variable delays and routes. Thus, data timestamps and temporal order of the packets are unknown. However, timing information is essential for many WSN applications. In this paper, we present a novel time synchronization protocol for application packets in multi-hop WSNs. The proposed protocol is based on calculating delays as packets traverse from node to node. It provides high energy-efficiency by minimizing communication overhead. Fault tolerance is achieved by fully distributed operation. The protocol computational complexity is low, including only simple operations. We experimented the proposed protocol using real WSN hardware communicating in 2.4 GHz radio band. The maximum synchronization errors ranged from 72 μs to 909 μs, and the average errors from 20 μs to 153 μs when the hop count was varied from two to six and varying protocol parameters were used. With values extrapolated from the experimental results, the maximum error was 15 ms, which occurred with 100 hops. The paper presents the design, implementation, and experimental results for low-complexity synchronization protocol with maximum errors ranging from 72 μs to 15 ms with varying network sizes.**

Keywords*-*Wireless Sensor Networks; Time synchronization.

I. Introduction

Ubiquitous computing requires small and low-cost devices embedded into everyday objects. Wireless Sensor Networks (WSNs) form a key building block in the construction of these smart environments. WSNs consist of densely deployed, independent, and collaborating low-cost sensor nodes, which are highly resource-constrained in terms of energy, processing, and data storage capacity [1]. The nodes can sense their environment, process data, and communicate over multiple short distance wireless hops. The network self-organizes and implements its functionality by co-operative effort. WSN nodes must operate for years with small batteries or by harvesting their energy from the environment. Minimizing communication is essential in achieving energy-efficiency since the radio transceiver is the most power-consuming component in a WSN node [2].

Ultra low-power WSNs use duty cycled Medium Access Control (MAC) protocols and autonomous dynamic routing. The nodes are active only a fraction of the time and the routes of subsequent packets can differ. This results in unknown sensing times, variable forwarding delays, and unguaranteed temporal order of the packets. However, many applications require that timing information of sensed phenomena can be resolved. In basic monitoring, it is usually critical to know the actual sensing times of the reported values. Furthermore, e.g., inferring target velocity from a series of proximity detections [3] requires that the order of (inferring) the sensed events is known.

The most straightforward way for obtaining accurate time information would be to equip every node with a Global Positioning System (GPS) receiver, Universal Time Coordinated (UTC) signal receiver or an accurate atomic clock. However, this would be infeasible in WSN nodes due to increased size, cost, and energy consumption. Although effective in the Internet, the widely used Network Time Protocol (NTP) [4] is too inflexible for WSNs with ad-hoc operation [5].

Commonly, WSN synchronization protocols target at achieving common time locally among a set of neighboring nodes or globally to the whole network. In these, the synchronization protocol operation is fully de-coupled from application-level operation resulting in continuous common time keeping among the nodes. Thus, there is constant synchronization overhead and the overhead is also included in nodes that do not require time information. In these protocols, the time service quality is common to every node and application. This may result in unnecessary accuracy and overhead for some applications, and redundant synchronization overhead during periods time information is not actually needed.

In this paper, we present a delay-based time synchronization (D-DYNC) protocol for application-level time resolution in multi-hop WSNs. The proposed protocol is based on calculating delays as packets traverse from node to node. D-SYNC provides following key benefits compared to related protocols:

- **High energy-efficiency:** D-SYNC does not require explicit messaging to achieve synchronization. The delay information is piggybacked in application packets. The timing accuracy can be adjusted with simple delay field shift operations according to application requirements. This adjustment minimizes the packet delay field size,

ISBN: 978-1-61208-144-1

and thus, the transmitted overhead data while providing required accuracy. Synchronization can be switched on on-demand only for required time periods, e.g., when a specific event is sensed. These characteristics minimize the communication overhead, and thus, energy consumed by the radio.
- **Fault tolerance:** D-SYNC is fully distributed having no dedicated time reference nodes nor single points of failure. In dynamic WSNs, there may be periods when nodes are not in the radio range of any other nodes in the network or multiple re-transmissions are required due to poor links. D-SYNC can still maintain timing for the buffered packets and the buffers can be unloaded as connectivity is re-established.
- **Minimal computational complexity:** The complexity of the D-SYNC protocol is low including only timestamp value saving, delay calculation using subtraction, and shifting for timing accuracy adjustment.

We present a mathematical analysis and prototype experiments for the D-SYNC protocol. The mathematical analysis provides tools for estimating the maximum achievable accuracy and the required minimum delay field width for variable WSN deployments. We experimented the D-SYNC protocol with real resource-constrained WSN nodes. Accuracy results are presented for varying hop and delay field shift amounts. The used hardware is typical for resource-constrained WSN nodes having a simple 8-bit microcontroller with 2 MIPS performance for processing and a low-power low-cost 2.4 GHz radio for communication. The radio frame length is 32 B requiring minimal protocol overhead for maximizing application payload data content in the packet.

The rest of this paper is organized as follows. The related work is presented in Section II. The D-SYNC design and mathematical analysis are presented in Section III. Section IV introduces the used prototype hardware platform. The experiments and results are presented in Section V. Finally, Section VI concludes the paper and present the future work.

II. Related Work

Next, we present the most essential multi-hop time synchronization protocols proposed for WSNs surveyed, e.g., in [5] and [6]. The protocols represent fundamental approaches to clock synchronization [6]. We compare them against D-SYNC benefits listed in the previous section.

The Timing-sync Protocol for Sensor Networks (TPSN) [7], the Lightweight Tree-based Synchronization (LTS) protocol [8], the Delay Measurement Time Synchronization (DMTS) protocol [9], and the TSync protocol [10] use a dedicated time reference node, a tree structure, and periodical synchronization messages forwarded throughout the tree. In these, the periodical synchronization message exchanges inflict continuous energy consumption overhead to the nodes. Fault tolerance against lost time reference node is not considered at all or is mitigated by using several reference nodes, which are chosen using a leader election algorithm. In either case, single points of failure, the reference node(s), exist in the network.

LTS and TSync provide also a de-centralized version where nodes can query time information from a reference node. In these, the messaging overhead is large since the queries have to be first forwarded to the reference node, possibly via multiple hops, and then synchronization is achieved by reversing the path of the query.

The Flooding Time Synchronization Protocol (FTSP) [11] floods synchronization messages, originating from a reference node, periodically through the network inflicting significant continuous energy consumption overhead. There is no single point of failure since any node in the network can act as the time reference.

In Reference-Broadcast Synchronization (RBS) [12], clock parameters are exchanged with every neighboring node. This method incurs significant messaging overhead and energy consumption penalty, which increases with the density of the nodes. RBS reduces the overhead with on-demand post-facto synchronization in situations where continuous synchronization is not not needed. Still, when continuous synchronization is required, a constant messaging overhead occurs also in RBS.

The protocol presented in [13] achieves multi-hop synchronization using a tree structure. The periodical synchronization message exchanges inflict continuous energy consumption overhead to the network. Also, since clock parameters are exchanged with every neighboring node, with which synchronization is required, the overhead increases with the synchronized neighbor amount.

The Time-Diffusion synchronization Protocol (TDP) [14], and the protocol proposed by Li et al. [15] achieve network wide time by co-operative effort of all the nodes in the network. The complexity of both protocols is high requiring lot of messaging [5]. The protocols do not rely on a single reference node increasing fault tolerance.

III. D-SYNC Design

The D-SYNC protocol is based on calculating cumulative delay for a packet as it traverses via a multi-hop network. Each synchronized packet includes a delay field, which is updated at every hop. Every node forwarding a synchronized packet adds the local delay incurred by that node using its local clock. Using this delay value included in the packet, a destination node can resolve the source event time in its local time base.

A. Single-Hop Delay Calculation

The undeterministic delays during a packet exchange are the main contributors for synchronization inaccuracy. The total delay incurred by one packet exchange can be divided into send, access, transmission, propagation, and receive delays [16]:

- The *send delay* includes the time spent at the source node to construct the packet, the delays incurred by the operating system, the buffering delays including re-tranmsissions, and the time required to transfer the packet to the MAC layer for transmission. The send delay is undeterministic and highly variable.
- The *access delay* includes the time spent waiting for access to the wireless communication medium. It depends on the used MAC protocol. The access delay is undeterministic and can be highly variable.
- The *transmission delay* (t_{tx}) refers to the time it takes to transmit a single packet over the wireless medium. It is directly proportional to the length of the transmitted packet and inversely proportional to the radio data rate making it relatively deterministic. This delay also includes deterministic and undeterministic radio specific delays. An example of a deterministic radio delay is the radio start-up transient time, during which the radio hardware is powered up.
- The *propagation delay* changes as a function of distance. For radio waves it is below 1 μs for distances under 300 m. The ns scale error caused by it does not contribute significantly to the error budget of synchronization in WSNs [12] and is considered to be zero.
- The *reception delay* consists of the time it takes for the radio receiver to process the incoming bits and notify the host of packet arrival. Commonly, much of the physical packet processing is done in the radio hardware and a packet is delivered to the host via a digital interface. Thus, this delay is relatively small. The undeterministic components can be mainly contributed to the small variability in interrupt latencies.
- The *receive delay* includes the time required to transfer the packet to the receiving layer. It includes the delays incurred by the operating system and the buffering delays. The receive delay is undeterministic and highly variable.

The operation principle for calculating single hop delay in the D-SYNC protocol is illustrated in Figure 1. A packet is triggered for transmission at time t_0 at the source Node i. The time value is saved to internal packet data structure in the local time of Node i, $L_i(t_0)$, where $L_i(t_k)$ denotes real time t_k in the time base of node i. After t_0, the packet is constructed and scheduled for transmission. Just prior to transmission, at time t_1, the delay value is calculated using the previously saved and current time values. This delay is inserted to the packet at the MAC protocol or the radio driver (depends on implementation specifics). The calculated delay value is $L_i(t_1) - L_i(t_0)$. Now, the packet is transferred to the radio transceiver hardware, which transmits it after a radio specific delay. Calculating the delay just before transmission mitigates the highly variable send and access delays.

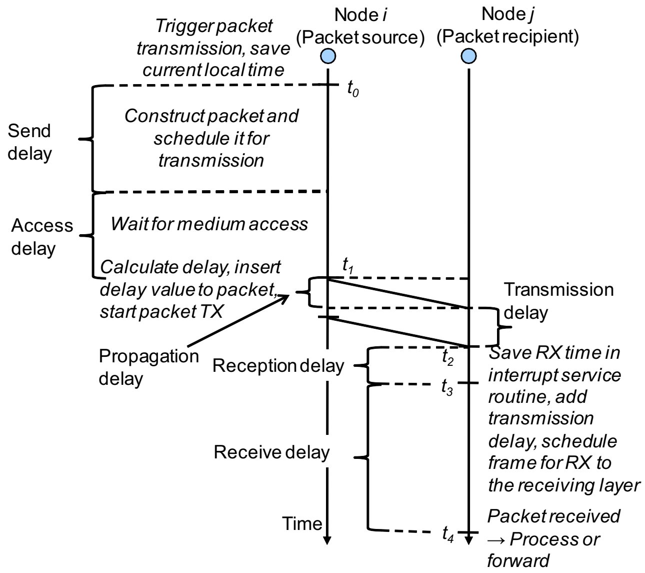

Trigger packet transmission, save current local time
Node *i* (Packet source)
Node *j* (Packet recipient)
t_0
Send delay
Construct packet and schedule it for transmission
Access delay
Wait for medium access
Calculate delay, insert delay value to packet, start packet TX
t_1
Transmission delay
Propagation delay
Reception delay
t_2
t_3
Save RX time in interrupt service routine, add transmission delay, schedule frame for RX to the receiving layer
Receive delay
Time
t_4
Packet received → Process or forward

Figure 1. Example of single hop delay calculation. The total delay incurred by one packet exchange can be divided into send, access, transmission, propagation, and receive delays. By calculating delay values just prior to transmission, the highly undeterministic delays can be mitigated and delay estimation accuracy improved.

Furthermore, the time $L_i(t_1)$ is again saved to the internal packet structure in case the packet is not actually received by the next hop node and re-transmission is needed. In the case of re-transmission the additional delay can be calculated and added to the current delay value using the saved time value $L_i(t_1)$ and the new transmission moment of the packet.

After the transmission delay (t_{tx}) and the propagation delay, the packet is received by the recipient Node j at time t_2. After interrupt latency, the recipient saves the reception time of the packet $L_j(t_3)$. This saved value can be again used for internal delay calculation at node j. Saving the time value just after reception mitigates the highly variable reception and receive delays. Furthermore, the transmission delay is added to the packet delay at this stage due to successful packet exchange. Thus, the only remaining inaccuracies are the propagation delay and the possibly undeterministic timing inflicted by the radio hardware.

B. Multi-Hop Delay Calculation and Time Resolution

To calculate the multi-hop delay, the single hop delay calculation is performed at every hop and the resulting delays added to the delay field in the packet. This means that also the inaccuracies accumulate hop-by-hop but are minimized by low-level timestamping. An example of the multi-hop delay calculation is presented in Figure 2. Node 1 schedules packet for transmission at time t_0 and saves time value $L_1(t_0)$ to the internal packet data structure. The physical packet transmission to the next hop (Node 2) is started at at time t_1. Thus, calculated delay from packet

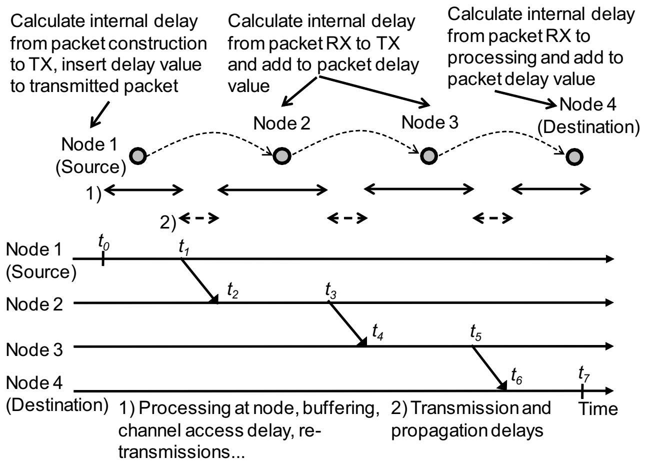

Calculate internal delay from packet construction to TX, insert delay value to transmitted packet
Calculate internal delay from packet RX to TX and add to packet delay value
Calculate internal delay from packet RX to processing and add to packet delay value
Node 1 (Source)
Node 2
Node 3
Node 4 (Destination)
1)
2)
Node 1 (Source)
Node 2
Node 3
Node 4 (Destination)
t_0
t_1
t_2
t_3
t_4
t_5
t_6
t_7
Time
1) Processing at node, buffering, channel access delay, re-transmissions...
2) Transmission and propagation delays

Figure 2. To calculate the multi-hop delay, the single hop delay calculation is performed at every hop and the resulting delays added to the delay field in the packet.

construction to transmission, inserted to the packet at Node 1, is $L_1(t_1) - L_1(t_0)$.

A packet is received at Node 2 at time t_2 and the time value $L_2(t_2)$ is saved. Also, the transmission time (t_{tx}) of the previous wireless packet exchange is added to the total delay at this stage. The physical packet transmission to the next hop (Node 3) is started at at time t_3. Thus, calculated delay from packet reception to transmission, added to the packet delay field at Node 2, is $L_2(t_3) - L_2(t_2) + t_{tx}$. Similarly, the added delay at Node 3 is $L_3(t_5) - L_3(t_4) + t_{tx}$.

Finally, the packet is received at the destination, Node 4, at time t_6. Again, the packet reception time is saved as the local time of Node 4, $L_4(t_6)$. The final delay, from reception to actual processing moment, added at Node 4, is $L_4(t_7) - L_4(t_6) + t_{tx}$.

Thus, the remove event time (t_0), as estimated by the destination Node 4 at time t_7 in its local time base, is $L_4(t_7) - t_{delay}$, where t_{delay} is the cumulative delay value obtained from the received packet. Since all the delay calculations are done locally no continuous synchronization nor time base conversions are required.

In general, the remote event time for packet k ($t_{pkt(k)}$) in the time base of destination node i is

$$L_i(t_{pkt(k)}) = L_i(t_{pr}) - t_{delay(k)}, \tag{1}$$

where t_{pr} is time the packet is processed at the destination, and $t_{delay(k)}$ is the cumulative delay value for packet k obtained from the packet.

C. Accuracy and Resource Consumption

There is a tradeoff between the maximum achievable accuracy and the amount of used bits for synchronization messaging (overhead). These can be be varied by simple shift operations. When the delay field is shifted to the right the least significant bits are lost but most significant bits are gained. This results in lower accuracy but less overheard bits and larger maximum delay value.

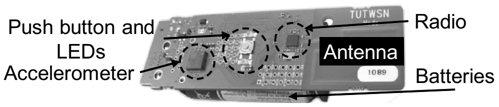


Figure 3. Prototype hardware platform circuit board.

Thus, the delay field overhead can be minimized when the scale of required accuracy, maximum number of hops in the WSN, and the worst case hop delay can be estimated. On the other hand, the shifting enables graceful degradation of accuracy at run-time. With a fixed delay field width, the field overflow can be monitored hop-by-hop and additional shifting can be used when required.

With shift amount of S bits, the maximum achievable accuracy in ticks (ε) is given by

$$\varepsilon = 2^S \tag{2}$$

and the maximum presentable delay in ticks (D) is

$$D = 2^{N_{bits}+S} - 1, \tag{3}$$

where N_{bits} is the number of bits in the delay field.

E.g., for a WSN where maximum number of hops is 20, the worst case hop delay is 10 s, and the internal clock tick in the nodes is 1 μs, the maximum achievable accuracy values and presentable delays with varying delay field widths are as follows.

With an 8-bit delay field the required shift amount is 20 bits resulting in maximum accuracy of 1 s and maximum presentable delay of 268 s. The corresponding values for 16-bit and 24-bit delay fields are shift of 12 and 4 bits, and maximum accuracy of 4 ms and 16 μs, respectively. For 16-bit and 24-bit delay fields the maximum presentable delay is 268 s with the given shift values. With a 32-bit delay field, no shift is required, maximum accuracy is 1 μs corresponding to one clock tick, and the maximum presentable delay is 4295 s.

IV. Prototype Hardware Platform

The prototype hardware platform is presented in Figure 3. The platform uses a Microchip PIC18F8722 MCU, which integrates an 8-bit processor core with 128 kB of FLASH program memory, 4 kB of RAM data memory, and 1 kB EEPROM. The used clock speed of the MCU is 8 MHz resulting in 2 MIPS performance.

For wireless communication the platform uses a Nordic Semiconductor nRF24L01 radio transceiver operating in the 2.4 GHz ISM frequency band. The radio data rate is 1 Mbps and there are 80 available frequency channels. Transmission power level is selectable from four levels between -18 dBm and 0 dBm with 6 dBm intervals and ±4 dBm accuracy. Loop type antenna is implemented as a trace on the Printed Circuit Board (PCB).

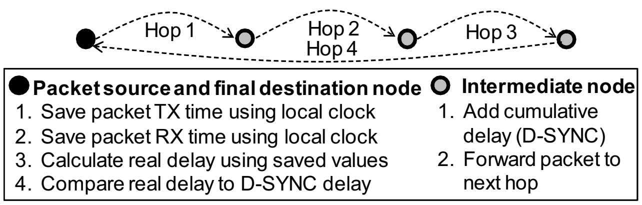

Hop 1
Hop 2
Hop 4
Hop 3
● Packet source and final destination node
1. Save packet TX time using local clock
2. Save packet RX time using local clock
3. Calculate real delay using saved values
4. Compare real delay to D-SYNC delay
○ Intermediate node
1. Add cumulative delay (D-SYNC)
2. Forward packet to next hop

Figure 4. Example experimental scenario for 4 hops. The packet traverses a loop and source node local clock is used as a reference.

The platform is equipped with a 32.768 kHz crystal for real-time clock implementation. This results in clock resolution of approximately 30 μs. The resolution is enhanced by using the MCU's own oscillator and delay loops for fine timing below 30 μs.

V. Experiments and Results

For the experiments, the D-SYNC protocol was implemented on the prototype hardware platform. For communication, a multi-hop protocol stack implementation with fixed routing and fixed one second delay per hop was used. A ring topology was used to implement scenarios with varying amount of hops. An example of the topology with 4 hops is illustrated in Figure 4. The scenarios are general as the path traversed by a single packet can always be reduced to a simple chain of nodes independent of the actual network topology used.

The experimental scenarios consisted of 2, 4, and 6 hops. Each of the three scenarios was experimented with no shift, giving a time resolution of 1 μs, and with shift of 8 bits, giving a time resolution of 256 μs. For each test, the packet was transmitted 100 times, of which maximum delay errors and average delay errors were calculated. It must be noted that the experimented scenarios represent a subset of possible scenarios that can occur in a WSN. The number of hops, the length of channel access delays, and the required shift amount can have variations. These are inherently handled by the D-SYNC protocol. The presented experimental setup provides a well-defined and reproducible framework for estimating the orders of magnitude of achievable accuracy. Furthermore, it gives total control over the scenario parameters.

In the experiments, the source node was set to be also the final destination node letting the transmitted packet traverse a loop. Upon transmitting a packet, the source node saved the initial packet transmission time and after reception the reception time. Using these times a reference delay was calculated. The intermediate nodes forwarded the packet and added the cumulative delay value given by the D-SYNC protocol. The reference time value was compared to the value given by the D-SYNC protocol for calculating accuracy results.

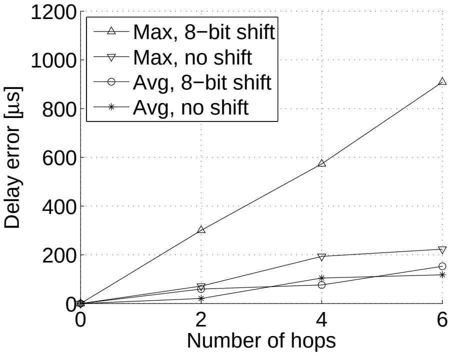


Figure 5. D-SYNC delay error as a function of hop count.

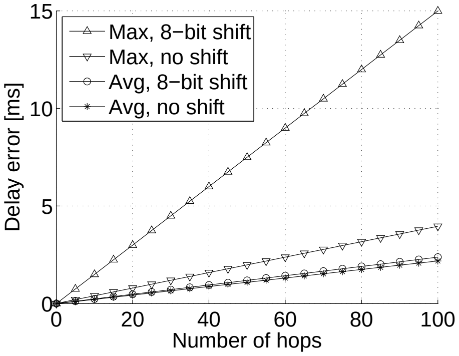


Figure 6. D-SYNC delay error results with least squares linear fitting and extrapolation to 100 hops.

The results are presented in Figure 5. With shift of 8 bits, the maximum synchronization errors were 300 μs for 2 hops, 573 μs for 4 hops, and 909 μs for 6 hops. The corresponding values with no shift were 72 μs, 194 μs, and 223 μs, respectively. With the 8-bit shift, average errors were 60 μs, 76 μs, and 153 μs with the given hop amounts. The corresponding average values with no shift were 20 μs, 105 μs, and 118 μs, respectively.

The results show an almost linear increase in delay as the hop count increases. Figure 6 presents least squares linear fit of the results extrapolated to 100 hops. Using these values the synchronization error can be estimated beyond the experimented 6 hops. With shift of 8 bits, for 25, 50, 75, and 100 hops, the estimated maximum errors are 3.7 ms, 7.5 ms, 11.2 ms, and 15.0 ms, respectively. The corresponding values with no shift are 1.0 ms, 2.0 ms, 3.0 ms, and 4.0 ms. With the 8-bit shift, for 25, 50, 75, and 100 hops, the estimated average errors are 0.6 ms, 1.2 ms, 1.8 ms, and 2.4 ms, respectively. The corresponding values with no shift are 0.5 ms, 1.1 ms, 1.6 ms, and 2.2 ms.

The main error source in the experiments was the real-time clock of the used prototype platform. The usage of delay loops for fine timing produces inaccuracies when clock resolution is below 30 μs. First, this incurs error in the actual delay calculation. Furthermore, the same clock is used at the radio driver. This causes errors in the packet transmission times and packet reception time registering.

The results show that the shifting has an obvious impact on the maximum synchronization errors, which accumulate when the hop count increases. However, the difference between the average synchronization errors between the 8-bit shift and no shift is much smaller. With the experimented four hops the average synchronization error was even smaller with the 8-bit shift compared to no shift. The surprisingly small difference in the average synchronization errors can be explained by the fact that with the 8-bit shift the maximum error fluctuated between positive and negative and was almost equal to both directions whilst with no shift the maximum error was almost always positive.

VI. Conclusion and Future Work

In this paper, we presented D-SYNC time synchronization protocol for multi-hop WSNs. The proposed protocol is based on multi-hop delay calculation for synchronized packets. High energy-efficiency is achieved by minimizing communication overhead. Accuracy versus overhead tradeoff can be adjusted using simple shift operations. Timing is maintained on-demand removing redundant overhead when synchronization is not needed. D-SYNC operation is fault tolerant having no single points of failure and supporting disconnected periods. The protocol complexity is low, including only timestamp value saving, delay calculation using subtraction, and shifting for timing accuracy adjustment.

We presented a mathematical analysis and tools for estimating the D-SYNC accuracy and overhead tradeoff. D-SYNC was implemented on real resource-constrained WSN nodes and its accuracy was experimented with variable hop and shift amounts. The maximum synchronization errors ranged from 72 μs to 909 μs, and the average errors from 20 μs to 153 μs when the hop count was varied from two to six and the shift amount was varied for zero to eight bits. With values extrapolated from the experimental results, the maximum error was 15 ms, which occurred with 100 hops and 8-bit shift. The experiments show that the errors have a predictable change when the hop amount increases and the error remains in the ms-scale even for large networks.

Our future work concentrates on clock drift compensation in the delay calculation. Synchronized MAC protocols rely on accurate local synchronization among neighboring nodes. This information can also be used to derive clock drift values for the D-SYNC protocol.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Communications Magazine, IEEE*, vol. 40, no. 8, pp. 102–114, 2002.

[2] M. Kohvakka, J. Suhonen, M. Kuorilehto, V. A. Kaseva, M. Hannikainen, and T. D. Hamalainen, "Energy-efficient neighbor discovery protocol for mobile wireless sensor networks," *Elsevier Ad Hoc Networks*, 2008.

[3] A. Cerpa, J. Elson, M. Hamilton, J. Zhao, D. Estrin, and L. Girod, "Habitat monitoring: application driver for wireless communications technology," in *SIGCOMM LA '01: Workshop on Data Communication in Latin America and the Caribbean*. New York, NY, USA: ACM, 2001, pp. 20–41.

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

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

[6] Y.-C. Wu, Q. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *Signal Processing Magazine, IEEE*, vol. 28, no. 1, pp. 124 –138, 2011.

[7] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. New York, NY, USA: ACM, 2003, pp. 138–149.

[8] J. V. Greunen and J. Rabaey, "Lightweight time synchronization for sensor networks," in *WSNA '03: Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*. New York, NY, USA: ACM, 2003, pp. 11–19.

[9] S. Ping, "Delay measurement time synchronization for wireless sensor networks," Intel Research Berkeley Lab, Tech. Rep. IRB-TR-03-013, june 2003.

[10] H. Dai and R. Han, "Tsync: a lightweight bidirectional time synchronization service for wireless sensor networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, no. 1, pp. 125–139, 2004.

[11] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. New York, NY, USA: ACM, 2004, pp. 39–49.

[12] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, 2002.

[13] M. Sichitiu and C. Veerarittiphan, "Simple, accurate time synchronization for wireless sensor networks," in *WCNC '03: Proceedings of the IEEE Conference on Wireless Communications and Networking*, March 2003.

[14] W. Su and I. F. Akyildiz, "Time-diffusion synchronization protocol for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 384–397, 2005.

[15] Q. Li and D. Rus, "Global clock synchronization in sensor networks," *IEEE Trans. Comput.*, vol. 55, no. 2, pp. 214–226, 2006.

[16] H. Kopetz and W. Schwabl, "Global time in distributed realtime systems," Technishe Univesität Wien, Tech. Rep. 15/89, 1989.

A Routing Protocol for WSN Based on the Implementation of Source Routing for Minimum Cost Forwarding Method

Fardin Derogarian
MAP-Tele, INESC Porto
Faculdade de Engenharia da Universidade do Porto
Porto, Portugal
mpt09020@fe.up.pt

João Canas Ferreira[1], Vítor M. Grade Tavares[2]
INESC Porto
Faculdade de Engenharia da Universidade do Porto
Porto, Portugal
[1]jcf@fe.up.pt, [2]vgt@fe.up.pt

Abstract—**This paper presents a routing protocol for wireless sensor networks (WSN), established on the basis of fundamental concepts in source based routing (SBR) for *ad hoc* networks and minimum cost forwarding (MCF) methods for heterogeneous WSNs. Neither routing tables nor network topology information is maintained at sensor level, which makes the proposed protocol part of the reactive routing protocols class. Despite the lack of network information at the sensor, the packets from the sink node to sensors, and vice-versa, always follow the optimal communication path with minimum cost. Simulation results have shown that the proposed protocol performs better than MCF protocol alone, and nodes always route the packets through the optimal path up to destination. In fact, according to the energy consumption and throughput found by simulation, this protocol improves on the MCF protocol for applications where the sink node, acting as a server or base station (BS), generates significant amounts of network traffic. All results are based on simulations and data treatment performed with OMNet++ 4, Matlab 7 and Microsoft Visual Studio2010(C#) platform tools.**

Keywords-Wireless Sensor Networks; Minimum Cost Forwarding; Source Based Routing,

I. Introduction

A wireless sensor network (WSN) comprises a large number of sensors equipped with wireless communication ports that are deployed closed or within the phenomenon to be monitored. Recent advances in wireless communication electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes. Small in size and capable of communications over short distances, this emergent technology has opened a wide range of application possibilities. Usefulness can be found in a panoply of areas, such as health, military, industrial and home applications [1][2]. Usually, in a sensor network, sensors cooperate to handover data from the source sensor to the destination. In most systems, a single sink node is responsible for collecting data from all sensors. Still, in numerous situations, this sink node also is a BS node to manage the sensors.

WSNs are *ad hoc* networks, employing techniques for network self-organization and packet routing [3]. However, there are many fundamental differences between the traditional wireless *ad hoc* networks and WSNs *per se*, which makes conventional wireless *ad hoc* network protocols unsuitable for WSN applications. Large number of sensors, proneness to failure, fast changing network topology, limited resources and low-power consumption are examples of such dissimilitude found in WSNs. The literature describes numerous protocol designs targeting specific WSN applications [1][2][3]. Sensor networks are limited-resource systems, therefore a significant amount of effort has been directed to reduce the size of the network part, overall power consumption and to the design of protocols that take these characteristics into consideration.

Routing protocols are classified in two general categories: proactive and reactive protocols [1]. Proactive routing protocols keep track of routes to all destinations in routing tables. LEACH [4], a protocol based on node clustering and PEGASIS [5], a protocol based on a token-passing chain, are two examples of proactive routing protocols. Unlike proactive protocols, reactive protocols acquire routes on demand and avoid saving information about the network topology. Flooding, Gossiping and MCF [6] [7] are examples of reactive protocols.

Traffic in sensor networks displays, in general, a heterogeneous nature [8]. In fact, in most cases, the communication patterns in sensor networks are characterized by:

a. Traffic between the BS node and sensor nodes. This type of traffic has two sources: 1) sensor nodes sending acquired data to the BS node (BS as a sink node); 2) BS node sending control information to the sensor nodes (for configuration of measurement parameters, for example). This type of traffic represents the largest part of the overall communication.
b. Traffic between adjacent nodes: adjacent nodes exchange information data for data transmission, get the conditions, connections, topology and etc.

These conditions need to be considered during the design of a network protocol for sensor networks. This paper proposes a reactive routing protocol where sensors have no information about the network topology, but packets from sensors to BS or vice-versa, always communicate over optimum paths with minimum cost. Since the proposed concept combines source routing with minimum cost forwarding, it is called the Source Routing for Minimum Cost Forwarding (SRMCF) protocol. In this approach, the routing information of the packets generated by the BS to be sent to sensor nodes is included in the packets.

The rest of the paper is organized as follows: Section 2 describes the related works. Section 3 describes the main protocol and section 4 explains the network initialization procedure. Section 5 presents and discusses simulation results from the proposed protocol. Section 6 unites the main conclusions drawn from this work.

II. Related Works

The MCF protocol is a good method for routing packets in a reactive sensor network [6][7]. This routing method is a cost field based approach and exploits the fact that the routing direction of data, flowing from sensors to sink, is always known and that cost is always minimum. In this method, sink node starts to setup the network with broadcasting its cost value and all nodes get minimum cost value to reach the sink node. With this method, sensor nodes have neither routing tables nor information about the network topology.

It is observable that this approach applies only for data sent from the sensor nodes to sink. If the sink node wants to send data to a specific node, other methods like flooding must be employed. In situations where the BS node simultaneously acts as a sink and server, and generates a significant amount of data, then implosion, overlapping and resource blindness problems, resulting from the flooding method, will reduce the network performance. Therefore, MCF is appropriate only for those applications where the sink node has an almost exclusive role of data collector.

For the BS to send data to a dedicated sensor, destination and routing path must be defined at the BS node like in source based routing (SBR) [9]. To implement source routing, the packet contains the address of each node on the routing path. Source routing requires determining the address of all nodes and routing paths from source to destination, as is done in protocols like Dynamic Source Routing (DSR) [9][10] for wireless *ad hoc* network and Link Quality Source Routing (LQSR) [11] developed by Microsoft for wireless mesh networks. DSR and LQSR protocols are reactive approaches and do not need routing tables. These protocols determine a route on-demand when the source node wants to send data to destination node and keep the routing information while communicating.

The source node establishes a route between source and destination nodes by broadcasting a RouteRequest packet. When the destination node receives the RouteRequest packet, it replies with RouteReply packet to the source node. This packet carries the routing path from source node to destination node. During the communication between the nodes, the intermediate nodes route the packets by using the routing information which is carried in the packet headers.

A higher connection setup delay in comparison with table-driven protocols and the absence of a mechanism for local repair of failed links are some of the disadvantages of the DSR and LQR protocols.

III. Description of the Protocol

Consider a wireless network composed of multiple sensor nodes and one BS node. The BS node maintains a table of minimum cost paths from itself to every sensor node in the network. If the BS node needs to send a packet to a given sensor node, over a specific path specified in the table, the intermediate nodes must be aware of the path and route the packet to the correct links. As aforementioned, sensors in a reactive network do not have any information about the network topology. Furthermore, it is impossible to route the packet over a predefined fixed path when nodes have no knowledge about network topology or routing information. However, if the packet carries the path information, like it is done in Trajectory Based Forwarding (TBF) [12] and DSR, then the intermediate nodes can use this information to route the packets to destination node.

Taking into account the heterogeneous traffic in a WSN and making use of minimum cost forwarding and source based routing concepts, a reactive protocol can be designed to have optimum routing in both communication directions (from BS to nodes and nodes to BS).

It should be noted that in this method there is only one routing table at the BS node: the other nodes use the information in that table when the BS node issues a route-packet. The routing of packets, originated from sensor nodes, is based on the minimum cost forwarding method, without resourcing to a routing table. It is necessary that nodes can identify the type of a packet, because the routing algorithms for packets, coming from the BS node and for packets generated by sensor nodes, are different. These algorithms are described below.

A. *Packets Sent from BS to Sensor Node*

Suppose that the BS node needs to send a packet to sensor N3 in Fig. 1. In a mesh network, there are many paths from BS to each node, but almost always there is only one optimum path that has minimum cost for forwarding packets. Suppose that the minimum cost path, between BS and sensor node N3, is the one shown in bold in Fig. 1. In the present protocol there is a routing table at the BS node that maps each sensor node ID to the minimum cost path from BS to sensor node. This table is formed during the network setup phase.

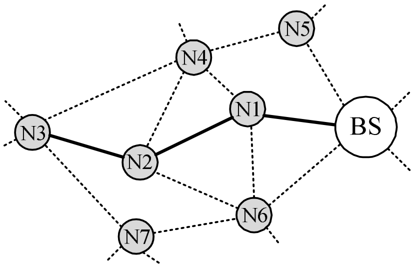


Figure 1. The minimum cost path between the BS and sensor N3

Fig. 2 depicts the proposed format for packets generated by the BS. The packet header includes three fields for routing purposes: a pointer, an offset and path information. The pointer determines the position in the information path for the next node. Each sensor node will decrease it by one unit before sending the packet to the next node. When the Pointer reaches zero, it means that the current node is the destination.

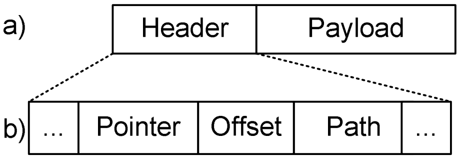


Figure 2. a) Packet generated by BS, b) Header of the packet

The packet header is variable in length and depends on the number of the nodes between the BS and destination nodes. The offset determines the length of the path: the destination node will use it to determine the start position of payload, while intermediate nodes can ignore it.

As an example, table I shows the paths from BS to nodes N2, N3 and N4 as presented in the routing table at the BS. The value saved for each node is an ordered list of intermediate node IDs.

TABLE I. ROUTING PATH FOR NODES N2, N3 AND N4 IN ROUTING TABLE

Node	Path
…	...
ID2	ID1
ID3	ID1, ID2
ID4	ID5
…	…

Fig. 3 shows the evolution of the pointer value as the packet passes through different nodes on the path form BS to N3. To send a packet from BS to N3, BS generates a packet with path (ID1, ID2, and ID3) and pointer value 2. When N1 gets the packet from BS, the pointer value is 2. N1 decreases the pointer by one and sends the packet to N2 (since ID2 was the node ID in position 2 as specified by the pointer). When arriving at N2, the packet pointer is 1: therefore the packet will be sent to N3 with new pointer value 0. N3 gets the packet when the pointer is 0: this means that N3 is the destination node.

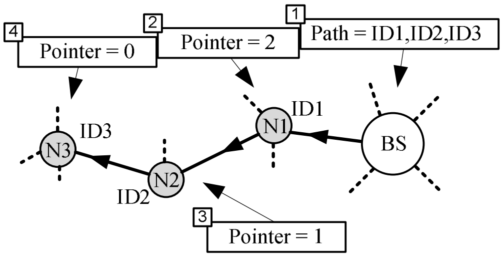


Figure 3. The pointer value in different nodes on the path between BS and N3

It can be seen that with this method sensor nodes can route packets without having information about the destination node and with minimum processing. All the information that they need to select the next node is available in the header, and sensors only select the next node based on the pointer value and the ID list present in the header.

B. *Packets from a Sensor Node to BS:*

Suppose now that the sensor N3 in Fig. 1 needs to send a packet to the BS node. As will be described later, during the setup phase of the network, to each node is assigned a minimum cost value and the ID of that adjacent node, on the path to the BS that has minimum cost. In this example, assume that the minimum cost neighbour is N2 for N3 and N1 for N2. Then N3 generates a packet that includes the N3 ID and sends it directly to N2. When N2 has a packet that must be sent to the BS node, it will send it to N1. In this example, N1 will send the packet to the BS directly. The received packet includes ID3 as the identification of the source node.

It is clear that the packet header is different for packets sent from the BS node and for those originated from the sensor nodes. In the latter, there is no information about the present path and the size of the header is fixed. Intermediate nodes decide how to handle each packet based on its type.

IV. NETWORK SETUP

Before normal operation, the network must be initialized. The setup phase has two steps. During the first all nodes determine their cost values for communicating with the BS node. During the second step the BS node generates the routing table. The setup processing is as follows.

A. *Determination of each Node Cost Value*

This step is similar to the minimum cost forwarding back-off process [6], however, differently from MCF; each node now has a unique ID. First all nodes except the BS, set their cost to infinity. The cost can be of any parameter such as hop count, transmission power, consumed energy, processing resources or delay. The BS node associated cost is zero. The BS broadcasts a cost advertisement message to the adjacent nodes. When a sensor node receives a cost message, compares its present cost with the new cost plus the link cost. If the new total cost value is less than its previous cost, the node changes the cost to the new value and saves the sender ID responsible for the advertisement message. The node then broadcasts an advertisement message to the adjacent nodes with its new cost value and ID. This process continues until all the nodes set their cost values to the minimum and introduce themselves to the BS. From the standpoint of the BS, one node does not exist unless it has introduced itself to BS. The BS node has to wait for the setup of the network to finish. The waiting time is set according to the number of nodes and network parameters such as link speed, delay and processing time.

In Fig. 4 example, node N3 has two links with N2 and N4, but the cost value from N2 is lower than the cost value from N4. Therefore, N3 will change its cost to 5 and register

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

the ID of N2 as being on the path with minimum cost to the BS node.

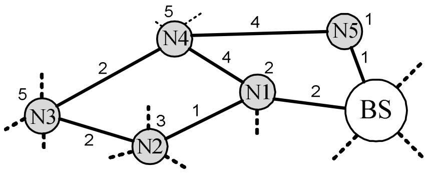


Figure 4. Forwarding along minimum cost

B. Routing Table Creation in the BS Node

As mentioned before, there is only one routing table for the whole network at the BS node. This table has information about all optimum paths with minimum cost values between the BS node and other nodes. The table creation step proceeds as follows:

When a node changes its cost value to a new value (during the setup of the network or even during normal operation of the system), it sends a message with its ID to the adjacent node from which it had calculated its own cost value. The receiver node adds its ID to the received message and sends it back to the next adjacent node along the optimum path. Eventually, when the BS receives the message, it has a message form the source node that includes the IDs of the nodes in the path between the source node and server node. The server node (BS) will save the IDs as a routing path in the row of the routing table, corresponding to that particular source node. This way, sensor nodes and sink node collaborate in the creation of the routing table. It should be noted that the same process will be performed during the normal network operation if a cost value, of a given node, changes. The cost value of nodes can change when a link or node failure occurs or still when a node gets a cost advertisement message with a lower cost value than their previous cost.

Fig. 5 shows the routing path creation for node N3, supposing that N1 and N2 are the nodes with minimum cost value to BS node, on the path between the N3 and BS. Table I shows the value of that row belonging to N3 with ID3 in the column “Node” of the routing table.

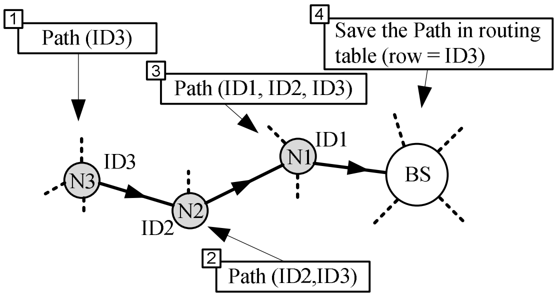


Figure 5. The cost value of N3 has changed

If a link or node failure occurs during normal operation, the cost value of the nodes and their related routing path in BS must be updated. For example, suppose that the link between N3 and N2 in Fig. 4 fails, so that the previous cost value of N3 is not valid anymore. Then N3 starts the process of getting a new cost value by changing its cost value to infinity and sending a cost request message to adjacent nodes. In this case, it obviously gets a new minimum cost value from N4. Now the new path with minimum cost between N3 and BS goes through N1 and N4. After the change in cost value, the routing path related to N3 in the routing table of BS is updated as mentioned in step B. Note that the cost value of the nodes located closer to the BS than the failing link are not affected. The process for updating the cost value after a node failure is similar.

V. SIMULATION AND RESULTS

In the above sections the routing protocol and the setting up of the network were exposed. This section presents the simulation results and the respective performance of a WSN using the proposed protocol. The results are compared with one employing the MCF protocol, and were obtained by implementing both protocols with OMNet++ 4. Matlab 7 and Microsoft Visual Studio 2010 with C# were also used to create the network and help on the analysis of the data from OMNet++.

The sensors were randomly scattered in a square area and remain fixed throughout the simulation. Table II shows the simulation parameters.

TABLE II. SIMULATION PARAMETERS

Parameter	**Value**
Sensor nodes	50, 100
Network area (m2)	100×100
The averages size of the packets generated with nodes (byte)	150
Packets maximum length (byte)	256
Antenna reach (m)	10
Processing delay (ms)	1
Nodes buffer size (byte)	1k
Simulation time (s)	30

The evaluation metrics are network throughput and energy consumption in terms of packet generated by BS node, when all the sensor nodes and BS node simultaneously generate packets. Another item in analysis is the average packet header size created by the BS node in terms of the number of sensor nodes in the network.

Fig. 6 shows the throughput of networks with 50 and 100 nodes using the proposed protocol. For comparison, the throughput for the same network, using the MCF protocol, is also shown. Results were collected for different amounts of traffic generated by the BS node. The results show that due to the use of optimum path information, during communication, SRMCF achieves higher throughput than MCF. Furthermore, for the range analysed, the throughput is almost constant with increasing data-rate.

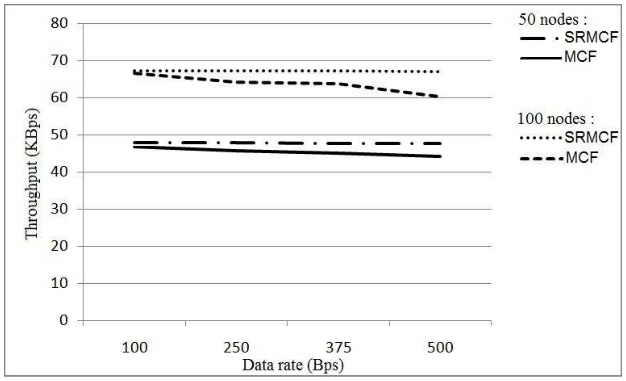


Figure 6. Throughput of the network in terms of the data rates of the packets generated by the BS node

In fact, with the proposed protocol, the traffic generated by the BS node is similar to the traffic of the other nodes. In contrast, the MCF protocol floods the packets from BS to nodes and increases the unwanted traffic in the network. Usually, increasing the traffic augments the probability of collisions and consequent packet loss, decreasing the network throughput.

Fig. 7 shows the network energy dissipation plot for the SRMCF and MCF protocols and for various values of generated packets by BS node. The simulation results are for 50 and 100 nodes.

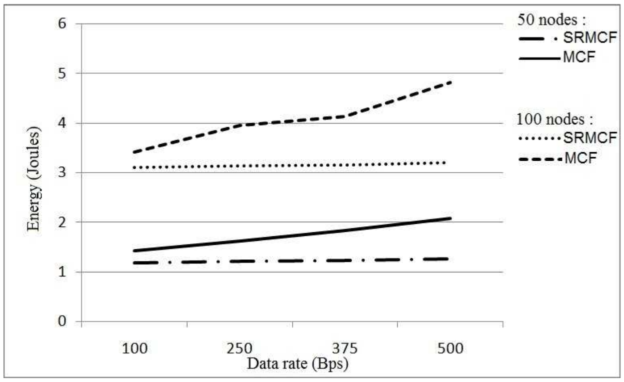


Figure 7. Energy consumption of the network.

The energy consumption during data transmission depends on distances between the nodes [13] and usually is in the range of a few nano joules per each communication bit in WSN applications (10.8 nJ/bit reported in [14]). After the network setup is finished, each node starts to send packets to the BS node, at 1kBps data rate. The BS node randomly selects nodes and sends packets to them. It can be seen that, for the same conditions, the MCF protocol leads to a larger energy dissipation that furthermore increases faster with increasing data-rate. This is consequence of the flooding method used by MCF to send packets from BS to the nodes. In contrast, the proposed protocol sends the packets directly from BS to the node over the optimum path.

The SRMCF packet header size, generated by BS, is variable and depends on the number of nodes in the paths between the BS and destination nodes. When the BS needs to send data to adjacent nodes uses a minimum length of 5 bytes for the header. Fig. 8 shows the average header size for networks with 100 to 1000 nodes. The figure implies that although the number of nodes increases 10 times, the average header size increases only by a factor of 2.26.

The effect of a variable header size has been taken into account in the energy dissipation simulations by considering the energy dissipated for each. In our simulations, the maximum packet size is 256 byte, so a one–byte filed is enough to specify the packet length. Both SRMCF and MCF have a fixed 5-byte header for packets generated by the sensor nodes. Both protocols have a relatively small header size in relation to the overall packet size.

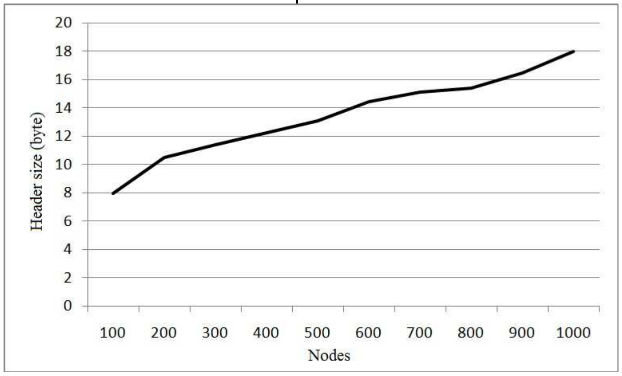


Figure 8. Packet header size

VI. CONCLUSION

This paper describes a routing protocol for wireless sensor networks based on the inclusion of routing information in the packets when minimum cost forwarding method is used. With the proposed protocol, and except for the BS node, there is no need to maintain explicit forwarding path tables in the intermediate nodes. The routing table on BS is formed in the network setup phase and updated after any change in network topology reported by sensor nodes. The intermediate nodes get routing information from the packets originating from the BS without having to know the network topology. In comparison with the MCF protocol, the traffic from sensor nodes to BS is the same, but the traffic from BS node to sensor nodes achieves better performance without significant changes on the sensor nodes side.

The simulation results indicate that not only the proposed protocol has higher throughput than MCF, but also dissipates less energy.

ACKNOWLEDGMENT

This research was partially funded by Fundação para a Ciência e a Tecnologia (FCT) under contract number PTDC/EEA-ELC/103683/2008 (PROLIMB) with Institute for Systems and Computer Engineering of Porto (INESC Porto).

REFERENCES

[1] M. Ilyas and I. Mahgoub, “Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems,” CRC Press LLC, USA, 2005.

[2] X. LI, “Wireless Ad Hoc and Sensor Networks, Theory and Applications,” Cambridge University Press, USA, 2008.

[3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” Communications Magazine, IEEE, vol. 40, 2002, p. 102–114.

[4] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on, 2002, p. 10–pp.

[5] W. Guo, W. Zhang, and G. Lu, “PEGASIS Protocol in Wireless Sensor Network Based on an Improved Ant Colony Algorithm,” 2010 Second International Workshop on Education Technology and Computer Science, Wuhan, China: 2010, pp. 64-67.

[6] F. Ye, A. Chen, S. Lu, and L. Zhang, “A scalable solution to minimum cost forwarding in large sensor networks,” Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on, 2001, p. 304–309.

[7] T.V. Padmavthy, G. Divya, and T.R. Jayashree, “Extending network lifetime in wireless sensor networks using modified minimum Cost forwarding protocol-MMCFP,” Wireless Communications, Networking and Mobile Computing, 2009. WiCom’09. 5th International Conference on, 2009, p. 1–4.

[8] C. Ma and M. Ma, “Data-centric energy efficient scheduling for densely deployed sensor networks,” Communications, 2004 IEEE International Conference on, 2004, p. 3652–3656.

[9] Yingji Zhong and Dongfeng Yuan, “Dynamic source routing protocol for wireless ad hoc networks in special scenario using location information,” in Communication Technology Proceedings, ICCT 2003. International Conference on, 2003, vol. 2, pp. 1287-1290 vol.2.

[10] J.E. Garcia, A. Kallel, K. Kyamakya, K. Jobmann, J.C. Cano, and P. Manzoni, “A novel DSR-based energy-efficient routing algorithm for mobile ad-hoc networks,” Vehicular Technology Conference, VTC 2003-Fall. IEEE 58th, 2003, p. 2849–2854.

[11] Microsoft Research, "Self-Organizing Neighborhood Wireless Mesh Networks", http://research.microsoft.com/mesh.

[12] D. Niculescu and B. Nath, “Trajectory based forwarding and its applications,” Proceedings of the 9th annual international conference on Mobile computing and networking, 2003, p. 260–272.

[13] W. Ye and F. Ordónez, “Robust Optimization Models for Energy-Limited Wireless Sensor Networks under Distance Uncertainty,” Wireless Communications, IEEE Transactions on, vol. 7, 2008, p. 2161–2169.

[14] B.H. Calhoun, D.C. Daly, N. Verma, D.F. Finchelstein, D.D. Wentzloff, A. Wang, S.H. Cho, and A.P. Chandrakasan, “Design considerations for ultra-low energy wireless microsensor nodes,” IEEE Transactions on Computers, 2005, p. 727–740.

Cooperative Sensor Relocation in a Mobile Sensor Network by Distributed Subgradient Algorithm

Vladimir Marbukh, Kamran Sayrafian-Pour
Information Technology Laboratory
National Institute of Standards and Technology
E-mails: {marbukh, ksayrafian}@nist.gov

Hamid Mahboubi, Ahmadreza Momeni, Amir G. Aghdam
Department of Electrical and Computer Engineering
Concordia University, Montreal, QC, H3G 1M8, CA
Emails: {h_mahbo, a_momeni, aghdam}@ece.concordia.ca

***Abstract*—Collaborative sensor relocation has potential to improve Mobile Sensor Network (MSN) performance while prolonging its life span by conserving sensor battery energy. However, lack of centralized control, variety of performance criteria, numerous uncertainties, and a possibility of sensor trapping in sub-optimal positions make collaborative sensor relocation an exceedingly challenging problem. Assuming that the sensing and communication operations are optimized much faster than sensors relocate, this paper proposes a sensor relocation strategy based on iterations of a Distributed Subgradient Projection Algorithm (DSPA). While maintaining inherently distributed nature of the computations, DSPA-based sensor relocation is capable of aligning sensor mobility with the overall MSN goals. This approach also accounts for a possibility of abrupt topology changes as sensors relocate.**

Keywords—mobile sensor network, controlled mobility, distributed subgradient algorithm.

I. INTRODUCTION

Mobile Sensor Networks (MSN) are envisioned to offer a novel set of applications in detecting, monitoring and tracking people, targets or events in pervasive computing environments [1]. Locations of sensors in a MSN affect both their ability to acquire information on the intended target(s) and event(s) as well as their ability to communicate this information to the intended recipient(s). The information acquisition needs, which require proximity to the target(s), often compete with the communication needs, which require proximity to the recipient(s) of the sensor information. Inherent traits of MSN such as lack of centralized control, variety of performance criteria, operational uncertainties, and possibilities of MSN topology change and sensor trapping in suboptimal locations make MSN optimization an exceedingly challenging problem.

Our previous publications [2]-[4] proposed aligning sensor mobility with the overall MSN goals by assuming that (a) sensing and communication are optimized much faster than sensors relocation and (b) sensor relocations are governed by cost/benefit analysis where "cost" of sensor battery energy expenditure for sensor relocation is weighted against projected performance gains due to new sensor locations. This approach is vastly superior to sensor mobility control based on *phenomenologically* defined potential fields and the corresponding virtual forces [5]. This is because dissipation of the non-renewable sensor battery energy, asymmetric virtual forces due to asymmetric wireless channels, and abrupt changes in the optimal MSN topology with sensor relocation are inconsistent with the existence of global potential field.

Practicality of the framework proposed in [2]-[3] depends on overcoming numerous challenges with (inherently) distributed nature of MSN being the most critical. An intelligent sensor may have direct knowledge of its current life expectancy determined by its battery energy level and depletion rate affected by the surrounding terrain as well as sensor information acquisition and transmission capabilities. However, a sensor typically has no direct knowledge of the effect of its relocation on the rest of the MSN.

This paper suggests that the class of Distributed Subgradient Projection Algorithms (DSPA) [6]-[7] has potential for addressing major challenges of controlled sensor mobility in MSN. Subgradient-based iterations allows for dynamic network topology optimization. Projection of the algorithm iterations onto the set of feasible sensor locations ensures sensor information acquisition and communication needs. Most importantly communication overhead reducing techniques allow for addressing the inherently distributed nature of MSN. While more conventional pricing-based algorithms effectively reduce the communication overhead in some particular situations, recently emerged consensus-based algorithms have low communication overhead in a more general setting.

Consensus-based algorithms differ in their approach to achieving consensus. In the first type [6] each agent maintains and iterates its own sequence, and consensus is achieved by communicating this sequence to the neighboring nodes, who incorporate neighbors' sequences into their own sequences through averaging. In the second type [7] all agents update a single sequence and consensus is achieved by passing the sequence instances to each other. In this paper, we concentrate on the first type since updating and passing instances of a single sequence by agents to each other exposes algorithm to the risk of manipulation by malicious agents while averaging with judiciously chosen weights can mitigate this risk. Comparison and suitability of a particular consensus-based algorithm for controlling sensor mobility is left for a future study.

A major difficulty with controlling sensor relocation is the possibility of sensor trapping in suboptimal locations (e.g. in non-flat terrain) due to typical non-convexity of the performance criterion [8]. A possible approach to overcoming this difficulty is allowing occasional random sensor relocations to escape the potential traps in the spirit of simulated annealing optimization algorithm [9]-[10]. One of the advantages of the proposed approach in this paper is in its ability to incorporate such random moves in a distributed way.

The rest of this paper is organized as follows. Section II quantifies the effect of sensors positions on MSN performance; while the effect of sensors relocation in discussed in section III. Section IV describes sensor relocation algorithms based on maximization of the corresponding performance gain. Section V discusses initial simulation results showing benefits of cooperative sensor relocation for the case of a MSN tracking a single target on a flat terrain. Finally, section VI briefly summarizes the proposed approach to controlled sensor mobility and outlines directions for future research.

II. MSN PERFORMANCE

We consider a Mobile Sensor Network (MSN) comprised of sensors/nodes $s \in \mathbf{S} = \{1,..,S\}$, which acquire and communicate information to a single destination formally identified as node $s=0$. The MSN topology is $\Gamma = (N, L)$ with set of nodes N and set of links L. Here, we assume that our goal is to maximize the MSN life span, given the constraints on sensors ability to acquire and communicate sensor information at some low fixed rate and sensor battery energy availability. Subsection A describes requirements on MSN ability to acquire and communicate information to the destination. Subsection B quantifies effect of sensor locations on MSN performance.

A. Sensing and Communication Requirements

Requirements on sensor ability to acquire and communicate information can be incorporated into performance optimization either through penalty for the corresponding energy expenditure or directly in terms of feasible sensor locations. In a wireless interference-limited network, capacity c_l of a link $l = (i, j)$ from node i to node j depends on the transmission power, channel condition, and node locations through the Signal-to-Interference Ratio:

$$SIR_{ij} = p_{ij}\xi_{ij} \Big/ [\eta_j + \textstyle\sum_{(n,k)\neq(i,j),n\neq i,j} p_{nk}\xi_{nj}], \qquad (1)$$

where path gain $\xi_{ij} = \xi_{ij}(x_i, x_j)$ depends on the locations of the link (i, j)'s end-points x_i and x_j, and the noise power $\eta_j = \eta_j(x_j)$ at the receiver located at x_j.

Specific form of channel capacity $c_{ij} = c_{ij}(SIR_{ij})$ as a function of SIR_{ij} depends on the modulation and coding schemes. We assume a threshold-based channel model: $c_l(SIR_l) = c$ if $SIR_l > \chi$ and $c_l(SIR_l) = c$ otherwise, where $c, \chi > 0$ are some constants. We also assume that the interference from simultaneous transmissions by different sensors is negligible as compared to the noise at the receiver:

$$\textstyle\sum_{(n,k)\neq(i,j),n\neq i,j} p_{nk}\xi(x_n, x_j) << \eta(x_j), \qquad (2)$$

Thus, Signal-to-Interference Ratio on the link (i, j) is a function of the transmission power on this link:

$$SIR_{ij} \approx p_{ij}\xi(x_i, x_j)\big/\eta(x_j). \qquad (3)$$

For a threshold-based channel model the minimal transmission power for node s on an active link $l = (s, j) \in L$ is:

$$\breve{p}_{sj}(x_s, x_j) = \chi_{sj}\,\eta(x_j)\big/\xi(x_s, x_j). \qquad (4)$$

The minimal total transmission power required for sensor s to send information at low rate to nodes $i : (s,i) \in L$ is

$$\breve{p}_s = \sum_{i:(s,i)\in L} \chi_{sj}\,\eta(x_j)\big/\xi(x_s, x_j) \qquad (5)$$

where the set of nodes active links L determines the network topology. In the case of free-space propagation:

$$\xi_{ij} = \zeta_{ij}\left\|x_i - x_j\right\|^{-\gamma} \qquad (6)$$

where ζ_{ij} and γ are positive constants, and $\left\|x_i - x_j\right\|$ is the Euclidian distance between sensors i and j with coordinates x_i and x_j respectively.

B. MSN Utility

Following [8] we assume that sensor $s \in \mathbf{S}$ utility of preserving battery energy until moment T can be quantified by utility function $u_s(T/T_s^*)$ where T_s^* is the corresponding target. Monotonously increasing functions $u_s(z), z > 0$ have an S-shape and steeply increases around $z \approx 1$. A convenient approximation for functions $u_s(z)$ is

$$u_s(z) = A_s\big/[1 + e^{-(z-1)/a_s}], \qquad (7)$$

where parameters $a_s, A_s > 0$ affect importance of conserving sensor battery energy as compared to other considerations, e.g. high information rates. At moment t, sensor s remaining battery energy level is $E_s = E_s(t)$ and the battery energy draining rate is $p_s = p_s(t)$. Then, the projected battery energy depletion time is $T_s \approx t + E_s/p_s$, and thus the corresponding utility is

$$u_s(T_s/T_s^*) = u_s[(t + E_s/p_s)/T_s^*]. \qquad (8)$$

Given sensor locations $\mathbf{x} = (x_s, s \in \mathbf{S})$ and sensor battery energy levels $\mathbf{E} = (E_s, s \in \mathbf{S})$, we quantify MSN performance by the aggregate utility:

$$W(t, \mathbf{x}|\mathbf{E}) = \sum_s w_s(t, \mathbf{x}|E_s) \qquad (9)$$

where

$$w_s(t, \mathbf{x}|E_s) = u_s\left(\frac{t + E_s/\breve{p}_s(\mathbf{x})}{T_s^*}\right) \qquad (10)$$

For simplicity we further assume that power draining rates $\breve{p}_s = \breve{p}_s(\mathbf{x})$ are only due to communication and thus are given by (5). We also assume that requirements on sensor ability to acquire information are formalized directly in terms of sensor locations, e.g., with respect to the tracked target(s) or with respect to their ability to cover the area of interest.

III. RELOCATION GAIN

Subsection A quantifies cost/benefits of sensor relocations, where cost is associated with the energy expenditure on sensor relocations; and, potential benefits are associated with better information gathering (or transmission) due to the new sensor positions. Subsection B introduces subgradients of MSN utility and describes the effect of sensor relocations on the MSN performance. It also discusses communication overhead required for estimation of these subgradients.

A. MSN Utility Gain

Sensor $s \in \mathbf{S}$ relocation from point x_s to point $x_s + \Delta x_s$ during time interval $[t, t+\theta)$ results in the following gain in this sensor utility:

$$v_s(\mathbf{x}, \Delta x_s) = w_s(t+\theta, \mathbf{x}'_s | E_s - \Delta E_s) - w_s(t, \mathbf{x} | E_s) \qquad (11)$$

where function $w_s(.)$ is given by (10), ΔE_s is the energy expenditure on communication and sensor relocation, $\mathbf{x}'_s = (x_i + \delta_{is}\Delta x_i, i \in \mathbf{S})$ is the vector of sensor positions after sensor s relocation, and δ_{is} is the Kroneker symbol (i.e. $\delta_{is} = 1$ if $i = s$ and $\delta_{is} = 0$ otherwise).

Assuming a small θ, the communication energy can be approximated by $\breve{p}_s(\mathbf{x})\theta$, where communication power $\breve{p}_s(\mathbf{x})$ is given by (5). Estimation of relocation energy E_s is more complicated since this energy depends not only on the initial and final positions of sensor s , but also on the entire relocation process. Here, we assume that given sensor s initial (x_s) and final positions ($x_s + \Delta x_s$) respectively, the relocation process will take place in such a way that minimizes the relocation energy. Therefore, making E_s a function of the initial and final locations only (i.e. $\mathsf{E}_s = \mathsf{E}_s(x_s, \Delta x_s)$). In practice, this minimization is possible for sufficiently small step size θ when one may assume that the sensor motion occurs on a straight line connecting points x_s and $x_s + \Delta x_s$.

Our assumptions above lead to the following approximation:

$$\Delta E_s(\mathbf{x}, \Delta x_s) = p_s(\mathbf{x})\theta + \mathsf{E}_s(x_s, \Delta x_s) \qquad (12)$$

Small sensor relocation from points $\mathbf{x} = (x_s, s \in \mathbf{S})$ to point $\mathbf{x} + \Delta\mathbf{x} = (x_s + \Delta x_s, s \in \mathbf{S})$ results in the following gain in the aggregate utility (9):

$$V(\mathbf{x}, \Delta\mathbf{x}) \approx \sum_s v_s(\mathbf{x}, \Delta x_s) \qquad (13)$$

A natural sensor relocation algorithm maximizes this gain (13) over $\Delta\mathbf{x}$ subject to feasibility of the new sensor locations:

$$\Delta\mathbf{x}^* = \arg \max_{\Delta\mathbf{x}:\mathbf{x}+\Delta\mathbf{x}\in X} V(\mathbf{x}, \Delta\mathbf{x}) \qquad (14)$$

B. Subgradients of MSN Utility

Assuming a fixed network topology $\Gamma = (N, L)$, gradient $g_{ss}(\mathbf{x}) = \nabla_{\Delta x_s} v_s(\mathbf{x}, \Delta x_s)\big|_{\Delta x_s = 0}$ characterizes the gain in sensor s utility (10) resulting from small sensor s relocation from point x_s to point $x_s + \Delta x_s$:

$$g_{ss} = -\frac{u'_s}{T^*_s \breve{p}_s}\left(\frac{E_s}{\breve{p}_s}\nabla_{x_s}\breve{p}_s(\mathbf{x}) + \nabla_{\Delta x_s}\mathsf{E}_s(x_s, \Delta x_s)\Big|_{\Delta x_s = 0}\right) (15)$$

Expression (15) accounts for change in sensor s communication power $\breve{p}_s(\mathbf{x})$ and the energy expenditure of the relocation. Gradient $g_{si}(\mathbf{x}) = \nabla_{x_i} w_s(\mathbf{x})$ characterizes the gain in sensor s utility (10) resulting from small sensor $i \neq s$ relocation from point x_i to point $x_i + \Delta x_i$:

$$g_{si} = -\frac{u'_s}{T^*_s}\frac{E_s}{\breve{p}_s^2}\nabla^T_{x_i}\breve{p}_{si}(x_s, x_i) \qquad (16)$$

The change in sensor $s = i$ utility is due to change in sensor s communication power $\breve{p}_{si}(x_s, x_i)$.

Sensor $s \in \mathbf{S}$ relocation only affects neighboring sensors $\mathbf{S}^-_s = \{i : (i,s) \in L, i \in \mathbf{S}\}$, that directly send information to s . Sensor $s \in \mathbf{S}$, on the other hand, is only affected by relocation of neighboring sensors $\mathbf{S}^+_s = \{i : (i,s) \in L, i \in \mathbf{S}\}$, that receive information directly from s . For brevity, we further assume that all links are un-directional, i.e., $(i,s) \in L \Leftrightarrow (s,i) \in L$; and thus, sensor s relocation directly affects only sensors $i \in \mathbf{S}_s \overset{def}{=} \mathbf{S}^+_s = \mathbf{S}^-_s$: $g_{is}(\mathbf{x}) \equiv 0$ if $i \notin \mathbf{S}_s$.

Pricing-based cooperative sensor relocation algorithm, described in Subsection B, assumes that each sensor s can estimate the subgradient

$$g_s(\mathbf{x}) = \sum_{i \in \mathbf{S}_s} g_{is}(\mathbf{x}) \qquad (17)$$

which quantifies the effect of this sensor relocation on the rest of the MSN. However, except for some particular situations, estimation of the subgradient (17) by sensor s is associated with high communication overhead. Subsection C demonstrates how this communication overhead can be reduced with consensus-based algorithm, which requires sensor $s \in \mathbf{S}$ to estimate $g_{si}(\mathbf{x})$ rather than $g_{is}(\mathbf{x})$ for $i \in \mathbf{S}_s$. This is a much easier task since an intelligent sensor s can estimate $g_{si}(\mathbf{x})$ by (a) measuring its remaining battery energy level E_s and communication powers $p_{si}(\mathbf{x})$ to the neighbors $i \in \mathbf{S}_s$, (b) estimating its relocation energy $\mathsf{E}_s(x_s, \Delta x_s)$, and (c) estimating the positions of the neighboring nodes $(x_i, i \in \mathbf{S}_s)$.

IV. COOPERATIVE SENSOR RELOCATION

Subsection A proposes cooperative sensor relocation following iterations of subgradient projection algorithm for solving the optimization problem (13)-(14). Subsection B proposes consensus-based cooperative sensor relocation, which mitigates high communication overhead. This is achieved by following iterations of the *decentralized* subgradient projection algorithm. Subsection C suggests that combining *decentralized* subgradient projection algorithm with simulated annealing may result in avoiding traps on non-flat terrains.

A. Relocation by Performance Gain Maximization

In the case when sensor $s \in \mathbf{S}$ is aware of the effect of its relocation on the entire MSN (as measured by (17)), the following relocation algorithm greedily maximizes the MSN performance gain. At step k, given sensor positions $\mathbf{x}^{(k)} = (x_s^{(k)}, s \in \mathbf{S})$ and remaining sensor battery energy levels $\mathbf{E}^{(k)} = (E_s^{(k)}, s \in \mathbf{S})$, cross-layer network optimization produces the optimal network topology $\Gamma^{(k)} = (N, L^{(k)})$. At the next step $k+1$, each sensor $s \in \mathbf{S}$ is relocated following the subgradient (17):

$$x_s^{(k+1)} = P_X[x_s^{(k)} + \alpha^{(k)} g_s(\mathbf{x}^{(k)})] \tag{18}$$

where scalar $\alpha^{(k)}$ is the step size, and P_X denotes the Euclidean projection onto the set of feasible sensor locations X. Then, given new sensor locations $\mathbf{x}^{(k+1)} = (x_s^{(k+1)}, s \in \mathbf{S})$ and remaining sensor battery energy levels $\mathbf{E}^{(k+1)} = (E_s^{(k+1)}, s \in \mathbf{S})$:

$$E_s^{(k+1)} = E_s^{(k)} - \mathsf{E}_s(x_s^{(k)}, x_s^{(k+1)} - x_s^{(k)}) - \breve{p}_s(\mathbf{x}^{(k)})\theta \,, \tag{19}$$

cross-layer network optimization produces new optimal network topology $\Gamma^{(k+1)} = (N, L^{(k+1)})$..

The main assumption that sensors $s \in \mathbf{S}$ are aware of the gradients (17) can be justified for the case of symmetric propagation matrix: $\xi_{ij} = \xi_{ji}; \forall i, j \in \mathbf{S}; i \neq j$, where $\breve{p}_{ji}(x_j, x_i) \equiv \breve{p}_{ij}(x_i, x_j)$. In this case, if each sensor $i \in \mathbf{S}$ estimates the "pricing" for its battery energy $\pi_i = (u_i' E_i)/(T_i^* \breve{p}_i^2)$ and propagates this price to its neighbors, then each sensor $s \in \mathbf{S}$ can directly estimate the effect of its relocation on the rest of the MSN as follows:

$$g_{is}(\mathbf{x}) = (\pi_i / \pi_j) g_{si}(\mathbf{x}) \,. \tag{20}$$

B. Relocation by Building Consensus

Assume that each sensor $s \in \mathbf{S}$ is aware of the impact of its own and neighbors $i \in \mathbf{S}_s$ relocation on its performance as measured by $g_{si}(\mathbf{x})$. We consider cooperative sensor relocation by iterations of Distributed Subgradient Projection Algorithm (DSPA) for solving optimization problem (13)-(14). DSPA achieves cooperation by building consensus on optimal sensor locations through information exchange between each sensor and its neighbors. In effect, DSPA offers a consistent approach to overall performance optimization while minimizing the communication overhead. Again for brevity, we only describe DSPA-based cooperative sensor relocation and refer to [6] for elaborate details of DSPA.

The algorithm proceeds in steps. At step k, each sensor $s \in \mathbf{S}$ updates its own position $x_s^{(k)}$ and vector of estimates of the positions of its neighboring sensors $i \in \mathbf{S}_{s,k}$, (i.e., sensors directly communicating with sensor s at step k, $\tilde{\mathbf{x}}_s^{(k)} = (\tilde{x}_{si}^{(k)}, i \in \mathbf{S}_{s,k})$) as follows:

$$x_s^{(k+1)} = P_X[x_s^{(k)} + \alpha^{(k+1)} g_{ss}(\tilde{\mathbf{x}}_s^{(k)})] \tag{21}$$

$$\tilde{\mathbf{x}}_s^{(k+1)} = P_X[\mathbf{z}_s^{(k)} + \alpha^{(k+1)} \mathbf{g}_s(\tilde{\mathbf{x}}_s^{(k)})] \tag{22}$$

where X is the set of feasible sensor location, $\alpha^{(k+1)} > 0$ is the step size, and P_X denotes the Euclidian projection onto the set of feasible sensor locations X.

The vector $\mathbf{z}_s^{(k)}$ is the weighted average of $\tilde{\mathbf{x}}_s^{(k)}$ which is computed by sensor s as follows:

$$\mathbf{z}_s^{(k)} = \sum_{j \in \mathbf{S}_{s,k+1}} a_{s,j}^{(k+1)} \tilde{\mathbf{x}}_j^{(k)} \tag{23}$$

Scalars $a_{s,j}^{(k+1)}$ are the non-negative weights that sensor s assigns to sensor j's iteration at step $k+1$. Equation (23) represents a "consensus"-based step that ensures under conditions specified in [6], sensor $s \in \mathbf{S}$ estimates of other sensor positions converge to their actual positions (i.e. $\tilde{x}_{si}^{(k)} \to x_i$ as $k \to \infty$).

C. Escaping Traps with Random Moves

Using local information for controlled sensor relocation on non-flat terrain is prone to sensor trapping in suboptimal positions due to typical non-convexity of the performance criterion [8]. A combination of controlled sensor mobility with simulated annealing to avoid sensor trapping has been proposed in [9] and further discussed in [10]. This subsection discusses a possibility of combining DSPA-based sensor relocation and simulated annealing algorithms.

Conventional simulated annealing algorithm [9] suggests sensor relocations from points $\mathbf{x} = (x_s, s \in \mathbf{S})$ to points $\mathbf{x} + \Delta\mathbf{x} = (x_s + \Delta x_s, s \in \mathbf{S})$ with probability

$$\Pr ob(\Delta\mathbf{x}) = Z^{-1} \exp[\beta V(\mathbf{x}, \Delta\mathbf{x})] \qquad (24)$$

where function $V(\mathbf{x}, \Delta\mathbf{x})$ is given by (13), Z is the normalization constant, and β is the inverse "temperature". In the case of "high temperature": $\beta \to 0$, sensors perform random walk, while in the case of "low temperature": $\beta \to \infty$, sensors perform optimization (13)-(14). The main advantage of simulated annealing algorithm is that it allows for obtaining guidelines on the "cooling schedule" in order to ensure convergence to the global solution of optimization problem (13)-(14). Therefore, with a proper cooling schedule sensors can avoid traps.

The problem with random relocations following (24) is that this algorithm is centralized. In the spirit of a distributed subgradient algorithm, it is natural to relocate sensor s from point x_s to point $x_s + \Delta x_s$ with probability

$$\Pr ob(\Delta x_s) = Z_s^{-1} \exp[\beta v_s(\mathbf{x}, \Delta x_s)] \qquad (25)$$

where function $v_s(\mathbf{x}, \Delta x_s)$ is given by (11), Z_s is the normalization constant, and β is the inverse "temperature". In a simulated annealing version of algorithm (21)-(23) sensor s relocates from point x_s to point $x_s + \Delta x_s$ with probability (25) while updating estimates of locations of its neighboring sensors following (22)-(23).

In the case of "low temperature": $\beta \to \infty$, sensors will perform optimization (21)-(23). The similarity with conventional simulated annealing suggests the possibility of existence of a cooling schedule that allows sensors to avoid traps with a distributed version of simulated annealing.

V. Example: Tracking a Single Target

Consider a MSN designed to track a single target G and communicate the desired information at a low rate to a fixed destination D. For brevity, here we only discuss simulation results for a scenario with $S = 6$ sensors on a flat terrain where signal attenuation depends only on the distance (6). To simplify further, we assume that energy required for relocation is proportional to the travelled distance. It can be shown that under some natural conditions and for sufficiently high initial sensor battery energy levels, the optimal MSN topology is linear with only one sensor tracking the target and the rest of the sensors relaying this information to the destination, formally identified as sensor $s = 0$.

Figure 1 shows this linear topology with six sensor, where information flows from the target to the destination: $G \to (s = 6) \to \ldots \to (s = 1) \to (s = 0)$, while the control information flows in the opposite direction: $(s = 6) \leftarrow \ldots \leftarrow (s = 1) \leftarrow (s = 0)$.

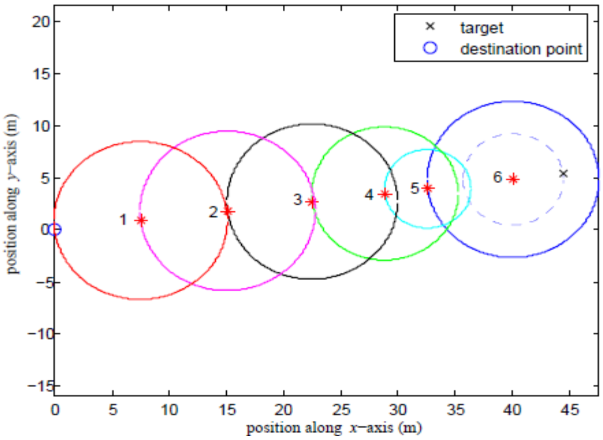


Figure 1. Linear topology with six sensors

We also assume that sensors have the maximal communication and sensing range indicated by the corresponding circles in Figure 1. Sensor relocation algorithm accounts for existence of these maximal ranges by projecting the iterations of the relocation algorithm onto properly designed set of feasible sensor positions.

In a case of stationary sensors, Figure 2 shows that sensor battery energy draining rate is completely determined by the initial sensor positions with respect to the stationary destination and the target (which may or may not be stationary).

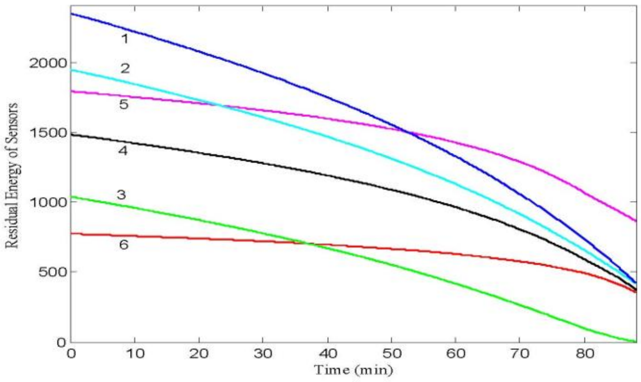


Figure 2. Residual sensor energy: stationary sensors

Figure 2 demonstrates imbalances in sensor battery energy draining rates due to initial sensor positions. Since in our model sensors use minimal communication power required for MSN operation, the network becomes non-operational in approximately 90 minutes. This is the time when sensor $s = 3$ spends all of its energy.

In the case of non-cooperative sensor relocation, shown in Figure 3, the MSN becomes non-operational in approximately 140 minutes when sensors $s = 1$, $s = 3$ and $s = 6$ deplete their battery energy.

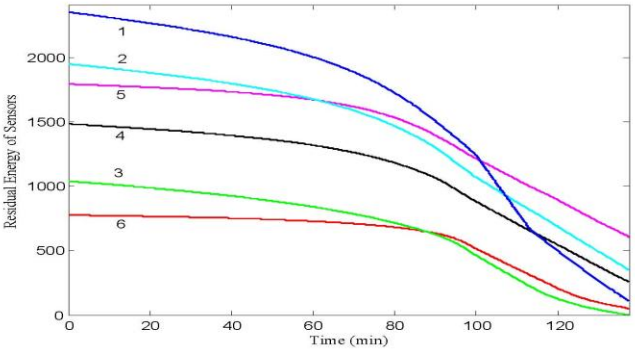


Figure 3. Residual sensor energy: selfish relocation

The inverse S-shape of the residual sensor energy evolution, shown in Figure 3, is indicative of selfish sensor relocation when each sensor attempts to prolong its own life-span. In this case, each sensor is attempting to minimize its transmission power by positioning itself at the "middle point" of the neighboring sensors without any consideration for conserving the neighbors' battery energy. This "selfish" sensor positioning is responsible for lack of coordination shown in Figure 3.

In a case of cooperative sensor relocation, shown in Figure 4, all sensors deplete their battery simultaneously prolonging MSN life span to approximately 300 minutes. This is achieved through cooperation, when sensors with longer life expectancy are "willing" to relocate longer distances in order to save energy for sensors with shorter life expectancy.

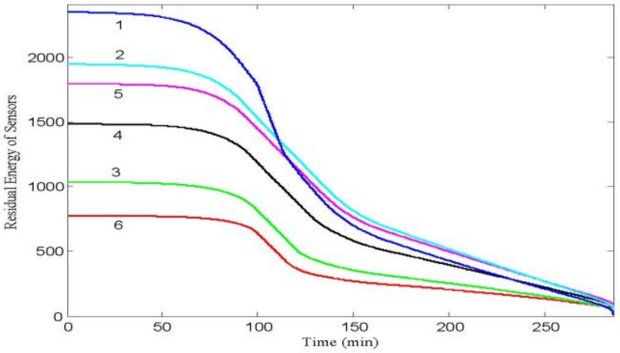


Figure 4. Residual sensor energy: cooperative relocation

VI. CONCLUSION AND FUTURE RESEARCH

This paper has proposed an approach to controlled cooperative sensor relocation in a Mobile Sensor Network (MSN) by following iterations of a Distributed Subgradient Projection Algorithm (DSPA) for aggregate performance optimization. The main advantage of this approach is in its ability to accommodate locally available information on battery energy availability, sensing, communication and relocation "costs" with respect to the energy expenditure in an inherently decentralized environment. Initial simulation results indicate viability of this approach for prolonging the MSN life-span.

Future research should evaluate pros and cons of different versions of distributed subgradient algorithms with respect to controlled sensor relocation, and provide guidelines for selection of the algorithm step size and free parameters involved in the "consensus" step. These selections affect the trade-off between the algorithm convergence rate and communication overhead. The ability to avoid traps on non-flat terrains is critical for practical implementation of controlled sensor relocation. This paper has suggested the possibility of achieving this by combining a distributed subgradient projection and simulated annealing algorithms. More studies need to be done to realize this possibility, including developing distributed cooling scheduling.

Finally, note that the important issue of initial network formation has not been discussed. Distributed subgradient optimization algorithms assume agent connectivity. However, at the initial stage mobile sensors may be organized in several disconnected clusters. In that case, controlled sensor mobility should simultaneously pursue two goals: collaboration within clusters and establishing connectivity between clusters. This problem, which can be broadly framed as controlled mobility in Disruption Tolerant Networks (DTN), would require new ideas and approaches.

REFERENCES

[1] I.F. Akyildiz, W Su, Y. Sankarasubramanian, and E. Cayirci, "Wireless sensor networks: a survey," Computer Networks, Elsevier, No. 32, 2002, pp. 393-422.

[2] V. Marbukh and K. Sayrafian-Pour, "A Framework for Joint Cross-Layer and Node Location Optimization in Mobile Sensor Networks," *Proc. Adhoc-Now'09*, Nice, France, 2008.

[3] V. Marbukh and K. Sayrafian-Pour, "Mobile sensor network self-organization for system utility maximization: work in progress". *Proc. The Fifth International Conference on Wireless and Mobile Communications (ICWMC'09)*, Cannes 2009.

[4] H. Mahboubi, A. Momeni, A. G. Aghdama, K. Sayrafian-Pour, and V. Marbukh, "Minimum cost routing with controlled node mobility for target tracking in mobile sensor networks" 2010 American Control Conference (ACC2010), USA, 2010.

[5] A. Howard, M.J. Mataric, and G.S. Sukhatme, " Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem," Proc. 6th Intern. Symp. On Distributed Autonomic Robotic Systems (DARS'02), Fukuoka, Japan, 2002.

[6] S. Sundhar Ram, A. Nedic, and V.V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," to appear in Journal of Optimization Theory and Applications, 2010.

[7] S. Sundhar Ram, A. Nedic, and V.V. Veeravalli, "Incremental stochastic subgradient algorithms for convex optimization," SIAM Journal on Optimization, 20 (2), 2009, 691-717.

[8] Y. Koren and J. Borenstein, "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation," *Proceedings of the IEEE Conference on Robotics and Automation*, Sacramento, California, April 7-12, 1991, pp. 1398-1404.

[9] R. Rao and J. Kesidis, "Purposeful mobility for relaying and surveilance in mobile sensor network," IEEE Transactions on Mobile Computing, No. 3, Vol. 3, 2004, pp. 225-232.

[10] V. Marbukh, K. Sayrafian-Pour, H. Mahboubi, A. Momeni, and Amir G. Aghdam, "Towards evolutionary-pricing framework for mobile sensor network self-organization," in Proceedings of 2010 IEEE World Congress on Computational Intelligence, Barcelona, Spain, 2010.

Mobility Model for Self-Configuring Mobile Sensor Network

Andrzej Sikora*†, Ewa Niewiadomska-Szynkiewicz*†
* *Institute of Control and Computation Engineering, Warsaw University of Technology*
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: asikora@elka.pw.edu.pl, ens@ia.pw.edu.pl
† *Research and Academic Computer Network (NASK)*
ul. Wawozowa 18, 02-796 Warsaw, Poland
e-mail: andrzej.sikora@nask.pl, ewan@nask.pl

***Abstract*—A self-configuring sensor network is a collection of wireless devices that collaborate with each other to form a network system that adapts to achieve a goal or goals. Such network is often built from mobile sensors that may spontaneously create a network and dynamically adopt to changes in the unknown environment and network requirements. Mobility pattern is a critical element that influences the performance characteristics of mobile sensor networks (MSN). In this paper, we discuss main directions to mobility modeling and present a systematic taxonomy of the mobility models that is provided in literature. Finally, we describe a novel algorithm for calculating mobility patterns for mobile devices that is based on a cluster formation and an artificial potential function. Our model can be used both to a simulation-based design of MSN, and to a motion planning for real, physical MSN. The presented simulation study for a rescue mission planning illustrates the possible application of our model.**

***Keywords*-self-configuring network; mobile sensor network; mobility models; potential field;**

I. Introduction

In the last years, wireless sensor networks (WSN) have gained increasing attention from both the research community and users [3], [11]. WSN are distributed architectures formed by a set of wireless devices that can freely and dynamically organize themselves into temporary network topologies. Typical WSN usually consists of stationary devices. In many real life applications networks formed by stationary nodes suffer insufficiency. Therefore, there is a need for mobile sensor network (MSN) that is capable to change its layout and position. Such a network is formed by mobile sensor devices.

This paper considers issues concerning self-configuring MSN design and development. We focus on network systems formed by mobile, wireless devices that may spontaneously create a network, manage movement of sensor nodes, assemble the network themselves, dynamically react to changes in the domain and network requirements. There are many benefits of these features, and many potential applications. Examples include an optimal coverage for monitoring of an unknown environment, producing a well connected network despite the limited resources, providing a connection between data sources and data sinks, which are not necessarily uniformly distributed across the network, and others. It is obvious that the mobility of MSN can be used to improve its performance characteristics, such as sensing coverage or network connectivity. The question is how the mobility can efficiently be managed toward a better network system performance. We present and describe a novel approach to design MSN. The proposed concept of mobility patterns calculation is based on a cluster formation and an idea of potential function commonly used in robots navigation.

II. Mobility modeling in MSN

Modeling of node mobility plays the crucial role in design and development of MSN systems. Simulation results show that the communication protocols performance may vary drastically across mobility models and performance rankings of protocols may vary with the mobility models used [1], [7]. Therefore, studying the performance of ad hoc networking protocols and application services in the presence of mobility is an important stage of the design process. It implies that the characteristics of mobility models of mobile nodes need to be analyzed and studied very carefully. It is obvious that real-life movement patterns are very difficult to obtain, and realistic models are usually very complicated. Many less and more detailed mobility models have been introduced, and are described in literature. The survey can be found in [7].

In this paper, we present the mobility models taxonomies provided in [1] and [7]. In general, we can distinguish two approaches for mobility patterns modeling [2], [7], [8]:

- *Motion traces models* (TM). The deterministic models, that require the accurate information about mobility patters (i.e., positions of nodes in time).
- *Syntactic models* (SM). The analytical random-motion models, that uses randomness in calculation of traversing patterns from one place to another. They can be classified based on the description of the mobility patterns into: individual mobile movements and group mobile movements, and based on the degree of randomness into: constrained topology-based models and statistical models. In constrained topology-based models the movement is restricted by various constraints:

pathways, speed limits, obstacles, etc. In statistical model the node is allowed to move anywhere in the domain, hence the model is based on total randomness.

Bai et al. [1] classify the mobility models based on their basic mobility characteristics:

- *Random models*. As in statistical models, nodes move randomly, and can be classified further based on the degree of randomness. A Random Mobility model (RM) that implements Brownian-like motion, and a Random Waypoint model (RWP) with randomly generated destination point and velocity are popular models from this group.
- *Models with temporal dependency*. The mobility patterns are influenced by the previously generated movement patterns. A Gauss-Markov and smooth random mobility model fall into this category.
- *Models with spatial dependency*. The nodes tend to move in a correlated manner. A reference point group mobility model belongs to this category.
- *Models with geographical restrictions*. The movements of all nodes are constrained by streets, roads, obstacles, etc. Path-based and obstacle mobility models fall into this category.

Roy [7] provides the alternative classification. He divides the models into seven groups:

- *Individual mobility models*. The mobility pattern for the individual node is calculated.
- *Group mobility models*. The mobility pattern for a group of cooperative nodes is calculated.
- *Autoregressive mobility models*. The mobility patterns are correlated with the mobility states (i.e.: position, velocity, acceleration at consecutive time instants).
- *Flocking and swarm mobility models*. The mobility patterns imitate the trajectories performed by dynamic nodes of self-organizing networks in nature (like swarms).
- *Virtual game-driven mobility models*. The mobility pattern calculation takes into account the interactions with all other nodes in a network or with groups of nodes.
- *Non-recurrent mobility models*. It is assumed that a network permanently changes its topology in time. A node moves in a totally unknown way, and previous patterns are not repeated.
- *Social-based mobility models*. The family of the mobility models that are associated as a community of groups within a society. The models describe non-homogenous behaviors in both space and time.

The mobility models that are examples of above-quoted categories of models are collected in Table I.

Table I
MOBILITY MODELS.

Group of models	Models
Individual mobility models	Random walk mobility
	Random waypoint mobility
	Smooth random mobility
	Geographic constraint mobility
	Realistic random direction mobility
	Deterministic mobility
	Partially deterministic mobility
	Random Gauss-Markov mobility
	Semi-Markov smooth mobility
	Steady-state generic mobility
	Graph-based mobility
	Hierarchical influence mobility
	Boundless simulation area mobility
	Behavioral mobility
	Fluid-flow mobility
	Potential field mobility
	Correlated diffusion mobility
	Particle-based mobility
Group mobility models	Reference point mobility
	Reference velocity mobility
	Reference velocity
	& acceleration mobility
	Structured mobility
	Virtual track-based mobility
	Drift mobility
	Group force mobility
Autoregressive models	Autoregressive individual mobility
	Autoregressive group mobility
Flocking and swarm models	Flocking mobility
	Swarm group mobility
Virtual game-driven models	Virtual game-driven mobility
	Virtual game-driven mobility
Non-recurrent models	Non-recurrent mobility
Social-based models	Time-variant mobility
	Community-based mobility
	Orbit-based mobility
	Entropy-based mobility
	Knowledge-driven mobility

III. MOBILITY MODEL FOR SELF-CONFIGURING MSN

We consider a problem of design a self-configuring network of mobile nodes, connected by wireless links. We assume that to achieve a goal network nodes should collaborate, and a whole network should enable continuous communication with the base station, hence the network must be connected. Collective motion is often required in mobile sensing networks. It involves communication among and between individual nodes or clusters of nodes to coordinate their movement. We assume that the network system should change its topology to achieve a goal. The objective is to calculate mobility patterns for all network nodes. The use of relatively simple random mobility models did not give satisfactory results in our experiments. Therefore, we have developed a novel algorithm to calculate mobility patterns for a mobile sensor network. Our mobility model resembles a collision-free movement of a group of mobile devices. It can be used in ad hoc networks simulation for design of network scenarios or for motion planning for real MSN.

In our research we have focused on the individual mobility where the mobility pattern of an individual node is considered. Our model combines two approaches – *potential field* and *particle-based mobility modeling*. The concept to build an artificial potential field where the mobile devices move from a *high-value* state to a *low-value* state, and define an associated potential function that captures both operational goals and the environment of a network is a popular direction in motion planning in mobile robotics [4], and mobile sensor networks [5]. Due to such model, the determined mobility pattern of each node includes attraction to the destination and repulsion from each obstacle. In these approaches sensor nodes not only receive forces from the surrounding environment, but also receive forces from one another. The particle-based mobility modeling [7] that considers each mobile node as a "self-driven" moving particle in the physics of Newtonian mechanics or quantum mechanics is the other popular technique in management of mobile devices. Each node is characterized by a sum of forces, describing its desire to move to the direction, avoiding collisions with other nodes and obstacles. The driving force is associated with each node, and is self-produced.

A. Problem Formulation and Network System Description

Let us consider a set of mobile wireless devices that compose MSN, and are assumed to operate in a three-dimensional field filled with obstacles. Each node navigates itself to a particular location to achieve the goal. The objective is to calculate the optimal motion trajectory from one configuration to another that meets the following requirements:

1) the mobility should be managed toward a better coverage and well connected network that enables a continuous communication with a base station,
2) the traversing pattern from one place to another has to be collision free and should allow to push the network node through the narrow passage,
3) the traversing pattern has to capture the environment requirements (the signal propagation can change in time).

We propose the scheme for management of nodes' movement based on a cluster formation and application of an artificial potential function that captures the above-quoted requirements. In our formulation all network nodes and obstacles form a set S of N entities; O_i, $i = 1, \ldots, N$. We assume that our obstacles can move as well (nodes can be obstacles for other nodes in the network), hence the obstacles are the same type of entities as the network nodes. We define each entity O_i as a solid body, which position is described by three Cartesian coordinates $[x_i, y_i, z_i]$ and orientation is given by a quaternion [4]: $Q^i = q_0^i + q_1^i\mathbf{i} + q_2^i\mathbf{j} + q_3^i\mathbf{k}$. We consider objects that are of different shapes. To simplify the calculation we made an assumption that the interactions between each pair of entities (O_i and O_j) are described by the interactions between points selected from O_i and O_j (see Fig. 1). Hence, in case of O_i the set of selected points is as follows: $P^i = \mathbf{p}_i^1, ..., \mathbf{p}_M^i$, $P^i \in O^i$, with $\mathbf{p}_1^i = \mathbf{c}^i$, where $\mathbf{c}_i$ is the central point, and M_i - 1 other points are selected by the user. Such a representation of a node allows us to implement translation and rotation, hence it is easy to rotate the network node and push through the narrow passage, Fig. 2. The rotation is given by a quaternion product.

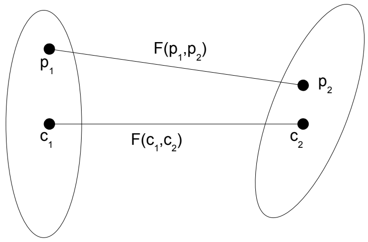


Figure 1. The node description.

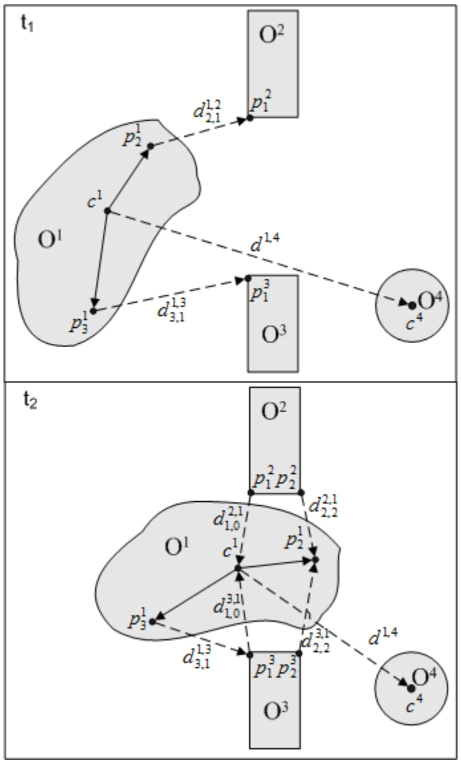


Figure 2. The rotation of the mobile device.

Obviously, it is possible to simplify the description, – each object O_i can be described by a single point $\mathbf{c}_i$ (similarly to commonly used mobility models). In such an approach the mobility pattern calculation simplifies, but the generated trajectory is less realistic.

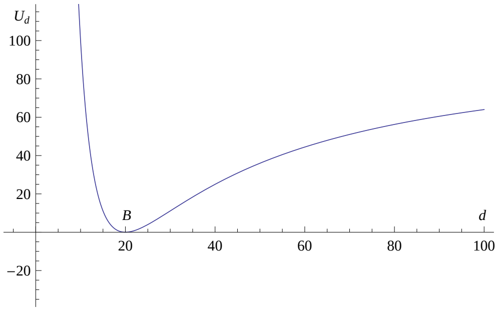


Figure 3. The artificial potential function.

B. Potential Field Mobility Model

Inspired by classical dynamics that study the motion of objects in the concept of an artificial potential fields, and particle-based modeling we propose the model in which the mobility of each node in the network is governed by the description of an artificial potential function. The potential function U is a differentiable real valued function, which value can be viewed as an energy, and hence the gradient of the potential is a force. The gradient is a vector, which points in the direction that locally maximally increases U. The potential function can be constructed as a sum of attractive and repulsive potentials. The meaning of the attractive/repulsive is straightforward: the goal attracts the mobile device while the obstacle repels it. Therefore, the sum of attractive and repulsive influences draws the mobile device to the goal while deflecting it from obstacles. The *gravity mobility model*, which is based on the use of Newton's gravitational law of motion in classical dynamics to calculate a mobility pattern is the example of this approach. Unfortunately, it is insufficient in many applications. The model often introduces oscillations into the movement of nodes. The oscillations are hard to eliminate. To address this problem, we constructed a simple potential function that captures all the requirements for calculated mobility pattern mentioned in the previous paragraph. The inspiration came from classical mechanics and liquid crystals where it is popular to model the interactions between a pair of neutral atoms or molecules via Lennard-Jones potential function (see [10] for details). We propose the simpler function with similar characteristics:

$$U_{ab}^{ij}(\hat{d}_{ab}^{ij}) = \begin{cases} \epsilon m_a^i m_b^j \left(\frac{\bar{d}_{ab}^{ij}}{\hat{d}_{ab}^{ij}} - 1\right)^2 & |\bar{d}_{ab}^{ij} - \hat{d}_{ab}^{ij}| > \tau_{ab}^{ij} \\ 0 & |\bar{d}_{ab}^{ij} - \hat{d}_{ab}^{ij}| \leq \tau_{ab}^{ij} \end{cases} \tag{1}$$

where $\hat{d}_{ab}^{ij}$ denotes the estimated distance between points $\mathbf{p}_a^i$ and $\mathbf{p}_b^j$, $\bar{d}_{ab}^{ij}$ the reference inter-node distance (calculated due to maximal radio range), ϵ, m_a^i, m_b^j and τ_{ab}^{ij} are parameters. The point reaches an unstable equilibrium for $\hat{d}_{ab}^{ij} \in [\bar{d}_{ab}^{ij} - \tau_{ab}^{ij}, \bar{d}_{ab}^{ij} + \tau_{ab}^{ij}]$, as depicted in Fig. 3. Similarly to the Lennard-Jones potential the form of U_{ab}^{ij} has no theoretical justification. In the reference position of our node $\hat{d}_{ab}^{ij} = \bar{d}_{ab}^{ij}$ we obtain $U_{ab}^{ij} \approx 0$; it means the best coverage on condition of full connected network. However, in unknown environment with obstacles it is usually impossible to move a node to an optimal position. Hence, we can calculate the estimated distance $\hat{d}_{ab}^{ij}$ solving the optimization problem

$$\min_{\hat{d}_{ab}^{ij}} U_{ab}^{ij}(\hat{d}_{ab}^{ij}) \tag{2}$$

It is obvious that the calculation of the whole trajectory since the node reaches the equilibrium point is not an easy task because of the numerous actors operating in the scene. However, we can describe our mobile network as a physical system consisting of objects that are forced to move in the advisable direction with the adequate speed. The **algorithm 1** for traversing pattern calculation at time instants $t_0 + \Delta t, t_0 + 2\Delta t, \ldots$ for i-th node is as follows:

- *Step 1.* Calculate the reference inter-node distances due to current maximal radio range and environment characteristics (for all points of O^i).
- *Step 2.* Calculate the values of $\hat{d}_{ab}^{ij}$ for all points of O^i using the formula (1).
- *Step 3.* Calculate the displacement for the whole object O^i (the i-th node) for results of *Step 2*.
- *Step 4.* Move the i-th node to the new position in the domain.
- *Step 5.* Rotate the i-th node (if necessary).
- *Step 6.* Calculate and broadcast to the network the new positions of all points of O^i. Return to *Step 1.*

C. Reference distances calculation

In the first step of our algorithm we have to calculate the reference distances for all points selected from a given node, due to the current maximal radio range, and assumed probability of connection between nodes in a network. To solve this problem we can use Q-function defined in [6]. Unfortunately, Q-function depends on two parameters: n called "distance-power gradient" that indicates the rate at which a signal strength decreases with a distance, and the signal disturbance X_σ that is a zero-mean Gaussian distributed random variable with standard deviation σ. To estimate both these parameters we apply the commonly used radio signal propagation model that indicates that received signal power decreases with a distance, both in outdoor and indoor environments. Therefore, the power of the signal received by a receiver P^r at a distance d is defined as

$$P^r(d)[dBm] = P^t[dBm] - PL(d)[dB], \tag{3}$$

where P^t denotes power used by a sender to transmit the signal and $PL(d)$ the average signal degradation (path loss) with a distance d. A path loss $PL(d)$ is modeled as follows:

$$PL(d)[dB] = PL(d_0)[dB] + 10nlog\left(\frac{d}{d_0}\right) + X_\sigma + \sum_i PAF_i \tag{4}$$

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

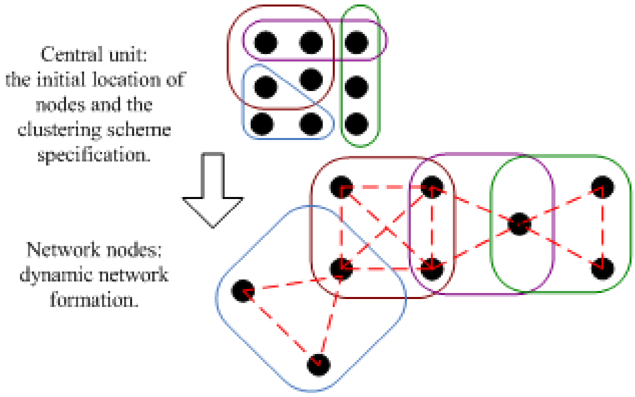


Figure 4. Two steps in formation of a mobile sensor network.

where d_0 is a close-in reference distance (for IEEE 802.15.4 usually d_0=1m), PAF_i is a partition attenuation factor for i-th wall (experimentaly determined).

We calculate n and X_σ using formulas (3) and (4), assuming $d = d_c^{ij}$ (d_c^{ij} is a real distance between nodes i and j calculated for known nodes locations) and measured P^r. We assume that all nodes are equipped in any location system and are aware of their own location. The detailed description of n and X_σ computing can be found in [6].

D. Self-configured MSN Design

Consider a situation where the task is to create a network that enables the continuous communication with a base station to achieve a given goal or goals. We can form a cluster structured MSN that covers the area between the base station and the cell containing a goal or a set of goals. The cluster formation is based on the following characteristics: 1) all nodes are grouped into overlapping clusters with two, three or four elements, Fig. 4, 2) nodes in a cell must be able to communicate with each other, 3) each cluster can communicate with all neighboring clusters.

The design of a self-configuring network is performed in two steps. We can distinguish two groups of calculation units: the central unit (the base station) and the set of local units – the mobile nodes (see Fig. 4).

Central unit: The central unit task is to determine the initial location of nodes and the clustering scheme i.e., the number of cells and the assignment of nodes to clusters. Decomposition of a network into clusters may be predefined or calculated by the dedicated clustering algorithm.

Network nodes: Each node sends at time instants $t_0 + \Delta t, t_0 + 2\Delta t, \ldots$ broadcast messages with its location, transmitted signal power P^t, cluster (or clusters) identifier (id), data about neighbors. We use MAC protocol with beacon synchronization. Next, the node calculates its displacement using the **algorithm 1** presented in Subsection III-B. The calculations are performed based on current distances between nodes and the measured signal power strength. We assume that for reference distances $\bar{d}_{ab}^{ij}$ in the formula (1) the probability of connection between nodes in cluster has to be equal to 99 percent or higher. Finally, the network consisting of calculated clusters is formed. It should be pointed that the value of the reference distance is adaptively modified due to the dynamic changes both in the deployment area and the set of network devices (decreased energy resources).

IV. Simulation Results

In order to evaluate the performance of our mobility model simulations of various ad hoc network topologies and tasks have been performed. In this paper, we present the example application of our model to support the design of a self-configuring, well-connected network that enables a continuous connection between a goal and a base station.

Consider a situation where the fixed network infrastructure in a disaster area is damaged due to an explosion at a chemical plant. We plan to send several rescue teams to work on the disaster scene (0.36km^2). A rescue mission requires that new communication channels be quickly established. MSN can be successfully used to solve this problem. It can enable communications with an adequate quality and can adapt to changing conditions and requirements in the danger zone. In case when we plan a rescue action it is useful to check various possible scenarios taking into account all constraints concerned with the environmental conditions. In presented problem the developed MSN should provide the continuous communication with all rescuers during the rescue action. Simulations were performed in our software platform for parallel ad hoc networks simulation, called MobASim, and described in [9]. The goal of the experiments was to create a network topology with minimal number of nodes that ensures the connection between all rescuers and the base station. We present the simulation of 180 seconds of given ad hoc network operation. The network was composed of 4 rescuers and 10 mobile devices used for re-establishing the communication infrastructure. The mixed outdoor and indoor environment was considered (see Fig. 5).

As a final result of our simulations we have obtained the network consisting of eight clusters with irregular shapes, as presented in Fig. 6. The estimated values of calculated distances $\hat{d}$ between nodes in clusters 0, 1, 5 and 6 calculated at given time instants are presented in Table II.

V. Summary and Conclusions

In this paper, we have described the novel approach to managing the mobility of a mobile sensor network that combines potential field and particle-based schemes for calculating the mobility patterns. We have defined the suitable artificial potential function for the optimal inter-node distances calculation that captures the task and environment of MSN requirements. In our opinion the proposed mobility model is a good compromise between representativeness

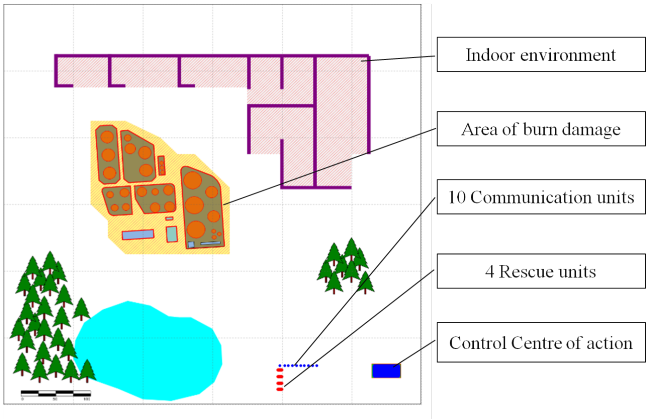


Figure 5. The disaster scene.

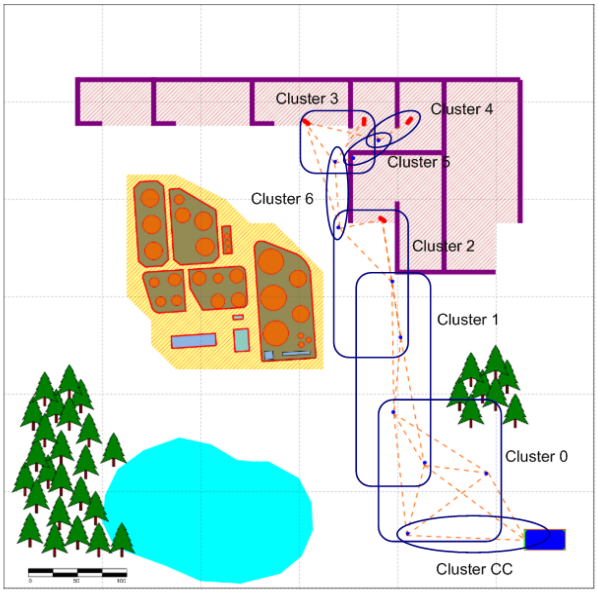


Figure 6. The final topology formed after 180 seconds of wireless devices operation - eight clusters with irregular shapes.

Table II
THE TEMPORAL INTERNODE DISTANCES $\hat{d}$ (CLUSTERS: 0, 1, 5, 6).

Cluster 0		Cluster 1		Cluster 5		Cluster 6	
T[s]	$\hat{d}$[m]	T[s]	$\hat{d}$[m]	T[s]	$\hat{d}$[m]	T[s]	$\hat{d}$[m]
101	80.754	101	82.438	101	6.608	101	17.781
102	82.914	102	83.124	102	6.688	102	17.926
106	83.164	103	84.108	103	6.999	103	18.081
107	83.555	104	85.824	105	7.284	107	20.034
108	84.577	105	86.740	106	7.629	108	20.866
109	85.399	132	87.150				
110	86.674	133	87.874				
111	87.629	139	88.175				
117	89.152	142	89.094				
118	89.747	143	88.668				
122	90.016	144	89.094				
139	90.359	147	88.783				
		174	89.403				

and simplicity. The presented case study showed that by employing a multihop wireless communication and mobile nodes acting as communication relay stations, with movement calculated due to our model even relatively distant points in the deployment area will be able to communicate with the base station. In future research we plan to compare our scheme with other existing models, and test its utility to other ad hoc systems.

ACKNOWLEDGMENT

This work was partially supported by National Science Centre grant NN514 672940.

REFERENCES

[1] Bai F., Sadagopan N., and A. Helmy, ”IMPORTANT: A Framework of Systematically Analyze the Impact of Mobility on Performance of Routing Protocols for Ad Hoc Networs”, *Proc. of INFOCOM*, Vol. 2, pp. 825-835, San Francisco, USA, 2003.

[2] Basagni, S., Conti, M., Giordano, S., and I. Stojmenovic, *Mobile Ad Hoc Networking*, Wiley-Interscience, IEEE Press, 2004.

[3] Bauer, P.H., ”New Challenges in Dynamical Systems: The Networked Case”, *International Journal of Applied Mathematics and Computer Science*, University of Zielona Gora Press, Vol. 18, No 3, pp. 271-278, 2008.

[4] Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L.E., and S. Thrun, *Principles of Robot Motion*, The MIT Press, Cambridge, 2005.

[5] Ma K., Zhang Y., and W. Trappe, ”Managing the Mobility of a Mobile Sensor Network Using Network Dynamics”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, No. 2, pp.106-120, 2008.

[6] Rappaport, T.S., *Wireless Communications. Principles and Practice*, Prentice Hall, USA, 2009.

[7] Roy, R.R., *Hanbook of Mobile Ad Hoc Networks for Mobility MOdels*, Springer, USA, 2010.

[8] Santi, P., *Topology Control in Wireless Ad Hoc and Sensor Networks*, John Wiley & Sons, Ltd, UK, 2006.

[9] Sikora, A. and E. Niewiadomska-Szynkiewicz, ”A Parallel and Distributed Simulation of Ad Hoc Networks”, *Journal of Telecommunications and Information Technology*, vol. 3, pp. 76-84, 2009.

[10] Singh, S., *Liquid Crystals. Fundamentals*, World Scientific Publishing Co. Pte. Ltd., Singapore, 2002.

[11] Verdone, R., Dardari, D., Mazzini, G., and A. Conti, *Wireless Sensors and Actuator Networks. Technologies, Analysis and Design*, Elsevier, 2008.

A Survey of Modeling Techniques for Wireless Sensor Networks

John Khalil Jacoub
University of Ontario
Institute of Technology
Oshawa, Ontario, Canada
john.khalil@uoit.ca

Ramiro Liscano
University of Ontario
Institute of Technology
Oshawa, Ontario, Canada
ramiro.liscano@uoit.ca

Jeremy S. Bradbury
University of Ontario
Institute of Technology
Oshawa, Ontario, Canada
jeremy.bradbury@uoit.ca

***Abstract*—Wireless Sensor Networks (WSNs) monitor environment phenomena and in some cases react in response to the observed phenomena. The distributed nature of WSNs and the interaction between software and hardware components makes it difficult to correctly design and develop WSN systems. One solution to the WSN design challenges is system modeling. In this paper we present a survey of 9 WSN modeling techniques and show how each technique models different parts of the system such as sensor behavior, sensor data and hardware. Furthermore, we consider how each modeling technique represents the network behavior and network topology. We also consider the available supporting tools for each of the modeling techniques. Based on the survey, we classify the modeling techniques and provide future directions to enhance the use of modeling in the design of WSNs.**

***Keywords*-Wireless Sensor Networks (WSN), Modeling, Code Generation, Model Checking, Analysis.**

I. Introduction

A WSN consists of small wireless units called motes, which are attached to a specific type of sensors. The sensors measure an environment phenomenon (e.g., humidity, temperature, or soil moisture) [6] and the measured value is expressed as an analog signal generated by the sensors. The analog signal is converted to a digital signal within the mote and transmitted wirelessly to a mote that has expressed interest in that data [21]. This mote is generally called a collector node. The collector mote could also be attached to a gateway to facilitate the transfer of data between the WSN and other devices on more conventional networks like IP. Additionally, some WSN applications have been extended to not only sense the phenomenon but also to react in response to the sensed data. These networks are typically referred to as Wireless Sensor Actor Networks (WSANs) [9].

WSN systems can be complex and many different challenges can arise during the design of a WSN. One challenge is the distribution of nodes and sensors in a physical environment that may result in lost and delayed data. A second challenge is the inclusion of real-time behavior within a distributed WSAN. Many WSANs require real-time communication and control when a specific phenomenon is observed [9]. A third challenge is memory management within the sensor nodes [6]. The small size of the nodes leads to physical limitations that restrict the available memory and therefore memory management is often required. A fourth challenge is operational reliability. WSNs often consist of self-powered nodes in environmentally challenging domains imposing strong reliability requirements. For example, there is a requirement to maximize the life-time of a network because the nodes have a limited power source [20]. A fifth challenge is improving the network performance, i.e. reduce network delays, packet loss, while increasing throughput. Network performance improvements may involve the use of concurrency and event driven communication, which can add additional complexity to the system.

The design of WSN systems usually occurs at the implementation level and does not involve design at higher levels of abstraction. This leads to a decrease in code portability and to platform-specific implementations [15]. A WSN system produced using this approach is prone to both design and implementation errors and very challenging code debugging [16] (user interfaces to sensor nodes are very limited so even simple text output is challenging.) If errors are not detected during the implementation and verification stages of development then they may appear once the system is deployed and is operational. The nodes of an operational WSN application are generally difficult to access once they are deployed in their working locations [20].

The challenges of developing WSN systems often benefit from higher level-design and analysis. The use of modeling languages and techniques can drive the design through different abstraction layers and analysis tools can help refine the model [15]. In this paper we survey 9 modeling techniques for WSNs. For each technique we examine how it models the WSN at the node and system-level.

The rest of the paper is organized into six sections. Section II gives an overview of the modeling technique reviewed in this survey. Section III discusses the modeling of WSN elements including sensors, nodes and hardware while Section IV discusses modeling at the system level. Section V discusses the supporting tools and the importance of each tool for WSN design. Finally, Section VI provides

ISBN: 978-1-61208-144-1

conclusions and future research directions.

II. Background

In this section, we provide an overview of each of the modeling techniques included in our survey (see Table I). Some of those modeling techniques have been used in the application development process, while others take WSNs as a case study for their modeling approach. Some use an appropriate notation to support the aim of the modeling, such as UML,while others use their own notation, such as the Insense technique [6]. The survey discusses the details of each modeling techniques, but not the contribution of each notation in the area of modeling.

The techniques use different basic elements (e.g., channels, processes, modules, components) to express a WSN as a model. A channel is used to represent the communication between two elements of a WSN. For example, channels can represent the characteristics of sensor-node communication, node-node communication and node-gateway communication. Processes, modules, and components are used to represent the sensors and nodes of a WSN. We will describe details of the modeling elements later when we discuss the individual modeling techniques.

The modeling techniques surveyed also vary in terms of the scope of modeling. For example, some techniques are intended to model a single node or communication between a pair of nodes while others are intended to model the entire WSN.

A. High-Level SDL Models (HL-SDL)

HL-SDL is a modeling language that uses the Specification and Description Language (SDL), which is normally used to model and simulate communication protocols [12] .SDL has been adapted by Dietterle, et al., to model TinyOS components using SDL processes (i.e., extended finite state machines) [10]. The system is modeled as a collection of channels and processes. The model can be used to generate nesC source code, which is commonly used in WSNs based on the TinyOS environment. While generating nesC code, each process (which is the smallest unit of the model) represents a component in TinyOS [13]. In their work, Dietterle et al. used manual optimization to enhance the generate code [10].

B. Insense

Dearle et al. use the Insense modeling language to create a component-based model for a WSN [6]. Components in Insense are concurrent and they communicate synchronously via directional channels that are used to abstract away from low-level synchronization and communication issues [6]. Insense is built in the Contiki operating system(Contiki is a popular operating system used for WSN) [11]. The Insense model has a translator that produces C source code that can be used to calculate important details such as worst case execution times (WCETs) and worst case space (WCS) within a given WSN [6].

Approach	**Notation**	**Modeling Scope**	**Modeling Elements**
HL-SDL [10]	SDL	Node	processes-channels
Insense [6]	Insense Language	Node	components, channels
MathWorks [16]	State Diagram and C	Node-Network	state-charts, communication medium
MDEA [15]	UML	Node-Network	components (wireless link as a class)
PM [19]	Promela	Network	processes, channel
SensorML [4]	XML - source code (optional)	Node	components-processes model
SystemC-AMS [21]	Block diagram - C++	Node-Node Communication	block diagram, source code
UM-RTCOM [9]	CORBA	Node-Network	components, channels
XRM [8]	eXtended Reactive Modules	Node-Network	modules

Table I
Overview of Modeling Techniques

C. MathWorks Modeling Approach

The framework aims to design, simulate, and generate the code for WSNs. The node behavior is modeled as a parameterized Stateflow block. Nodes in the MathWorks approach also containing timing and random number generators that are used for simulation. Additionally, the communication medium, which is used to define the connectivity between the nodes, is represented at a lower abstraction level and is implemented in the C language. By leveraging MathWorks tools such as animated state charts, chart displays, scopes, and plots, analysis of the WSN can be performed. According to the results the model can be refined. The final stage is to generate the WSN code using the Target Language Complier (TLC) which can generate C code for MANTIS [2] and nesC code for TinyOS [13]. The Mathworks approach has been used successfully to generate the code for Energy Efficient and Reliable In-Network Aggregation (EERINA) [17] algorithm for clustered sensor networks.

D. Model Driven Engineering Approach (MDEA)

Losilla, et al. use UML and a Model Driven Engineering (MDE) approach that includes three modeling layers:

- WSN Domain Specific Modeling: a meta-model that is created by a domain expert.
- Component-based Platform Independent Models (PIMs): A UML-like language primarily composed of activity diagrams and state-machine diagrams.

- NesC Platform Specific Model (PSM): used with the UML PIMs to generate the NesC source code.

Transformation rules control moving from one modeling layer to another. Moreover, refinement can occur after every transformation to improve the generated model. This MDE approach is supported by the Eclipse IDE as well as a number of Eclipse plug-ins (e.g., MOFScript) that are responsible for automating the transformation process. The MDE approach has been used to generate the nesC code for the MITRA WSN application which was designed for the precision agriculture applications [15].

E. Promela Model (PM)

Modeling an Ad-Hoc Sensor Network in Promela is an example of using the input language of a model checker to specify a WSN. The model checker used is Spin and the input language is Promela. The Promela model includes the physical location of all the dynamic nodes and supports adding and removing nodes as well as changing their physical location. The Spin model checker is used to perform a network connectivity check. Specifically, the physical location data of the nodes is analyzed in conjunction with the data about the coverage range of the sensors [19].

F. SensorML

SensorML is an XML based language that supports modeling each sensor by specifying the sensor's meta-data (e.g., sensor Id, sensor type). The model includes representations of the physical elements (e.g., the sensor and actuators) and the non-physical element (e.g., mathematical operation within the sensor). All of the elements are modeled as processes that are linked together explicitly through inputs and outputs. Linked sequences of processes form process chains that correspond to the behavior inside a single node [4]. Sensor Web Language (SWL) is a simple version of SensorML and is used in the WSNs deployments to achieve the interfaces between the network elements base station, sensors, and web browser. Additionally, the language can be used to achieve the interface between two or more different WSNs [18].

G. SystemC-AMS

SystemC-AMS combines C++ with block diagrams to model the WSN and simulate the system. For each node, SystemC-AMS models the Analog to Digital Converter (ADC), the microprocessor, and the channel. The goal of SystemC-AMS, is to calculate the Signal to Noise Ratio (SNR) for the ADC and Bit Error Rate (BER) for the communication channel between two nodes [21].

H. UM-RTCOM Model

UM-RTCOM is a real-time component based modeling framework written in CORBA that is composed of sensors, actors and a coordinator (the coordinator in located on a base station). Actors gather data from groups of sensors based on their physical locality and respond to a phenomenon identified by the coordinator [9]. Sensors communicate with actors and actors communicate with the coordinator using channels. Communication via a channel is modeled as a tuple. "A tuple is a sequence of fields with the form: (t1, t2, ..., tn) where each field ti can be: a TC identifier (or) a value of any established data type of the host language where the model is integrated" [9]. A UM-RTCOM model can also be used for several kinds of analysis WCET, deadlock freedom, and verification of liveness properties. The communication channel protocol modeled in UM-RTCOM has been tested in an actual sensor network deployment by Barbaran et al. [1]. The deployment shows the improvement of the middleware overhead compared to another deployment where the motes send the sensed data periodically to the actors.

I. eXtended Reactive Modules (XRM)

XRM is an extension language of Reactive Modules (RMs). Demaille, et al. used WSNs as a case study to evaluate the XRM modeling language [8]. The case study successfully used XRM to model multiple nodes as modules and was able to support several network issues such as communication capability, memory, and energy consumption. Model checking of XRM is possible via a transformation to the original RM language. RM modules can be used with the model checking tools PRISM and APMC [7].

III. MODELING AT THE NODE AND SENSOR LEVEL

In this section, we consider how the different modeling techniques represent WSN elements, including nodes, sensors, and hardware. In particular, we consider the modeling of node/sensor behavior, sensor data, and hardware components (see Table II).

The node behavior column tries to capture which particular characteristics that an approach focused on modeling. The sensor and hardware modeling column considers if the actual WSN hardware, such as the ADC, microprocessor, and the wireless channel are included in a model. Hardware modeling also considers the types of sensors that a modeling technique can represent.

A. Node Behaviors

Most of the modeling techniques use a form of component-based modeling to represent a sensor node. The WSN behavior is modeled by specifying the component's internal behavior, component to component interactions, and the communication channel's characteristics. It should be noted that the approaches could be divided into two distinct types. Those that focused on the augmentation of the models to capture particular features such as concurrency, event-driven behavior, and real-time behavior and those that leveraged standard models like state space and procedural coding that were later used for code generation or performance analysis. This seperation also lets us clearly see those approaches that have included concurrency, event-driven behavior, and real-time behavior, since these three features are crucial for WSN design.

The only technique that we felt did not model node behavior was the paper using the Promela Model Checker [19]. The authors of this work decided to simply focus on the modeling of network connectivity as opposed to including any significant modeling of the node behaviors. Promela though can be used to model and analyze node bahaviors.

B. *Modeling Sensors and Hardware*

Most of the modeling techniques surveyed can be used to create a platform independent model. However, even in a platform independent model there is a necessity to include some of the hardware details.

One of the reasons for including the hardware details is that the software in the nodes of a WSN is tightly coupled to the hardware elements of the node. Therefore the binding of software and hardware components should be represented in the model. An example of a hardware-software binding is the interaction between the sensor (e.g., humidity, temperature or moisture) and the software component that handles the readings. The sensor type is modeled as a component that uses a communication channel to transfer data to the software components [6].

Another reason that hardware information may need to be represented is to be able to generate source code from the model. Generated source code is interacting with the node hardware (timers, ports, sensor types) and therefore the model has to be aware of the hardware components in order to generate the correct code [15], [10].

Finally, hardware representation also helps in the analysis stage. For instance the ADC circuit has to be modeled to calculate the SNR, the communication channel has to be model to calculate the BER value [21].

Approach	**Node Behaviors**	**Sensors & Hardware Modeling**
HL-SDL [10]	concurrency, event-driven	-
Insense [6]	concurrency, real-time	sensor types
MathWorks [16]	procedural, state space	-
MDEA [15]	procedural, state space	timers, ports, wireless channel
PM [19]	-	-
SensorML [4]	event-driven	sensor types
SystemC-AMS [21]	procedural	ADC, microprocessor, wireless channel
UM-RTCOM [9]	concurrency, real-time, event-driven	-
XRM [8]	procedural, state space	-

Table II
MODELING OF WSN ELEMENTS

IV. MODELING AT THE SYSTEM LEVEL

This part of the paper focuses on modeling contributions to the distributed nature of sensor networks. The modeling techniques deal with various distribution issues, such as network behavior and topology modeling. The modeling techniques that deal with the network system are shown in Table III.

A. *Network Behavior*

Modeling network behavior in a WSN is crucial because many important performance values are based on the network. For example, the trade-off between packet loss and power. Due to the fact that the node has limited power resources, it is common to use a power management algorithm that controls the wake up state of a node from active to sleeping and vice versa. Packages can be lost if this is not done properly. XRM for example, calculates the package delivery probability. This can be helpful for applications in which the package deliverance is an important factor. Also related to power management, modeling the power consumed in the wireless communication process between the nodes is an important factor in increasing the life-time of the WSN. XRM models the power consumed by each wireless communication channel. Every time the node model is provoked to send or receive a signal, a specific amount is subtracted from the energy level [8].

Another example where modeling at the network behavior is important is in capturing the deployment and interaction of the software components across the network. For example, MDEA divides the software elements into two groups, those residing on the nodes and those residing on the gateway. The generated code should guarantee the interaction between the node and the gateway [15].

In a similar fashion UM-RTCOM models the network elements (sensors, actors, and the gateway) as three virtual machines (VMs), where each VM models a single element. The system behavior is modeled by the interaction between the three VMs [9].

B. *Topology Modeling*

This section focuses on how the topology is modeled, in the other words how the physical locations of the nodes have been modeled. The topology of WSN systems can be dynamic or static. The static topology represents the nodes in a fixed location while the dynamic topology represents the nodes while they are in a moving state. Ad-Hoc SNMC captures the dynamic topology by recording the physical location of the nodes in a Location Manager (LM). While the nodes change their physical location, they send the updated location to the LM. Through the use of model checking, the nodes connectivity can be checked [19]. Additionally, based on the modeling target, the technique models the number of hops in the network design. For instance, SystemC-AMS analyzes the communication channel between two nodes,

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

Approach	**Network Behavior**	**Topology Modeling**
HL-SDL [10]	-	-
Insense [6]	-	-
MathWorks [16]	Node/base station interaction	Single hop, static topology
MDEA [15]	Node/base station interaction	-
PM [19]	Nodes connectivity	Multi hop, dynamic topology
SensorML [4]	-	-
SystemC-AMS [21]	-	Single hop, static topology
UM-RTCOM [9]	Nodes/actors/base station interaction	Single hop, static topology
XRM [8]	Power management-wake up states	Single hop, static topology

Table III
MODELLING AT THE SYSTEM LEVEL

Approach	**Code Generation**	**Model Checking**	**Execution & Analysis**
HL-SDL [10]	NesC	-	WCET
Insense [6]	C	Spin (Channel protocol)	WCS
MathWorks [16]	NesC, C	-	functional analysis
MDEA [15]	NesC	-	-
PM [19]	-	Spin (Connectivity)	-
SensorML [4]	JavaBeans [14]	-	-
SystemC-AMS [21]	-	-	BER, SNR
UM-RTCOM [9]	-	-	Deadlock, WCET
XRM [8]	-	Prism, APMC	execution, debugging

Table IV
SUPPORTING TOOLS

such that the model deals with single hop communication issues between two nodes.

XRM is an example of modeling for static topologies. The topology is modeled as a grid location. Each node location is modeled as an X-Y variable [8].

In MathWorks, the framework is able to model the static topology by modeling the nodes with state chart and the communication medium which is implemented in C, models the connectivity between the network node [16]. In the UM-TRCOM model, single hop communication is used between the network nodes because of the application requirements and nature of the problem. behavior is modeled by the interaction between the three VMs [9].

V. SUPPORTING TOOLS

Almost all of the modeling techniques surveyed offers some tools to support the design of WSNs. In this section we discuss support tools that include code generation, execution and analysis, and model checkers (see Table IV).

A. *Code Generation*

Code generation is the process of generating source code from a model or another source code representation. Tools for generating source code from WSN models are beneficial with respect to design for two main reasons:

1) Implementing the source code for the nodes is tedious, time consuming and requires a lot of time and effort from the developers.
2) Debugging the design at the source code level is also a very challenging and time consuming process.

Modeling can help to solve these problems by designing the system at higher abstraction layers and generating the target code from that layer [15]. The simplicity of the code generation process depends on the degree of similarity between the modeling notation and the generated code notation. The Mathwork technique generates nesC code and C code from ANSI C modeling notation. Generation of C code is developed through minor changes in the modeling notation versus the generation for nesC needs a lot of the changes for ANSI C to generate the proper code [16].

One criticism of code generation tools is that the code produced is not as efficient as hand-written source code. Manual optimization by the user is one solution to achieving better performance from generated source code [16]. An example of manual optimization of WSN source code is modifying the communication between the components from asynchronous in the model to synchronous in the target platform. In addition to manual optimization of the generated source code, simulation can be used at the model level to refine the model (with respect to performance) prior to code generation [10].

Our survey reviewed four modeling techniques which are capable of generating source code: MDEA, HL-SDL, Insense and MathWorks. MDEA and HL-SDL can generate nesC code for WSNs [15]. MathWorks generates nesC as well as C code that executes under the MANTIS operating system [16] while Insense generates C code [6].

B. *Model Checking*

Model checking is a formal methods technique for software engineering [5]. A model checker takes as input a model of a system and a property specification. The model is converted into a finite state model and the model checker uses an exhaustive state space search to verify the specification. The model checker will determine if the model satisfies the specification. If it does not, than a counter example (error trace) may be provided.

Applying model checking to WSN models allows the designer to verify that the design is correctness as well as detect potential errors. In response to errors, the model can be modified and the design improved prior to implementation. Model checking for WSNs can be classified as direct and indirect model checking. Direct model checker occurs when a model checker exists that can take the WSN model as input. An example of direct model checking is in PM where the modeling language, Promela, is also the input language for the model checker Spin [19].

Indirect model checking occurs when no model checker exists for the WSN modeling language and model transformation is required in order to transform the WSN model to a language that can be input to a model checker. For example, XRM models need to be transformed into RM in order to be used with the model checkers PRISM and APMC [8]. A drawback of indirect model checking approaches such as the one used in XRM is that the model checking results are given with respect to the RM model and need to be transformed back into an XRM form. The challenge of transforming between the WSN modeling language and the model checker input language is know as the semantic gap problem. Another example of indirect model checking is in IM where the authors manually created a Promela model of the component communication channel in order to verify the correctness of the communication protocol. Their verification identified an error that lead to a modification to the original Insense model [20].

Model checking has also been used to check the network topology of WSNs [3]. However, we have excluded this work from our survey since our primary focus is on software design.

C. Model Execution and Analysis

In addition to code generation and model checking we also consider other tool support including tools that execute and analyze the WSN models. Model execution refers to the execution or interpretation of the WSN design at the model level. XRM is the only techniques in our survey supports model execution. The XRM compiler, a domain specific compiler, allows for model execution and debugging as well as model optimizations (e.g., dead code removal) [8]. Model analysis includes a variety of static and dynamic techniques and a number of the approaches in our survey include some kind of analysis tool. We will not discuss some of the analysis provided by these tools.

Real-time behavior is an important system requirement for WSANs and real-time design often includes schedulability analysis. Schedulability analysis includes WCET (the maximum time length taken to execute a process), WCS (the sum of the space requirements of each component's parameters), and deadlock analysis [6] [9]. HL-SDL, Insense and UM-RTCOM provide some form of schedulability analysis (see Table IV).

Other modeling techniques to include analysis tools are MathWorks and SystemC-AMS. MathWorks includes functional analysis of the WSN algorithms. SystemC-AMS calcules factors to judge the electronic system of the nodes by simulation. The first factor is SNR for the ADC process inside the node. The second factor is BER for the wireless communication channel [21].

VI. Conclusions and Future Directions

Modeling helps to resolve some of the WSN software implementation challenges before deployment.

As depicted in Table I, several approaches have focused only on the modeling of software elements in a sensor node while others also model the sensor network. Both of these are important to be modeled from a sensor network perspective. Also many of the modeling languages support components as one of its basic modeling elements. Component based modeling is a fundamental way to partition software entities because it can be used to support multi-threading design and analysis. Most of the approaches reviewed can also model the communication channel. A few like PM and SensorML can model a process. Process modeling is important in capturing a systems behavior.

As depicted in Table II, the modeling techniques are targeted to analyze specific software challenges like concurrency, real-time, and event modeling. We also see that they may be focused on simply modeling the sensory information or hardware to facilitate code design as is the case with MDE and SystemC-AMS.

Modeling at the system level is also another feature that some modelers support. As seen in Table III, several modeling techniques like UM-RTCOM, XRM, PM, MDEA, and MathWorks can all model the sensor network but there is a focus for each on what behavior they model. They all model node activity and take into account node to node communication but not all can explicitly model the network topology as is the case with MDEA.

Support for analysis and code generation tools is vital for a modeling technique. Table IV reflects, the fact that certain modeling technique can do code generation while others cannot. This depends primarily on the focus of the developers of the modeling technique and the maturity of the approach. It should be noted that very few of the techniques support model checking. We speculate that this is largely due to the gap between the designing process and the model checking. This gap exists because the design takes place in domains such as CORBA and UML. In order to check the model with model checking methods, the design has to be re-modeled again in the model checking domain.

As a future direction for WSN modeling, enhancements for code generation tool are required. Such enhancements can improve the quality of the generated code in terms of the code size and avoidance of manual optimization for the generated code. Additionally, the modeling domain should be selected such that model checking can be done without redoing the model in the model checking domain. Moreover, the modeling domain should support analysis at the design stage, which helps the software system developer detect and correct software system problems at an earlier stage of the sensor system design. Some of the reviewed papers have performed analysis for WCET, WCS, deadlock, SNR for sensor interfaces, and BER for the communication channel. To the best of our knowledge, package delay and data losses have not been considered in the analysis but these factors are important for sensor networks.

REFERENCES

[1] J. Barbaran, M. Diaz, I. Esteve, D. Garrido, L. Llopis, B. Rubio, and J. Troya. Tc-wsans: A tuple channel based coordination model for wireless sensor and actor networks. In *Computers and Communications, 2007. ISCC 2007. 12th IEEE Symposium on*, pages 173 –178, 2007.

[2] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han. MANTIS OS: An embedded multithreaded operating system for wireless micro sensor platforms. *Mobile Networks and Applications*, 10:563–579, 2005.

[3] S. Bhatti, J. Xu, and M. Memon. Model checking of a target tracking protocol for wireless sensor networks. In *Proc. of IEEE 10th Int. Conf. on Computer and Information Technology (CIT'10)*, pages 2867–2872, Jul. 2010.

[4] M. Botts and A. Robin. OpenGIS sensor model language (SensorML) implementation specification. Technical report, OGC, Jul. 2007.

[5] E. M. Clarke Jr., O. Grumberg, and D. A. Peled. *Model Checking*. The MIT Press, 1999.

[6] A. Dearle, D. Balasubramaniam, J. Lewis, and R. Morrison. A component-based model and language for wireless sensor network applications. In *Proc. of 32nd Annual IEEE Int. Conf. on Computer Software and Applications (COMPSAC '08)*, pages 1303–1308, Aug. 2008.

[7] A. Demaille. Probabilistic verification of sensor networks. In *Proc. of 4th IEEE Int. Conf. on Comp. Sci., Research, Innovation and Vision for the Future (RIVF'06)*, pages 45–54, 2006.

[8] A. Demaille, S. Peyronnet, and B. Sigoure. Modeling of sensor networks using XRM. In *Proc. of 2nd Int. Symp. on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2006)*, pages 271–276, Nov. 2006.

[9] M. Diaz, D. Garrido, L. Llopis, B. Rubio, and J. Troya. A component framework for wireless sensor and actor networks. In *Proc. of IEEE Conf. on Emerging Technologies and Factory Automation (ETFA '06)*, pages 300–307, Sept. 2006.

[10] D. Dietterle, J. Ryman, K. Dombrowski, and R. Kraemer. Mapping of high-level SDL models to efficient implementations for TinyOS. In *Proc. of Euromicro Symp. on Digital System Design (DSD 2004)*, pages 402–406, Aug.-Sept. 2004.

[11] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proc. of 29th Annual IEEE Int. Conf. on Local Computer Networks*, pages 455–462, Nov. 2004.

[12] ITU-T. *Specification and description language (SDL)*, z.100 (11/99) edition, 1999.

[13] P. Levis. *TinyOS Programming*. Cambridge University Press, 2009.

[14] R. Liscano and K. Kazemi. Integration of component-based frameworks with sensor modelling languages for the sensor web. In *Proc. of IEEE GLOBECOM Workshops (GC Wkshps)*, pages 235–240, Dec. 2010.

[15] F. Losilla, C. Vicente-Chicote, B. lvarez, A. Iborra, and P. Snchez. Wireless sensor network application development: An architecture-centric mde approach. In *Software Architecture*, volume 4758 of *Lecture Notes in Computer Science*, pages 179–194. 2007.

[16] M. Mozumdar, F. Gregoretti, L. Lavagno, L. Vanzago, and S. Olivieri. A framework for modeling, simulation and automatic code generation of sensor network application. In *Proc. of 5th IEEE Comm. Soc. Conf. on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'08)*, pages 515–522, Jun. 2008.

[17] L. Necchi, A. Bonivento, L. Lavagno, A. Sangiovanni-Vincentelli, and L. Vanzago. E2rina: an energy efficient and reliable in-network aggregation for clustered wireless sensor networks. In *Wireless Communications and Networking Conference*, pages 3364 –3369, 2007.

[18] B. Nickerson and J. Lu. A language for wireless sensor webs. In *Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on*, pages 293 – 300, May 2004.

[19] V. Oleshchuk. Ad-hoc sensor networks: modeling, specification and verification. In *Proc. of 2nd IEEE Int. Work. on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pages 76–79, Sept. 2003.

[20] O. Sharma, J. Lewis, A. Miller, A. Dearle, D. Balasubramaniam, R. Morrison, and J. Sventek. Towards verifying correctness of wireless sensor network applications using Insense and Spin. In *Model Checking Software*, volume 5578 of *Lecture Notes in Computer Science*, pages 223–240. 2009.

[21] M. Vasilevski, N. Beilleau, H. Aboushady, and F. Pecheux. Efficient and refined modeling of wireless sensor network nodes using SystemC-AMS. In *Conf. on Ph.D. Research in Microelectronics and Electronics (PRIME 2008)*, pages 81–84, Apr. 2008.

Wireless CiNet Network Analysis and Diagnostics Using Neighbourtables

Ismo Hakala and Timo Hongell
University of Jyväskylä / Kokkola University Consortium Chydenius
P.O.Box 567, FI-67701, Kokkola, Finland
Email: {ismo.hakala, timo.hongell}@chydenius.fi

***Abstract*—Reliable communication is crucial for successful deployment of a wireless sensor network. By taking into account communication and other wireless sensor networks constraints, networks can be analyzed efficiently. The diagnosis of the network's operation can be done using the information gathered by the sensor nodes in the network. This paper discusses wireless sensor network diagnostics describes so-called neighbourtables used to collect diagnostic data of the wireless CiNet network and their construction. The information stored in neighbourtables can be used to monitor network behavior and to improve networks' packet routing and fault recognizing for the nodes, in both, nodes' and centralized applications' point of view.**

***Keywords*-neighbourtables, wireless sensor network, diagnostic, management**

I. Introduction

In recent years, study of wireless sensor networks (WSN) has become a rapidly developing research area. A WSN is a set of wireless sensor nodes where each node measures a physical value using selected sensor probes and sends the value to a database through specific sink nodes. Nowadays, WSNs are widely used in civil and industrial applications such as smart home or environment monitoring [1], [2], [3]. Compared to traditional sensing methods, wireless sensor networks technology offers some benefits: wide areas can be covered with inexpensive, energy-efficient battery-powered devices, which make long-term monitoring and real time access to measuring data possible. Often the nodes of WSN also are able to self-configure themselves, which enables quick and easy system deployment.

The use of WSN-applications also reveals many different constraints that can decrease the possible number of real application deployments. These constraints may be defined in different categories based on the constraints, such as the system's memory, processors' limitations and energy consumption. Out-of-date communication equipment and their bandwidth as well as physical environmental and measurement factors related to sensors' location and calibration also may be the cause of different constraints. A WSN can take into account some of the possible constraints by gathering diagnostic information from the network. For example, radio link quality is affected by many internal and external factors, and it can be evaluated by using the Received Signal Strength Indicator (RSSI). RSSI values as well as other diagnostic data, such as battery levels, the number of received packets etc., can be stored in one table, the so-called neighbourtable.

In WSN nodes, the diagnostic data collected to the node's neighbourtables, gives the nodes direct information about the network around them and also information about the sink node. The node can directly use the neighbourtable information to determine its routing and clustering possibilities based on the collected and calculated information about the neighbour nodes. Since the nodes' neighbourtables can be centrally collected to a server database, they can be utilized in different diagnostic applications. For example, we have developed a graphical real time application, CiNetView, to make wireless sensor network deployment and monitoring easier. The application visualizes, in real time, the nodes' relative locations as well as shows the links' quality, which make the deployment of WSN much quicker and easier.

This paper discusses reasons for wireless sensor network diagnostics and describes the so-called neighbourtables used in the wireless CiNet sensor network. The paper is organized as follows: First, we provide a brief description of some related research and then discuss the useful diagnostic data and neighbourtables in WSN. Section IV presents the neighbourtables construction in CiNet network as well as their usage. Other functionalities of the tables are also described in more detail. A survey of the neighbourtables' energy cost is presented. Finally, some experiences of the use of neighbourtables in real wireless sensor network implementations are discussed.

II. Related work

Analysis and monitoring of WSNs are highly evolving research topics in the field of wireless technology. However, real time performance of wireless communication is not that widely studied. In [4], Meier et al. discuss link behavior and metrics that can be used to evaluate link performance. They have used statistical link analysis in their studies. Ferrari et al. have done indoor performance

ISBN: 978-1-61208-144-1

studies of WSNs [5]. Sensor network diagnostics and visualization are discussed in [3]. However, there has not been much discussion on real time inspections of the changes in link qualities, whether uplink or downlink, in sensor networks. There are some commercial network visualization and diagnostic applications, for example, MOTE-VIEW [6] and Surge View [7], developed by Crossbow Technology. One approach that uses additional messages to construct and utilize neighbourtables is considered in [8]. Routing using neighbourtables is studied in [9] and [10]. Jacuot et al. have discussed indirect diagnosis of the node state with few messages, using an SNMP-like LiveNCM management tool [11].

III. Network variables and diagnostic

For a customer's point of view, wireless network is fine when the wanted information is received and processed correctly, that is, when the network works properly. On the other hand, wireless networks include large amounts of all kinds of data that can be used to give useful information about the networks' operational performance and reliability, not only for the user or customer, but also for the networks' maintenance crew. Due to the limited amount of memory and space in the sensor nodes and the transmitted data frames, it is reasonable to define the relevant information and metrics that can be used in WSN diagnostics. Fynn [12] has done a collective study of WSN performance analysis methods and metrics, the topics including storage, routing, real time communication, power management and architecture.

The most important information from the node's point of view are its neighbours' addresses, since without them other information cannot be linked to a specific node and it is almost impossible to perform any data routing. The sink node is defined to be the so-called root of the network, and its relative location need also be known. Therefore, a metric called hop count is essential information for the nodes. The bigger the hop count the greater the number of relaying nodes that are needed for sending packets between the specific sensor node and the sink node. This value does not necessarily directly indicate the physical distance, but rather the relative length of a path that a data packet needs to be forwarded to reach the sink node. The hop count value tells the node the direction of the sink that the packet is to be sent to with the minimum number of forwarding retransmissions. Hop count values can also be defined to be calculated from the nearest gateway of the node.

Radio link quality is affected by many factors, which can be divided into a device's internal and external factors. The internal factors are caused by imperfections of the device's hardware or software. E.g., different radio chips do not behave exactly in the same way and each node has its own radiation pattern that is not uniform [2]. The external factors, such as fading, shadowing, multipath propagation, and dynamic environmental factors affect wireless communication and make it difficult to predict the radio performance beforehand. Link quality can be evaluated by using the Received Signal Strength Indicator (RSSI), which indicates the strength of the radio signal between two nodes at the receiver's position.

RSSI values can be used to determine whether the link is acceptable or not. The nodes typically have been programmed to respond to a predefined RSSI lower bound to determine whether the link is strong enough to be useful. In WSNs, radios typically operate in the 2.4GHz ISM band and are based on the IEEE 802.15.4 standard due to which RSSI value -85dBm is considered to be the acceptable lower bound. It is also possible that the link quality may only be suffering from temporary deterioration, for example when people suddenly walk between the nodes. Therefore the neighbour RSSI values also need to be averaged to avoid useless routing changes. Since the data packets' path RSSI value, from node to sink or vice versa, cannot be any larger than the worst link's case, the path RSSI values indicate the lowest RSSI value between the sink and the node. This information can directly be used in packet sending and routing decisions. Meier et al. [4] have used RSSI values too, but also, e.g., number of packets, average packet reception rate (PRR) and link quality indicator (LQI) to perform efficient WSN link diagnostic.

When sent packages are spread from the sink node to the last node in the network, it is possible that some of the packages arrive through multiple paths and are somewhat delayed. Nodes need to be able to recognize the packets that have been sent on the same synchronization period. Therefore, every packet has its own sequence number, which also needs to be stored. Some of the packets may be received from different neighbour nodes, meaning that the packets have traveled through different paths from sink to the receiver node. Packet routing is also one interesting network management related topic that will be discussed later.

Wireless nodes naturally need energy to operate, and they usually are battery powered. The varying shape and utilization of the network cause that the battery levels of the nodes do not consume at the same rate all over the network. Therefore, it is essential to know that how much the batteries have power left.

In addition to these considerations, the nodes can be programmed to have several different counters that can be set to count, for example, the number of received and missed packets. These counters can then be used to calculate, e.g., throughput and reliability values of the sensor or relay nodes. All this information needs to be stored somewhere. One solution is to use neighbourtables.

Typically, when talking about neighbourtables, people are thinking about routing tables. A short comparison of neighbourtables and routing tables is needed. A routing table is a set of rules that is used to decide where data packets traveling over network will be directed. Each packet contains information about its origin and destination. The routing table contains the information necessary to forward a packet along the best path toward its destination. Usually a routing table includes the following information:

- Destination: The address of the packet's final destination
- Next hop: The address to which the packet is to be forwarded
- Metric: Assigns a cost to each available route so that the most cost-effective path can be chosen

Our definition of neighbourtable is that neighbourtables basically include, not only the same information as routing tables, but also some additional information. They can be used to aid routing decisions, but they can also be used in different diagnostic and management solutions as well. For a single node's point of view, a neighbourtable is a multifunctional set of information about the nodes' neighbour nodes and the links between them. Globally speaking, the neighbourtable file, constructed by the server, includes all essential information about the whole network, and the whole network's operation can be diagnosed and monitored in real time from there.

IV. The CiNet Neighbourtable

We are using CiNet nodes [2][13] in our study. CiNet is a research and development platform for the WSN implemented in Kokkola University Consortium Chydenius. The hardware in the CiNet node is specially designed for WSNs and consists of inexpensive, standard off-the-shelf components. The CiNet node includes all the basic components necessary for WSNs. In our CiNet, the nodes use cross-layer architecture [13].

The main idea of the cross-layer architecture is to implement a wireless sensor network's basic tasks, such as topology management and power saving functionalities, as separate protocols in a cross-layer management entity. Data structures, which are in common use, are in this study implemented in the cross-layer management entity as a neighbourtable data addition. The use of a cross-layer implementation reduces computational and memory requirements - not all the information needs to be transmitted between application interfaces and protocol layers. The architecture also allows the implementation of the application and protocol stacks be as simple as possible, since they are practically free of the tasks related to network management.

In every node, the neighbourtable is stored to a one common data storage in the cross-layer management entity, where all the protocol stack layers can utilize the same information, see Figure 1. This reduces the total amount of memory storage space needed, but the use of cross-layer architecture causes challenges related to maintenance, which have to be considered in the implementation. The problem has been approached by using message multiplexing in the data link layer and modular structure in the cross-layer management entity.

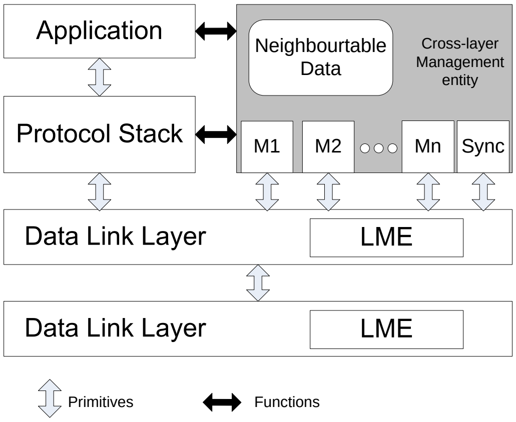


Figure 1. CiNet network's cross-layer architecture.

Basically the neighbourtable of each node consists of d levels with b entries at each level. More precisely, every level d is a different neighbour of the node and each entry b includes stored information, such as node ID, battery level, RSSI and hop count.

All the nodes' neighbourtables are also collected to a single data file on a server, from where all the information can be retrieved. This centralized neighbourtable can be used in different WSN management and diagnostic applications and tools. The format of the file is shown in Table I.

Table I
Format of the server's neighbourtable data file.

Seq No	Node ID 1	Neigh. 1 data	Neigh. 2 data	···
Seq No	Node ID 2	Neigh. 1 data	Neigh. 2 data	···
⋮	⋮	⋮	⋮	⋮
Seq No	Node ID N	Neigh. 1 data	Neigh. 2 data	···

A. Neighbourtable construction

In a CiNet network, each node constructs and maintains its own neighbourtable, as defined in Table II, in which the node stores information about its neighbours, which are the nodes that it hears. The neighbourtable of each node is updated in every synchronization period of the network, see Figure 2. The neighbourtable construction and update is defined to be one part of the synchronization protocol. The sink node broadcasts the synchronization message isotropically, and every node that hears it broadcasts that message onwards through the network during a predefined time period. In this way the whole network can be synchronized. The synchronization frame structure is shown in Figure 3(a). The SYNC frame also includes additional data that is not used in the neighbourtables, such as the CMD and a command byte that can be used to control the nodes' operation. Before relaying the synchronization message, the nodes update it with their own information. Based on the received synchronization messages and the data included in the synchronization frames, the nodes update their neighbourtables. Note that the synchronization messages are heard by all the nodes' neighbours, including the predecessors, which means that the neighbourtable information can be collected in both directions. To prevent any ping-pong effect, the nodes broadcast the synchronization message only once in every synchronization period. Continuous synchronization of the network is vital to ensure valid operation of the network.

Table II
CiNet nodes' neighbourtable

U16INT u16NbAddr	Neighbor address
S8INT s8RSSI	Neighbor link RSSI value
U8INT u4Bat:4	Battery level of the neighbor
U8INT u4HopCnt:4	Hopcount of the neighbor
U8INT u8NodeType	Sink, Relay, Sensor
U16INT u16Received	Number of received sync packets
U16INT u16Missed	Number of missed sync packets
S8INT s8AvgRSSI	Avg RSSI of the neighbour link
S8INT s8PathRSSI	Path RSSI (path's weakest RSSI)
U8INT u8PrevSeq	Previous sequence number
U8INT u8Ntp	UpLink throughput
U8INT u8UplinkTp	Path throughput
U32INT u32NbLastSeen	Last seen time, for entry maintain
U8INT u8Status	Sync status

After every synchronization period, every node in the network now has real time information about the network around, including information about the sink node. After the synchronization period, the nodes also can send the neighbourtable information through the sink node to the server during the management period. The time interval of this neighbourtable update can be defined to meet the application demands. A minimum update interval is one synchronization period, but it can also be much longer. The management frame includes a section where the neighbourtable information is sent. The management frame structure is shown in Figure 3(b) Diagnostic and other management data is sent (and acknowledged) as a unicast transmission through a selected route to the sink node. If retransmissions are not needed, each management frame is sent once in every synchronization period.

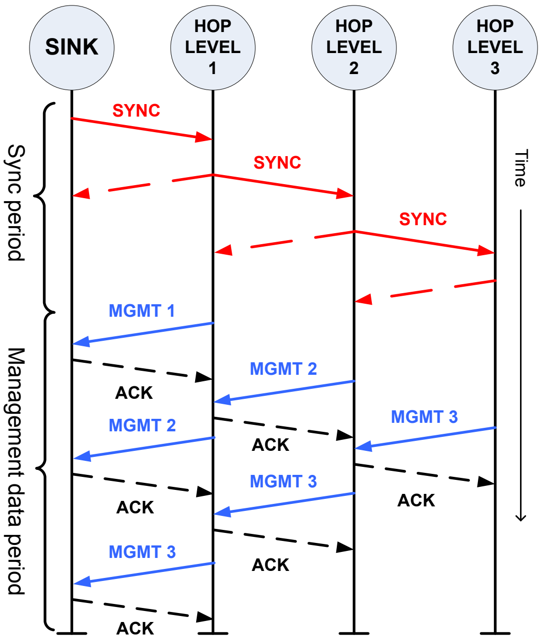
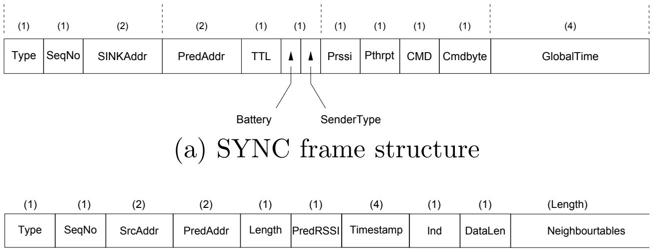


Figure 2. Synchronization, management and ACK messages during one synchronization period.



(a) SYNC frame structure

(b) MGMT frame structure

Figure 3. Synchronization and management frame structure

B. Neighbourtable utilization

Neighbourtables are specially used for collecting information for real time deployment and for monitoring of the WSN. They can be locally used in the nodes or with some management tool. Since every neighbourtable of every node is known, it is possible to count different diagnostic values using the information collected to the neighbourtables.

One of the most useful values that can be calculated from the neighbourtable information is the throughput of one link. The throughput value can directly indicate the reliability and robustness of the sensor node or a link between two nodes. A good throughput values should be near 100%, but in WSNs typically at least some of the packets are missed. Since network synchronization is done periodically, the node knows how many times it should have heard the synch message after having received the first synch message. The sequence number

ISBN: 978-1-61208-144-1

of the received message is checked, and if the number has increased more than one, then a packet or packets have been missed. The throughput information is calculated based on the nodes' packet counter values.

When a node has been working for a while, the throughput value will settle to some level that will indicate the basic information about the links reliability. If the throughput value suddenly begins to go lower, it will be a clear indication about something having changed or broken in the network, for example, a car might have been parked between two nodes, interfering with the signal.

A node's battery level information can be used to alert the network supervisor to do network maintenance. If a low battery level is noticed before a link stops working, due to power out, it is possible to keep the network topology controlled. Another point is that if the node's battery level is decreasing more rapidly than in the other nodes, it is possible that that the node is not working correctly, or that it is relaying too much data.

For packet routing, nodes typically use basic routing tables. Using the routing table, nodes can efficiently transfer data from each node to a sink, but the routing decisions can be thoroughly justified with the extended information of the neighbourtables. Naturally, the nodes' address information is the most important, but other information can be used to optimize the networks' routing performance and reliability. Almost all the information that the neighbourtable contains can be used to improve the network's data routing.

In our solution, the routing protocol uses hop count and RSSI values. The minimum hop metric is used in many routing protocols due to its simplicity and isotonicity [14]. Hop count value can directly indicate the logically shortest transmission path from a node to a sink. RSSI values also indicate the links' or the whole paths' quality. In order to maximize packet throughput, it is reasonable to choose links that are more likely by the next relay. Throughput values indicate the total amount of packets that have successfully been transmitted through a specific link. If a link has a good throughput value, it is more likely to perform the transmission successfully again. The battery levels of the nodes' neighbours can also be used to define the data routing. If the battery level of one node is getting low, then an alternatively routing should be used, if possible, to avoid unnecessary dropouts.

Transmitting power for the nodes' radios can also be optimized using the RSSI information in the neighbourtables. In some applications and hardware solutions, it is possible to adjust the radio chips' transmitting power in real time. An acceptable lower bound of RSSI can be fixed, and, based on it, the transmitting powers can be lowered or increased when necessary.

V. The Cost of Neighbourtables

In CiNet network, the nodes are synchronized periodically to ensure valid operation. The length of the synchronization period, during which the neighbourtables are collected, depends on the application used and on the WSN measurement solution. Thus, data collection is embedded to the synchronization messages, and no extra messages are needed to be sent to collect the neighbourtable information. The maximum size of the SYNC frame is 16 bytes. Of these, 4 bytes are directly related to the neighbourtable usage. It can be stated that basically almost all information in the SYNC frame would be sent even without the neighbourtable usage, since the information is used for routing protocols in any case. Therefore, it can be said that the neighbourtables are filled with almost free of additional energy cost. Only the size of the synchronization frame is increased.

As every node can store the information of eight neighbours and as 18 bytes of memory have been reserved for each neighbour, this means a maximum memory use of 144 bytes (8 x 18 bytes). From these, the six best are sent in the management phase (one management frame can fit six neighbours).

The main additional cost of the neighbourtables incurs when the tables are also sent to the sink node as a part of the management frame. This increases the number of sent packets and time to spend for sending the data. It also consumes more energy. The total cost of one sent and received management frame is defined as $E_{frame} = \Delta E_{tx} + \Delta E_{rx}$. Our measurements have shown that a management frame transmission takes about 0.216 mJ of energy and receiving takes about 0.142 mJ, so the total consumption is about 0.4 mJ [15], [16].

The energy overhead of the whole network to transmit the neighbourtables to the sink node depends on many different factors. These include, for example, the size of the network used, the number of hops and the application that determines the number of sent and relayed packets. The transmission time of one node is determined by the size of the sent packet. The node's transmission power and battery voltage also affect energy consumption. The size of one transmitted management frame that includes the neighbourtables is between 50 and 128 bytes, depending on the number of the node's neighbours.

VI. CiNetView; A Neighbourtable utilization tool

Neighbourtables can be used to help sensor network diagnostic and visualization. We have been using neighbourtables in our CiNetView application [17]. The CiNetView application is a graphical tool for making the deployment and monitoring of a WSN easier and more assured. CiNetView is based on diagnostic information

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

that the nodes have collected and stored to neighbourtables. The application is server-centralized and it reads information from the neighbourtable file. The application displays network topology based on relative locations produced by the MDS-algorithm. It can also use real background images and maps, where the user can exactly pinpoint the nodes' true locations. CiNetView displays the essential network diagnostic information and helps the user to see the changes in the network's behaviour. Because of the real time presentation of the network's connections and the quality of these connections, the advantages of this application can be seen most clearly in the network's deployment phase. The application can be used to diagnose and monitor a WSN through the network's lifetime.

VII. Conclusions

In this paper, we have discussed the main topics about wireless sensor network diagnostics and defined the essential metrics for WSN diagnostics. The presented neighbourtables are constructed, without significant additional transmission overhead, using modified network synchronization messages and in every node they are stored to a common data storage in the cross-layer management entity, where all the protocol stack layers can utilize the same information. Neighbourtables can be used in WSN diagnostic and management.

For future work the idea is to improve the utilization of the neighbourtables from the nodes' point of view and in general diagnostic applications. The goal is to make a database that collects historical data from the wireless sensor network and can be used in backtracking errors that have been noticed in the network.

References

[1] Y. Chen, J. Chiang, H. Chu, P. Huang, and A. Tsui, "Sensor-Assisted WI-FI Indoor Location System for Adapting to Environmental Dynamics," in *Proceedings of the 8th ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Montreal, Quebec, Canada, October 10-13 2005.

[2] I. Hakala, J. Ihalainen, I. Kivelä, and M. Tikkakoski, "Evaluation of Environmental Wireless Sensor Network - Case Foxhouse," *International Journal on Advances in Networks and Services*, vol. 3, no. 1 and 2, pp. 22 – 32, September 2010.

[3] Y. Hu, D. Li, X. He, T. Sun, and Y. Han, "The Implementation of Wireless Sensor Network Visualization Platform based on Wetland Monitoring," *Second International Conference on Intelligent Networks and Intelligent Systems*, 2009.

[4] A. Meier, T. Rein, J. Beutel, and L. Thiele, "Coping with Unreliable Channels: Efficient Link Estimation for Low-Power Wireless Sensor Networks," *5th International Conference on Networked Sensing Systems, INSS 2008*, pp. 19 – 26, 2008.

[5] G. Ferrari, P. Medagliani, S. Di Piazza, and M. Martalò, "Wireless Sensor Networks: Performance Analysis in Indoor Scenarios," *EURASIP Journal on Wireless Communications and Networking*, 2007.

[6] M. Turon, "MOTE-VIEW: A Sensor Network Monitoring and Management Tool," *The Second IEEE Workshop on Embedded Networked Sensors*, pp. 11 – 18, 2005.

[7] *Wireless Sensor Network: Getting Started Guide*, Crossbow Technology, Inc., Crossbow Technology, Inc. 4145 N. First Street, San Jose, CA 95134, September 2005.

[8] H. Liu and S. S. Lam, "Neighbor Table Construction and Update in a Dynamic Peer-to-Peer Network," *23rd International Conference on Distributed Computing Systems*, pp. 509 – 518, 2003.

[9] Z. Yao, J. Jiang, P. Fan, Z. Cao, and V. O. K. Lit, "A Neighbor-Table-Based Multipath Routing in Ad Hoc Networks," *The 57th IEEE Semiannual Vehicular Technology Conference*, vol. 3, pp. 1739 – 1743, 2003.

[10] C.-S. Nam, H.-Y. Cho, and D.-R. Shin, "Efficient Path Setup and Recovery in Wireless Sensor Networks by using the Routing Table," *2nd International Conference on Education Technology and Computer (ICETC)*, vol. 4, pp. V4–156 – V4–159, 2010.

[11] A. Jacuot, J.-P. Chanet, K. M. Hou, X. Diao, and J.-J. Li, "Livencm : A new wireless management tool," *IEEE AFRICON 2009*, pp. 1–6, September 2009.

[12] A. Fynn, "Performance analysis of wireless sensor networks," Washington University in St. Louis, Available at: http://www1.cse.wustl.edu/j̃ain/cse567-06/sensor_perf.htm, Tech. Rep., 2006.

[13] I. Hakala and M. Tikkakoski, "From vertical to horizontal architecture: a cross-layer implementation in a sensor network node," *Proceedings of the first international conference on Integrated internet ad hoc and sensor networks*, vol. 138, no. 6, 2006.

[14] W. Dargie and C. Poellabauer, *Fundamentals of Wireless Sensor Network: Theory and Practice*, ser. Wiley Series on Wireless Communications and Mobile Computing, X. Shen and Y. Pan, Eds. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom: John Wiley & Sons Ltd, 2010.

[15] T. Instrumens, *Data Sheet for CC2420 2.4GHz IEEE 802.15.4 RF transceiver*, CHIPCON, Available at: http://focus.ti.com/lit/ds/symlink/cc2420.pdf.

[16] I. Kivelä, C. Gao, J. Luomala, J. Ihalainen, and I. Hakala, "Design of Networked Low-Cost Wireless Noise Measurement Sensors (ISSN 1726-5479)," *International Journal on Sensors & Transducers*, vol. 9, no. Special Issue, pp. 171–190, February 2011.

[17] I. Hakala, T. Hongell, and J. Luomala, "CiNetView - Graphic Interface for Wireless Sensor Network Deployment and Monitoring," *Proceedings of The Fourth International Conference on Sensor Technologies and Applications SENSORCOMM*, pp. 395 – 401, July 2010.

Machine Learning and Dataming Algorithms for Predicting Accidental Small Forest Fires

Vasanth Iyer*, S. Sitharama Iyengar†, N. Paramesh §, Garmiela Rama Murthy*,
Mandalika B. Srinivas‡
*International Institute of Information Technology, Hyderabad, India - 500 032
†Louisiana State University, Baton Rouge, LA 70803, USA
§University of New South Wales, Sidney, Australia
‡Brila Institute of Technology & Science, Hyderabad Campus, Hyderabad-500078, India

vasanth,rammurthy{@research.iiit.ac.in},iyengar@csc.lsu.edu,paramesh@cse.unsw.edu.au,srinivas@bits-hyderabad.ac.in

Abstract—**Extracting useful temporal and spatial patterns from sensor data has been seen before, the technical basis of Machine learning with Data mining is studied with the evidence collected uniformly over many years and which allow using users' perspective in collected evidence. This model helps in probabilistically forecasting fires and help forest department in planing day to day schedules. Using a model to predict future events reliably one needs to collect samples from sensors and select a feature, which does have any particular bias. Due to practicable problems most of the collected data have 80% of attributes missing and the remaining has numeric values, which are hard to discretization. To adapt to such limitations, we use nominal data type, which allows better understanding of the temporal and spatial features, which are learnt. We encounter several practicable limitations as forest fires events are very rare and manual classification is extremely costly. Another is the unbalanced nature of the problem of the many forest fire events many are of the burnt area is very small and gives skewed distribution. Most of the examples naturally group into batches, which are collected from evidence satellite photography and collaborative reports from national parks departments. The second set of database was collected from the meteorological weather station about several weather observations, which are located very close to the reported fires. Finally, the compiling task is to serve as a filter and provide the user to vary the false alarm rate. We show by regression analysis of the compiled dataset that the forest fire classifier has a minimum false alarm rate when including temporal features. The machine learning algorithms successfully classifies accidental small fires with 85% reliably and large fires by a much lower accuracy of 30%.**

Index Terms—**Machine Learning, Datamining, Naive Bayes, Forest fires, Fire Weather Index (FWI), Temporal Patterns, WEKA machine learning framework.**

I. INTRODUCTION

Accidental small forest fire can lead to heavy loss of precious natural reserves in protected lands in which many different species thrive due to their balanced ecology. Tropical rain forests are also a factor in keeping the sensitive balance global warming trends seen recently due to heavy deforestation due to human needs. One of objective of Datamining is to allow modeling the users' perspective such as temporal properties, which are cause and effect of forest fires, which allows in reducing false detection. The hidden patterns are mined, which allows to find the underlying hidden structure of the data. This allows learning the concepts needed for forest fires classification. The features extracted of the predicted class by means of datamining allows to apply many machine learning algorithms to the transformed data. This framework forms the technical basis for the supervised and unsupervised classification.

Temporal query properties like weekday and weekends help probabilistically bias the predicted outcome of class variables to be classified. As a small human accidental fire or a possibility of occurrence of large natural fire disasters can further be classified according to the users choice. Attribute value transformations are equally important when formulating attribute dependencies within a weather class [4,5,6,7] nominal values such as cool, windy and high humidity for successful formulation of machine learning rules.

Following motivation the rest of the paper with is organized as follows. In Section III, we evaluate the performance of machine learning algorithms and develop a weak learner for temporal features. Section IV presents initial basis for user queries without significant error analysis, that is without any ranking criteria. In Section V, we evaluate Naive Bayes [1] with Tree based classifiers and compare the method on the task of accidental fire prediction. Section VI investigates alternative feature and computational aspects of the method respectively and explains the results. Section VII concludes the paper.

II. STATE OF THE ART

The historic information recorded by it does not reveal any hidden patterns to calculate the likelihood of forest fires. Classifiers model depends on accurate class conditional probabilities but in practice, samples are limited, most of the estimates are approximate which further biases the sample search space. With limited samples, the Bayesian method does the better estimation of the class conditional probabilities, when compared to maximum likelihood method. The internal representation of the classifier data model uses a weighted term, and best evaluates the quantity purity of the labeled class. This provides classification of the same-labeled pattern with insufficient samples, with pure and impure groups and helps its internal ranking. The internal representation captures the hidden pattern of the training samples, once the hidden patterns are quantitatively verified with a base classifier such as Nave Bayes, these representative patterns are further classified by user specified attributes such as which month and which day the particular pattern has maximized the likelihood of a phenomenon such as fire event.

There are standard benchmarks for performance comparison of classifiers and Bayes gives the lowest error rate compared to others. We also study the kappa score, which compares our classifier with a J48 tree classifier for the same input data set and normalizes using the results of the confusion matrixes. A high kappa score is generally preferred for a classifier to be efficient, which needs using of good pre-processing algorithms. Since sensor data are, highly unreliable most of well-designed classifiers perform badly and cannot adapt to the sensor data stream. By using post-processing of miss-classified samples and identifying falsely classified data also called outliers, we further improve the reliability. From authors previous work [1], we have shown data aggregation eliminate redundancies and improves reliability in sensor network performance. The current ML algorithm focuses on event aggregation over a long period of time from user reports and collaborative sensor network stream, which have been further classified to a particular application. We study the effects of predicting forest fires in a given region using sensor aggregated data.

III. MACHINE LEARNING RULES

Consider the concept leaning, in particular the learner considers some finite hypothesis [6] space H defined over instance space X, in which the task is to learn some target concepts $c : X- > 0,1$. As we are building a fire event predictor from the sensed data, we assume that the network learner is given some sequence of training measurements $((X_1, d_1)...(x_m, d_m))$ where x_i is some instance from X and where d_i is the target value of $d_i = c(x_i)$. As we are learning from a knowledge base such as data repository the sequence of instances $(x_1...x_m$ is held fixed, so that the training data D can be written as the sequence of target values $D = d_1...d_m$.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \qquad (1)$$

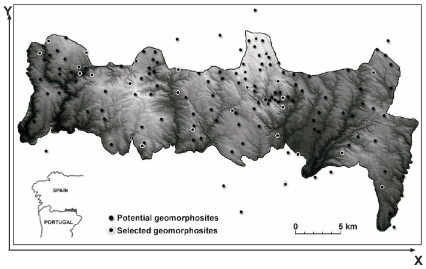


Fig. 1. location of inventorised geomorphosites in the montesinho natural park.

$$h_{MAP} = argmax_{h \epsilon H} P(h|D) \qquad (2)$$

The assumptions for the concept learning follows:

- The training data D is noise free.
- The target concept c is contained in the hypothesis space H.
- We have no a priori reason to believe that any hypothesis is more probable than any others.

Since we assume noise free samples, the first hypothesis can be formed for the detection of forest fires [2] for an approximate target function V as shown in equation (3)

$$V(Fire\,Location) = XPos + YPos \qquad (3)$$

where every incident of forest fire is documented and its location in terms of <X, Y> [2] are recorded. The learning algorithm uses a boosting [5,6] method to learn from the forest fire events and its corresponding correlated sensor measurements. The model does not use real-time sensor inputs and data samples to classify but on the other hand it uses recorded fire events and probabilistically predict the new sensor input closest to the already seen training sample (data aggregated over time) using Naive Bayes or Tree classifiers. This computational model can further post-processed using supervised learning to improve on the purity of the classes and detect any outliers which may create false alarms. The unreliability of accurately detection real and outlier events is an open problem in sensor networks. The above equation is dependent on <X, Y> positions in Figure 1 map, for grid of 10x10 it may need 100 combinations of every other dependent variable making the model unfriendly. The estimated formula of the above equation (3) can estimated in temporal terms for the Fire Event as shown in equation (4).

$$\begin{aligned} \mathrm{V_{train}}(\mathrm{FireEvent_{Day\,of\,week}}) \leftarrow & \qquad (4) \\ \hat{V}(FireEvent_{Day\,of\,week})\textit{Temporal Variable} & + \\ \hat{V}(FireEvent_{Day\,of\,week})\textit{Correlated measurements} & \end{aligned}$$

$$\begin{aligned} \text{Temporal Variables} = \textit{Month of the year} + & \qquad (5) \\ \textit{Day of the week} & \end{aligned}$$

$$\text{Correlated measurement} = \textit{temperature} + \textit{humidity} + \textit{wind} + \textit{rain} \tag{6}$$

$$\text{Classfires} = \{\text{accidental; small, medium, large}\} \tag{7}$$

A. Estimating training values with sample data

Sample datasets are based on UCI forest fire repository. The equation representing the Bayes probability model of the hypothesis is given in equation (1). In our case the hypothesis to be maximized is shows in equation (2) for a four class classification, as shown in equation (7). The assumption here is that the training set D is a unbiased representation to learn the concept c and can estimate the inputs x_i. The previously defined dependent variable **Fire Location**, which is used to estimate given the independent correlated measurements and its relation to the temporal attributed are given in equations (4), (5) and (6). The target concepts are present in the training samples and, we like to see the influence of adding sensor measurements to further accurately learn the concepts of the human induced accidental fires versus the more natural accruing types of the medium and large fires. For the sake of clarity of machine learning domain we convert the correlated sensor data to nominal [5] types, as illustrated below.

$$temperature = \{cool; mild; hot\} \tag{8}$$

$$humidity = \{low; medium; high\} \tag{9}$$

$$wind = \{true; false\} \tag{10}$$

The model estimation of the the target function with weights w_1, w_2 as shown allows to minimize the training error, where x_1, x_2 are temporal and correlated measurements.

$$\hat{V} = w_1 x_1 + w_2 x_2 \tag{11}$$

The learning algorithm needs to define the best fit for the given hypothesis and adjust the weights to minimizing the error and miss classifications.

$$E \equiv \sum (V_{train}(FireEvent) - \hat{V}(FireEvent))^2 \tag{12}$$

B. Algorithm complexity

Search space consists of all the possible patterns of the features, given our data model, $3 * 3 * 2 * 4 = 72$ possibilities for each rule when using attributes 3 for temperature, 3 for humidity, 2 for wind for 4 classes of fire categories. As there are 517 rules from the collected dataset instances the complete search space [5] will have ${}^{n}P_r = 72^{522} \approx 10^{969}$ different possibilities. To minimize the complexity of search space, we can further cut down on the sample instances by using spatial clustering and removing any redundancies in similar features. Given the `<X,Y>` positions, we can cluster into groups the possible fires types into accidental small fires and others which have medium and larger burnt area as large fires.

FIRE TYPES	RECORDED
Accidental (AF)	247
Small (SF)	175
Medium (MF)	71
Large (LF)	24

TABLE II
TARGET VARIABLE OCCURRENCES.

As measuring ambient phenomena are correlated, we expect them to be independent, then all the i.i.d's can be aggregated to ${}^{n}C_r = 517 C_{72} \approx 10^{89}$ possibilities, were in this case the combination is calculated. As these methods are used with pre-processing to reduce data overloads in the model further real valued dataset search space optimization is possible. Domain knowledge as in the case of WSN can be used practically to reduce the complexity of machine learning algorithms if well studies and calibrated. To judge the affectiveness of the model and the classification effectiveness, we initially rely on real-valued numeric model such as [1] to estimate the errors. In contrast to the previous approach, we use nominal values as defined in equations (8), (9) and (10) to build a tree classifier and further reduce errors.

IV. NAIVE BAYES

One can use Naive Bayes [5], which by design presumes the class densities, which have been determined and accurate. The model calculates the class conditional probabilities of the input feature vectors. To understand the underlying skewed structure of the dataset, we further create thresholds for accidental small fires compared to medium and large fires as shown in Table II. So we have the four possible values for the target variable as shown in equation (7).

A. User query

To validate the model let us predict the outcome of a peak month, from the dataset [2] August has significant number of reported fires compared to other months. Estimating the probabilities of fire events given the attribute values for the class

$$? = \{Month = August; Day = Monday\}$$
$$\{Temprature = Cool; Humidity = High; Wind = True\}$$

The estimated class conditional densities for the independent variables temperature, humidity and wind conditions are calculated using temporal attributes **month** for the dataset are shown in Table II. The datasets further is explored using two temporal variables, which are **month** and the **day of the week** as shown in Table I and Table IV. The temporal variables introduced into the dataset helps gain the insight of users' dependencies with fire prediction model.

$$g_i(x) = P(\omega_i \| \mathbf{x}) = \frac{p(\mathbf{x} \| \omega_i) P(\omega_i)}{\sum_{i=0}^{i=4} p(x \| \omega_j) P(\omega_j)} \tag{13}$$

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

Burnt Area(hectors)	AUG	MON	TEMP	HUMIDITY	WINDY	PRIOR PROB	PREDICTOR VAR
GT 1h	▶ 0.34	▶ 0.14	▶ 0.46	▶ 0.17	▶ 0.42	▶ 0.47	▶ 57%
GT 1h LEQ 10h	0.39	0.15	0.35	0.13	0.38	0.33	25.0%
GT 10h LEQ 50h	0.30	0.14	0.43	0.19	0.50	0.13	17.0%
GT 50h	0.33	0.08	0.16	0.08	0.37	0.04	0.02%

TABLE I
POSTERIORS PROBABILITIES FOR BACKGROUND WEATHER DATA FOR THE PEAK MONTH AUGUST.

FIRE TYPE	MONTH=AUG
Accidental	0.004
Small	0.002
Medium	0.001
Large	0.00004

TABLE III
LIKELIHOOD OF FIRES FOR THE MONTH OF AUGUST.

DAYS	ACCIDENTAL	SMALL	MEDIUM	LARGE
MON	▶ 35	▶ 27	▶ 10	▶ 2
TUE	28	21	11	4
WED	22	24	5	3
THU	30	21	9	1
FRI	42	31	12	0
SAT	42	24	11	7
SUN	48	27	13	7
TOTAL	▶ 247	▶ 175	▶ 71	▶ 24

TABLE IV
POSTERIORS PROBABILITIES FOR TEMPORAL FEATURE DAY OF THE WEEK.

Substituting the corresponding highlighted values from Table 1 through to Table IV in the above equation (13), we get the posterior probability of accidental small fire

$$fir\hat{e}_{accidental} = \frac{0.0007547}{0.003565} = 57\% \tag{14}$$

$$fir\hat{e}_{Small} = \frac{0.000333}{0.003565} = 25\% \tag{15}$$

$$fir\hat{e}_{medium} = \frac{0.000223}{0.003565} = 0.17\% \tag{16}$$

$$fir\hat{e}_{large} = \frac{0.000000287}{0.003565} = 0.02\% \tag{17}$$

The posterior probabilities for the month of August for the data collected in Portugal [2], the likelihood of accidental small fires are very high. From cross-validating from the known fact that in summer the likelihood of wild fires are higher the Bayes rule is able to classify the dataset for accidental and small fires with high accuracy. We use a simulation framework in the next sections to further prove our intial conclusion from the datasets, it is shown that the training time for Naive Bayes scales linearly in both the number of instances and number of attributes.

V. TREE CLASSIFIER

In this section, we will focus on the domain rules, which are applicable to the learning system. Tree classifiers lend itself to use ML rules [6] when searching the hypothesis by further branching on specific attributes. The design of such a classifier needs to sort the weights or entropies [5] of the attributes, which is the basis of its classification effectiveness.

ID3 is a popular tree classifier algorithm, to implement ID3 as illustrated in Figure 2 and Table X with our attributes. Let (S) be a collection of samples then using the tree algorithm, which uses entropy to split its levels is given by

$$Entropy(S) = \sum_{i=0}^{i=c} p(i) \log_2 p(i) \tag{18}$$

Let us assume a collection (S) has 517 samples [2] with 248, 246, 11 and 12 of $accidental, small, medium, large$ fires respectively. The total entropy calculated from equation (18) is given by

$$\begin{aligned} Entropy(S) = {} & \frac{248}{517}\log_2\frac{248}{517} + \\ & \frac{246}{517}\log_2\frac{246}{517} + \\ & \frac{11}{517}\log_2\frac{11}{517} + \\ & \frac{12}{517}\log_2\frac{12}{517} = \\ & 1.23 \end{aligned}$$

A. Attribute selection

ID3 uses a statistical property called information gain to select the best attribute. The gain measures how well the attribute separates training targeted examples, when classifying them into fire events. The measure of purity that we will use is called information and is measured in units called bits. It represents the expected amount of information that would be needed to specify whether a new instance should be classified accidental, small, medium or large fires, given that the example reached that node. The gain of an attribute is defined by and illustrated in Table V. Using the calculated attribute for information gain we show that **temp** attribute is used before the **wind** attribute to split the tree after the tree root.

$$Gain(S, A) = Entropy(S) - \sum_{i=0}^{i=c} \frac{S_v}{|S|} Entropy(S_v) \tag{19}$$

$$\begin{aligned} Entropy(S_{Hot}) = {} & \frac{9}{36}\log_2\frac{9}{36} + \\ & \frac{23}{36}\log_2\frac{23}{36} + \\ & \frac{3}{36}\log_2\frac{3}{36} + \\ & \frac{1}{36}\log_2\frac{1}{36} = \end{aligned}$$

$$1.282$$

$$Entropy(S_{Medium}) = \frac{23}{96}\log_2\frac{23}{96} + \frac{65}{96}\log_2\frac{65}{96} + \frac{3}{96}\log_2\frac{3}{96} + \frac{5}{96}\log_2\frac{5}{96} = 1.175$$

$$Entropy(S_{Cool}) = \frac{117}{269}\log_2\frac{117}{269} + \frac{146}{269}\log_2\frac{146}{269} + \frac{2}{269}\log_2\frac{2}{269} + \frac{4}{269}\log_2\frac{4}{269} = 1.05$$

$$Entropy(temp) = \frac{43}{517} * 1.282 + \frac{139}{517} * 1.175 + \frac{335}{517} * 1.05 = 1.08$$

$$Gain(S, temp) = 1.23 - 1.08 = 0.192$$

$$Entropy(S_{HIGH}) = \frac{162}{249}\log_2\frac{162}{249} + \frac{72}{249}\log_2\frac{72}{249} + \frac{8}{249}\log_2\frac{8}{249} + \frac{7}{249}\log_2\frac{7}{249} = 1.1952$$

$$Entropy(S_{LOW}) = \frac{68}{133}\log_2\frac{68}{133} + \frac{59}{133}\log_2\frac{59}{133} + \frac{2}{133}\log_2\frac{2}{133} + \frac{4}{133}\log_2\frac{4}{133} = 1.24$$

$$Entropy(wind) = \frac{361}{517} * 1.1952 +$$

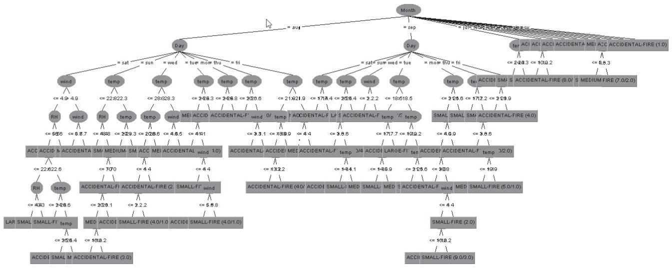


Fig. 2. Tree classifier and attribute view.

Month	Temp	Wind
Not shown	info: 1.08	info: 1.20
Not shown	gain: 1.23-1.08 = 0.192	gain: 1.23-1.08 = 0.025

TABLE V
GAIN RATIO CALCULATION FOR TREE IN FIGURE 2.

$$\frac{156}{517} * 1.24 = 1.20$$

$$Gain(S, wind) = 1.23 - 1.20 = 0.025$$

The internal tree representation for m attributes from n samples will have a complexity of $O(\lg n)$, with increasing inputs, given by parameter n, the height of the tree will not grow linearly as in the case of Naive Bayes. On the other hand complexity of building a tree will be $O(mn \lg n)$

VI. SIMULATION

Open-source workbench called WEKA [3] is a useful tool to quantify and validate results, which can be duplicated. WEKA can handle numeric attributes well, so we use the same values for the weather data from the UCI [4] repository datasets. The class variable has to be a nominal one, to allow WEKA [3], we convert all fire types to ″0″ or ″1″. Where ″0″ is of accidental small fire and ″1″ is for large fires making it a two class classifier, the results are shown as confusion matrix in Table VIII and Table IX. Naive Bayes correctly classifies accidental and small fires(209 out of 247) were as the J48 Tree classifier does far more, 219 out of 247. As WEKA uses kappa [3] stats for evaluating the training sets, a standard score of $> 60\%$ means training set is correlated, using J48 simulation, we get 53.56% just below the standard. The comparison on results shows that tree classifier does better than Naive Bayes by 25% overall and equally well for accidental and small fires as shown in Table VI and Table VII, when randomly tested it falls just short of the expected 60%. Therefore using sensor network measurements accidental and small fires can be predicted reliably.

WEKA Stats	Results	Summary
Correctly Classified Instances	267	51.64%
Incorrectly Classified Instances	250	48.35%
Kappa statistic	0.1371	
Mean absolute error	0.3022	
Root mean squared error	0.3902	
Relative absolute error	94.86%	
Root relative squared error	97.84%	
Total Number of Instances	517	

TABLE VI
EVALUATION ON TRAINING SET FOR NAIVE BAYES.

WEKA Stats	Results	Summary
Correctly Classified Instances	373	72.14%
Incorrectly Classified Instances	144	27.85%
Kappa statistic	0.5356	
Mean absolute error	0.1938	
Root mean squared error	0.3113	
Relative absolute error	60.83%	
Root relative squared error	78.04%	
Total Number of Instances	517	

TABLE VII
EVALUATION ON TRAINING SET FOR J48 TREE CLASSIFIER.

A. Simulation analysis

WEKA attribute statistics and its effective correlation score. Table VI and Table VII show kappa and other comparison statistics for Naive Bayes and J48 tree classifier.

B. Error analysis

Equation (12) specifies the model error and the Confusion matrix from the simulation score are shown in Table VIII and Table IX, upper bound of small fire(AF+SF) has over 80% accuracy for J48-Tree and 61% for Naive Bayes. The corresponding baseline performances including all fires categories is 72.1% for J48-Tree and Naive Bayes is 51.64%, which is due to large fires not correlated.

1) Correlation of attributes: From statistical point of view if the attributes have similar values then it creates high bias creating what is called over-fitting error during learning. In our case **temp** and **humidly** may have similar values and needs to be avoided and substituted with a suitable attribute. To pre-process and analyze, we use all the available in the dataset and WEKA provides the attribute selection as illustrated in Table X.

We use the attribute selection wizard of WEKA to find out the best match. The analysis shows from Table X that the Month(100%), Day(10%) and Wind(0%) are highly dependent on the precision. As most of the attributes are nominal it lends more to a tree classifier, which are more flexibility in handling nominal types by design.

	LF	MF	SF	AF
LF	0	1	7	16
MF	0	5	12	54
SF	0	7	53	115
AF	0	0	38	209

TABLE VIII
CONFUSION MATRIX FOR NAIVE BAYES USING TRAINING SET.

	LF	MF	SF	AF
LF	7	0	7	10
MF	0	29	15	27
SF	1	7	118	49
AF	0	5	23	219

TABLE IX
CONFUSION MATRIX ON TRAINING SET FOR J48 TREE CLASSIFIER.

Number of folds (%)	No.	Attribute
10(100 %)	1	month
1(10 %)	2	day
0(0 %)	3	temp
0(0 %)	4	RH
0(0 %)	5	wind

TABLE X
ATTRIBUTE SELECTION 10 FOLD CROSS-VALIDATION (STRATIFIED)

VII. CONCLUSION AND FUTURE WORK

The future research work will focus on how to rank sensor queries with high reliability which otherwise be biased due to unverifiable outliers present in the form of noise, spikes and false positives in the time-series data. The training sample sorting allows to weigh the precession versus relevant evidence based on the ranking criteria, such has F-scores and correlated Fire Weather Index (FWI) to further compare the likelihood of predicting large fire events reliably. The statistical analysis of the data collection helps in exploring the higher and lower bounds of the FWI ranges and its corresponding robustness to predict large fires using our implemented algorithms.

VIII. ACKNOWLEDGEMENT

One of the authors like to thank Shailesh Kumar of Google, Labs, India for suggesting the machine learning framework WEKA and UCI for providing the most needed datasets on forest fires. Dean P.J. Narayanan has been a mentor in introducing dimension reduction and attribute selection. The first author like to express appreciation and support from Dr. S.S Iyengar for funding this work under his LSU research grant. Authors also like to thank the anonymous reviewer's comments which has improved the final quality of the submission.

REFERENCES

[1] Vasanth Iyer, S.S. Iyengar, G. Rama Murthy, and M.B. Srinivas. INSPIRE-DB: Intelligent Networks Sensor Processing of Information using Resilient Encoded-Hash DataBase. SENSORCOMM 2010, The Fourth International Conference on Sensor Technologies and Applications, Venice, pp. 363-368.

[2] Paulo Cortez and Anibal Morais. A Data Mining Approach to Predict Forest Fires using Meteorological Data. Department of Information Systems-R&D Algoritmi Centre, University of Minho, Portugal.

[3] WEKA Machine learning software. $http : //www.cs.waikato.ac.nz/\ ml/weka$ [Accessed May 15th, 2011].

[4] Frank, A. and Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. [Accessed May 20th, 2011].

[5] Ian H. Witten and Eibe Frank. Datamining, Pratical machine learning. Elsevier 2005.

[6] Tom M. Mitchell. Machine Learning. MaGRAW-Hill Publications 1997.

A Simulation Framework for the Performance Evaluation of Localisation Techniques over WSNs

Aggeliki Prayati
Dept. of Telecommunication
Systems and Networks,
Technological Educational Institute
of Messolonghi
Nafpaktos, Greece
prayati@ece.upatras.gr

Nikos Paliatsas
Dept. of Telecommunication
Systems and Networks,
Technological Educational Institute
of Messolonghi
Nafpaktos, Greece
tastoises@hotmail.com

Vassilis Triantafillou
Dept. of Telecommunication
Systems and Networks,
Technological Educational Institute
of Messolonghi
Nafpaktos, Greece
triantaf@teimes.gr

***Abstract*— Wireless sensor networks (WSNs) are nowadays considered to be a very active research field. WSNs consist of interconnected small devices, which are managed by users. Their demands in hardware are limited and the usage of particular communication protocols lead in great autonomy. A WSN application is the localization of targets invading in the area covered by sensors' range. This target may be an animal, human, object or anything else. Localization can only take place when the sensors are capable of processing data and of communicating with each other. When these conditions hold, localization is achieved. A simulation environment has been developed in Matlab for studying the above described problem, where the localization algorithm may be integrated together with different scenario parameters. These scenarios have been implemented for evaluating the localization algorithm performance for different mobile object trajectories and speeds as well as for different network topologies. Finally, the evaluation results are discussed and conclusions drawn are presented.**

Keywords-wireless sensor networks, localization, simulation

I. Introduction

As wireless sensor network (WSN) applications cover a growing number of fields, the need for fast yet accurate exploration of optimal alternatives becomes more demanding, thus necessitating the development of structured and efficient performance evaluation strategies.

An efficient localization algorithm is expected to predict the path of the mobile object with high accuracy. Localization applications using WSNs range from health monitoring to house safety and even to military surveillance [1]. On the other hand, the localisation performance is affected by a series of parameters that need to be addressed and co-evaluated during the simulation process [2]. These parameters are related to the communication channel, to the target mobility pattern and to the localization nodes' positions. The type of signal used for the target position varies from RF to IR and ultrasound depending on their suitability for different application scenarios and on the available power [3]. The mobile target speed and mobility pattern are important parameters as they directly affect the time that the target resides in communication range and indirectly the tracking accuracy. Moreover, the network topology may facilitate or harden the localization task, i.e. a randomly deployed set of nodes has less probability to achieve high communication coverage in the localization area than a pre-defined deployment scheme. Finally, the limited resources of WSNs impose restrictions to their capabilities and performance; the limited memory restricts the use of complicated and demanding algorithms and the limited battery storage restricts the power consumption and extensive execution time [2]. After sensing and processing data, WSN nodes need to consume it or send it immediately to a storage point [4].

Several approaches have been proposed in the literature for the evaluation of localization techniques in WSN deployments. A methodological approach to the evaluation of localization algorithms is thoroughly presented in [5]. Other approaches focus on comparing the localization error with respect to energy [6] and others are limited to customize the simulation environment to specific deployment and application characteristics [7]. However, all of them base their evaluation procedure on theoretical models for the signal propagation, do not consider the particularities of the network topology scheme or the environmental conditions that affect network connectivity and the communication pattern.

In this work, the issue of localisation applicability and performance with WSNs is discussed. For this purpose, a simulation strategy and environment has been developed, incorporating models of all aspects of WSN performance affecting the localisation accuracy. Section 2 presents the most commonly used localisation techniques, whilst Section 3 covers WSN related parameters that need to be considered during a performance evaluation study. Section 4 presents the simulation process and scenarios followed by this work, as well as the simulation environment developed to perform the experiments and Section 5 discusses the experimental results. Finally, conclusions are drawn in Section 6.

II. LOCALISATION TECHNIQUES

As many localization techniques exist in the literature, this work focuses on the most widely used for fixed network topologies and analyses them with respect to the network and mobility parameters to select the most representative one for studying localization performance of a mobile target in fixed WSN topologies.

At the initial phase of the localization algorithm, every fixed node makes known its position to the rest of the nodes in the network. At the end of this phase, nodes keep only their neighbor's positions so as not to overload their memory. Once a mobile target enters the localization area, the fixed nodes attempt to track its path based on their relative distances from the target. The result of the localization algorithm is then either communicated to a node centrally placed in the network area or consumed by the mobile target or even by a set of fixed nodes in the network. This last step depends on the application case.

The complexity of impact factors and varying application cases affecting the localization task success have led to a wide variety of existing localization techniques targeted to specific application requirements. These techniques differ in the type of signal used to estimate the target position, with the most popular being the RF signal, ultrasound, infrared and the received signal strength (RSS). Some techniques have been developed for fixed topologies and others for dynamically changing networks. Focusing on the fixed topology localization techniques, grid and random scenarios are discriminated as the varying impact factors are seriously affecting localization performance.

For grid topologies, the most common technique is the fingerprinting approach, which computes the target position based on the comparison between the RSSI and predefined signal strength measurements stored in a database [8]. For random topologies, the most famous technique is hop-counting, which uses RF signals and hop count tables for every node's neighbors [9].

The work presented in this paper adopts the triangulation technique based on RSSI, which belongs to the hop-counting localization family. This method bespeaks at least three fixed nodes in the vicinity of the mobile target, for the localization to be performed. The mobile target can be tracked inside the triangular area covered by the three fixed nodes in communication range to the mobile, as shown in Fig. 1. The distance among each nearest fixed node and the mobile target is calculated based on the RSSI from the mobile target. This procedure is repeated for all fixed nodes nearest to the mobile target along its path until the mobile target exits the communication area of the WSN.

III. WSN MODELING PARAMETERS

The WSN parameters have been modeled to facilitate the evaluation of the triangulation technique for localization through simulation. These parameters are:

a) the signal propagation model used to translate RSS to distance and vice versa,
b) the fixed nodes topology,
c) the mobile target speed and
d) the mobile target mobility pattern.

A. *Signal propagation model*

Distance computation is achieved based on the RF propagation model developed in [2], which takes into account environmental parameters and the fixed topology impact on the RSSI. As such, the propagation model adopted by this work is based on experimental measurements and is accurate enough and representative of a real-life scenario, as opposed to the theoretical RSS-distance characteristic provided by the TelosB manufacturer [12].

As discussed and concluded in [11], for a RSSI-based triangulation technique to be applied in a fixed topology with satisfactory results, the mobile target is considered to be at 0.5m and the fixed nodes at 0.55m. The RSSI vs. distance characteristic, adopted by [11], shown in Fig. 2, represents a realistic RSSI characteristic, considering the environmental impact on communication and connectivity conditions. The blue characteristic has been quantified to form the RF propagation model during the simulation experiments conducted in this paper.

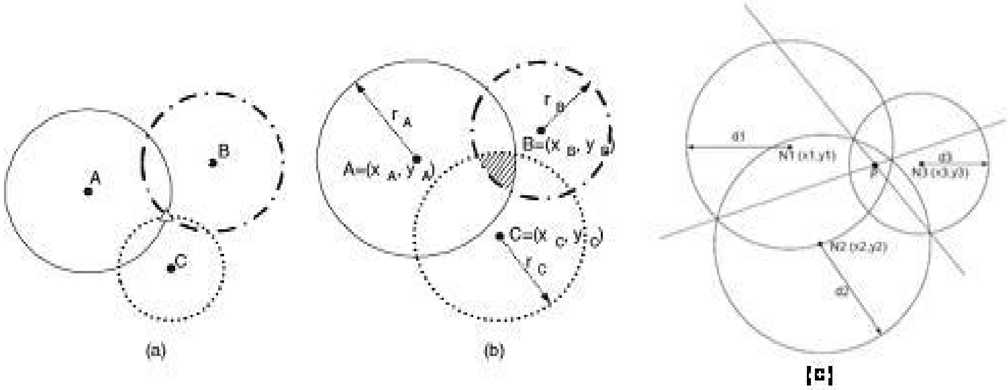


Figure 1. Location calculation based on triangulation

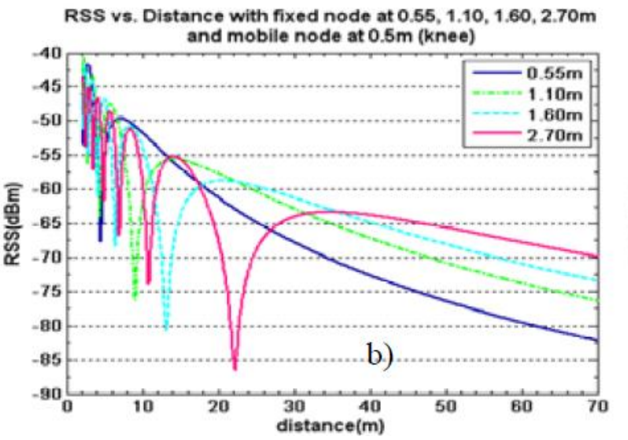


Figure 2. RSSI-distance diagram with mobile object at 0.5m and fixed nodes at many heights

B. Network Topology

The network topology has a high impact on the localization accuracy, as the fixed nodes positioning defines the communication links quality with the mobile target and thus the localization success. If the fixed nodes are deployed in a grid, the probability that the mobile target is reachable by at least three fixed nodes is high and thus it can easily located. On the other hand if the network topology is random, the communication quality is less probable to be good in the whole localization area, leading to low mobile target connectivity to the fixed nodes and thus degrading localization performance. Following the analysis of [10], the network topology of the fixed nodes is considered to be a grid with distance of 55m among them in order to achieve high communication quality and thus good connectivity.

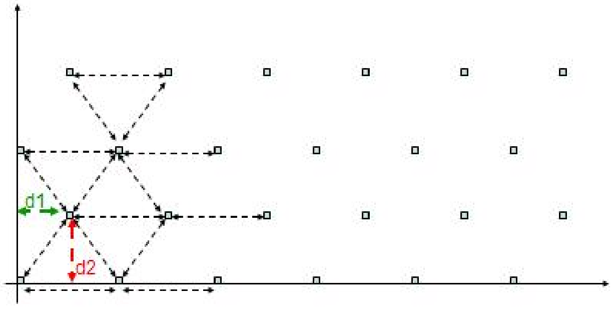


Figure 3. the Grid topology scheme used in the experimental analysis

C. Mobile object parameters

The two parameters affecting localization accuracy are the speed and the trajectory of the mobile target. If the speed of the mobile target is low, the three nearest nodes have enough time to apply localization calculations and to predict its position. For low mobile target speed, the three fixed neighboring nodes may need more rounds of predictions for increasing the localization accuracy. On the other hand, if the mobile object moves too fast, the time window during which the mobile target moves in the fixed nodes' communication area may not be long enough to allow message exchange and position calculation.

As far as the second parameter is concerned, namely the mobile target trajectory, its impact is lower relatively to speed. However, in combination with the fixed nodes topology, it can make communication very difficult and thus the localization task too hard. In this work, the impact of both the mobile target's speed and trajectory on localization accuracy are studied.

IV. SIMULATION PROCESS AND SCENARIOS

The evaluation methodology followed in this work assumes a fixed network topology and a mobile target moving in the network range. The localization technique adopted in this work is triangulation and the parameters affecting its performance are modeled in a simplified yet efficient way in order to provide a complete set of impact factors for inclusion in the developed simulation framework.

As shown in Fig. 4, the simulation-based evaluation procedure starts when the mobile object starts moving from a known starting point in the WSN covered area. While the mobile object enters the localization area, the fixed nodes start communication with it in order to compute the mobile object's RSSI. The three fixed nodes with the highest mobile object's RSSI are considered to be the nearest to it and start the localization process. The three fixed nodes translate the RSSI into distance based on the radio propagation model developed in [2], which depicts the real signal propagation of the TelosB platform [12]. Based on the range of each fixed node the algorithm calculates the points at which, the three communication ranges intersect. Thus, the triangle that the localization algorithm needs has been formed and its middle point is computed. For simplicity reasons, the mobile target position is assumed to be the center of the triangle formed by the communication range of three nearest fixed nodes. This procedure is repeated by the fixed nodes corresponding to the nearest range of the mobile object trajectory until the mobile object leaves the network area.

The impact factors on the localization algorithm performance form the set of simulation parameters and their combination the relative simulation scenarios:

- fixed network topology: grid and random
- the speed of the mobile object: 5Km/h (man walking) and 15Km/h (man running)
- the trajectory followed by the mobile object: beeline and random

The performance metric of the localization algorithm is the localization error.

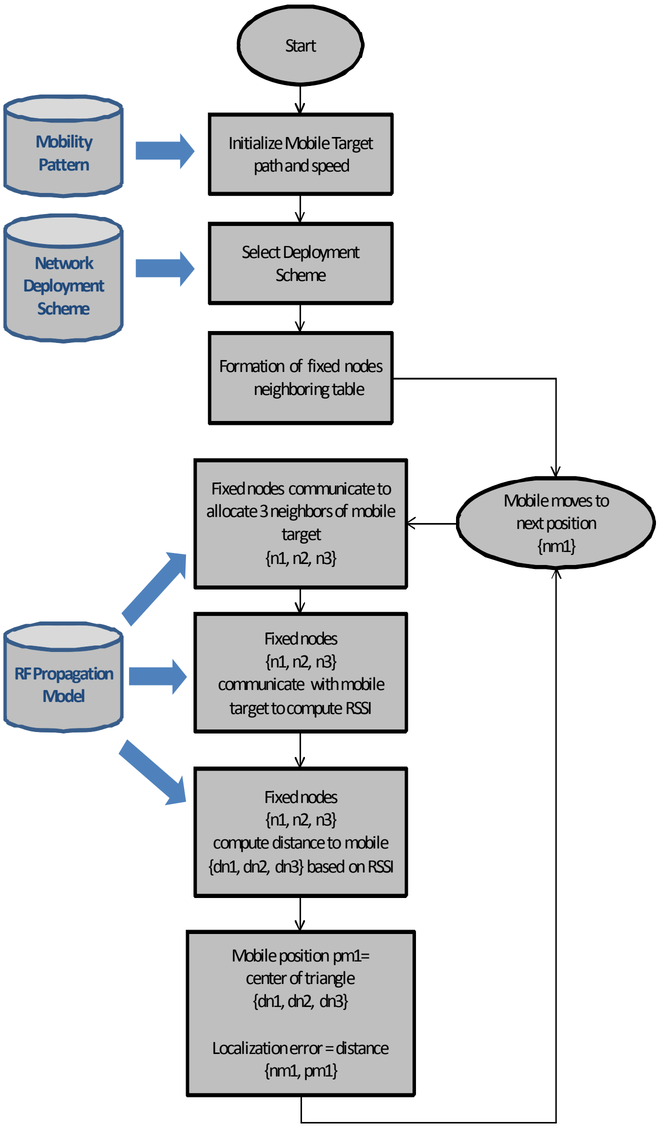

Start
Mobility Pattern
Initialize Mobile Target path and speed
Network Deployment Scheme
Select Deployment Scheme
Formation of fixed nodes neighboring table
Fixed nodes communicate to allocate 3 neighbors of mobile target {n1, n2, n3}
Mobile moves to next position {nm1}
RF Propagation Model
Fixed nodes {n1, n2, n3} communicate with mobile target to compute RSSI
Fixed nodes {n1, n2, n3} compute distance to mobile {dn1, dn2, dn3} based on RSSI
Mobile position pm1= center of triangle {dn1, dn2, dn3}
Localization error = distance {nm1, pm1}

Figure 4. Evaluation Process Diagram

V. LOCALISATION SIMULATION RESULTS

The simulation environment developed has been implemented in MATLAB 2009. The results of the simulation scenarios are presented and discussed in the following. Each figure depicts a ground plan (x-y axes grid in meters) of the network, with fixed nodes represented by blue squares, the blue line representing the actual trajectory of the mobile target, the small red crosses representing the positions predicted by the triangulation algorithm and the light green area representing the communication range of the mobile target to the nearest fixed nodes.

Fig. 5 and Fig. 6 show localization results of a mobile target in a grid fixed topology moving slowly (speed = 5Km/h), in a beeline and a random path respectively. The average localization error is 3.7m and 5.6m respectively, with low variance. As long as the mobile targets moves at low speed within the communication area of fixed nodes, the effect of its trajectory on the localization success is relatively low.

On the other hand, when the speed of the mobile target is tripled for the same network conditions and fixed nodes processing time, the percentage of successfully received packets is reduced by 30%, thus leading to position miscalculations. As shown in Fig. 7 and Fig. 8, the number of position calculations is considerably reduced to 53% for the straight path line and to 71.5% for the random path. Another conclusion drawn from the 4 latter figures is that the localization algorithm is easier to predict the straight path than the random path with the localization error to be 4.5m and 5.8m respectively. The red spots that the algorithm predicts in straight move is nearest than them in random move. Therefore the prediction in straight move is better than the random move.

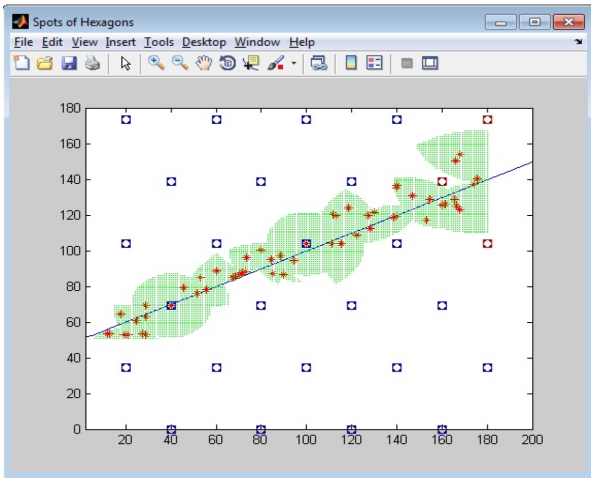


Figure 5. Straight move in grid topology with speed 5km/h

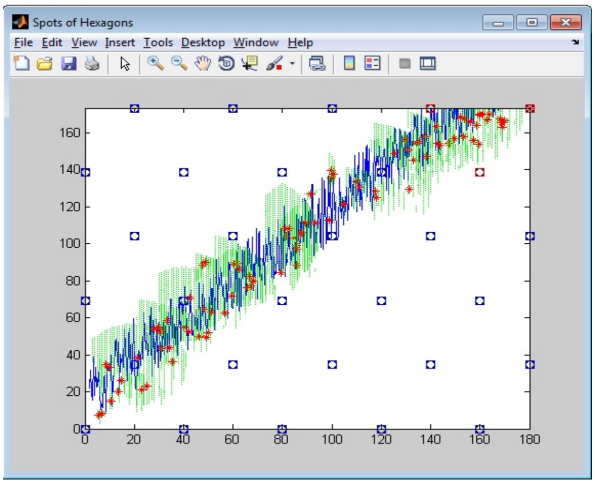


Figure 6. Random move in grid topology with speed 5km/h

ISBN: 978-1-61208-144-1

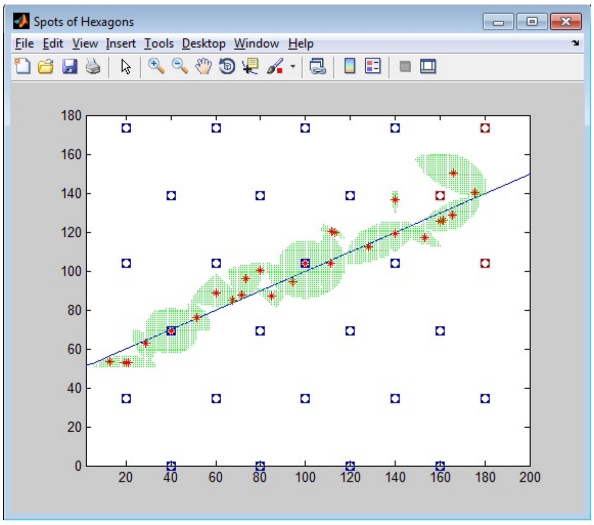


Figure 7. Straight move in grid topology with speed 15km/h

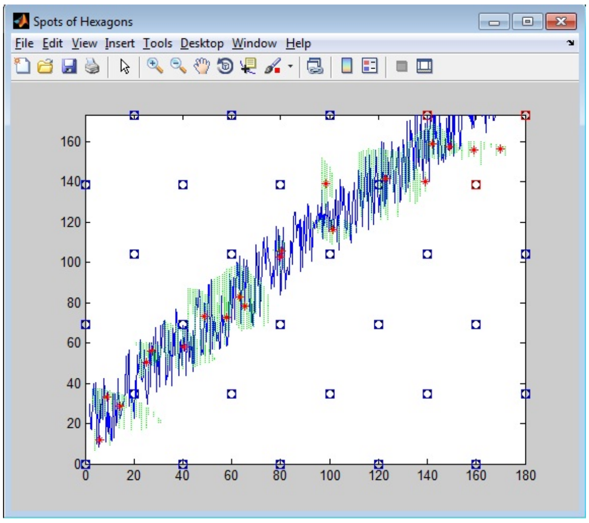


Figure 8. Random move in grid topology with speed 15km/h

Fig. 9 and Fig. 10 depict the mobile target localization in a random fixed grid, moving at slow speed of 5Km/h. It is evident that the random placement of fixed nodes narrows the communication range along the mobile target trajectory, thus leading to localization failure in spots, where 3 fixed nodes cannot be found within range of the mobile target. That is the reason why, in Fig. 9, the mobile target trajectory from points (140,110) until (200, 150) is not tracked at all.

This is not the case for the random mobile path shown in Fig. 10, where the mobile target has more spots within the communication range of fixed nodes, causing the prediction to rise up to 20.5% more than the straight move. However, it must be noted that for the random deployment scheme, even though the prediction level increases, the localization error is higher. It is, therefore, evident that the fixed nodes should be deployed in such a way as to cover as evenly as possible the localization area from a communication point of view.

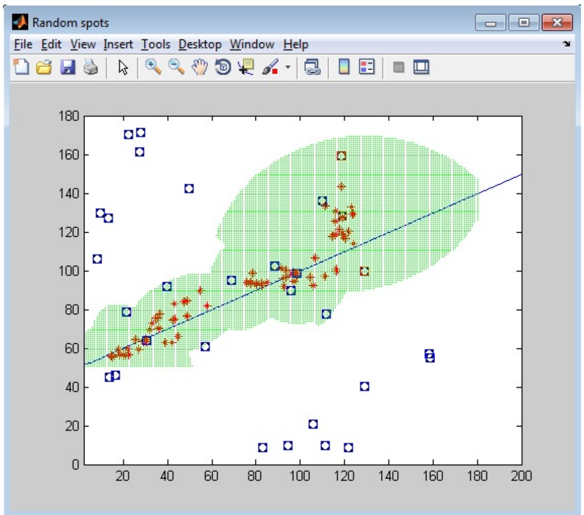


Figure 9. Straight move in random topology with speed 5km/h

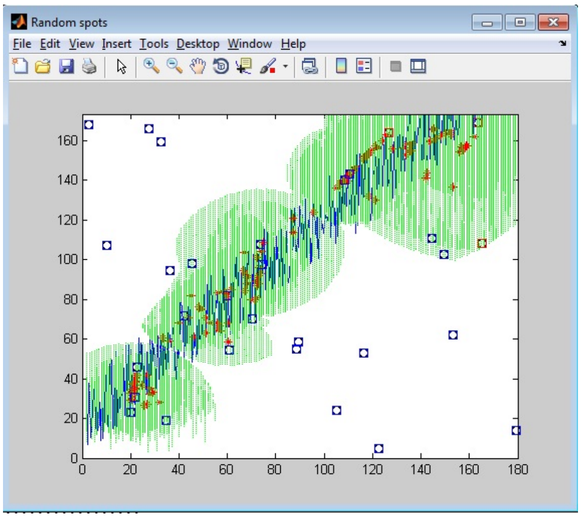


Figure 10. Random move in random topology with speed 5km/h

Fig. 11 and Fig. 12 show localization results of the worst-case scenario, where the network topology is random, the mobile target moves at high speed of 15Km/h and its path goes out of the fixed nodes’ communication range for a high percentage. The triangulation algorithm performance is seriously degraded with a low percentage of localization success close to 10.6m and 38.3m for beeline and random trajectory respectively.

Overall, for the triangulation technique to achieve high accuracy, the network topology should be carefully studied to be at least similar to a grid. The mobile target trajectory plays a minor role for the localization error, as long as the target stays within the fixed nodes communication range. Finally, the mobile target speed has high impact on localization performance but as this is a parameter that cannot always be managed, the density and deployment scheme of the fixed nodes is again a parameter that should be used to compensate to achieve the localization goal.

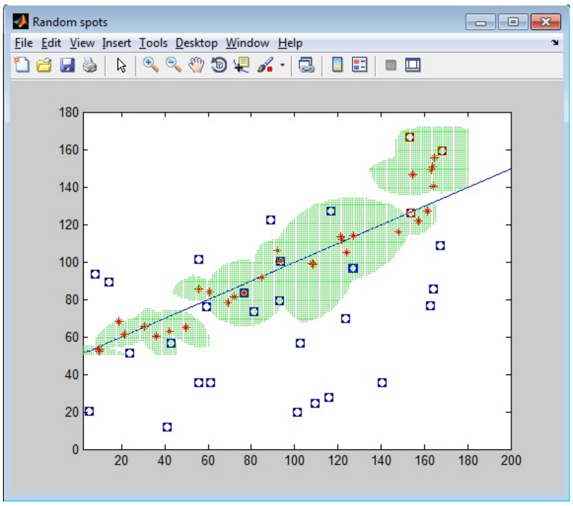


Figure 11. Straight move in random topology with speed 15km/h

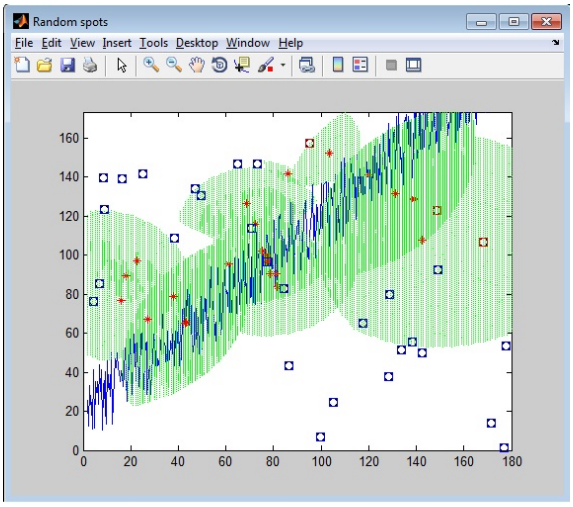


Figure 12. Random move in random topology with speed 15km/h

VI. CONCLUSIONS

This papers deals with the issue of localisation applicability and performance in WSNs. For the evaluation of localization techniques, several approaches have been studied and the triangulation technique has been chosen for the case study. WSN characteristics have been modelled and a simulation strategy has been developed. The simulation environment, developed in Matlab, incorporates these models of all aspects of WSN performance affecting localisation accuracy. The presented methodology principles may be applied to other localization evaluation attempts and be cross-validated by experimental setups.

The simulation environment that has been developed follows a holistic approach covering all aspects of WSN nature and as such may be generalised to incorporate a library of localization techniques. Moreover, further enhancements of the type of evaluation scenarios may support the applicability study of existing or new localization techniques to different applications i.e. the variation of network topology schemes or the number of mobile targets. Overall, the simulation framework has been developed in a modular way as to cover the mobility parameters in combination to the network topology and radio propagation characteristics and is extensible as to be able to incorporate more evaluation and impact parameters in the form of libraries.

REFERENCES

[1] Marin, E. Arceredillo, A. Zuloaga, and J. Arias, “Wireless Sensor Networks: A Survey on Ultra-Low Power-Aware Design”, Proceedings of World Academy of Science, Engineering and Technology, Vol. 8, pp. 44-49, October 2005

[2] T. Stoyanova, F. Kerasiotis, A. Prayati, and G. Papadopoulos, “Evaluation of Impact Factors on Accuracy for Localization and Tracking Applications”, Proceedings of the 5th ACM international workshop on Mobility management and wireless access, pp. 9-16, 2007

[3] L. Barboni, and M. Valle, “Experimental Analysis of Wireless Sensor Nodes Current Consumption”, The 2nd International Conference on Sensor Technologies and Applications, pp. 401-406, Aug. 2008

[4] Paolo Baronti, Prashant Pillai, Vince Chook, Stefano Chessa, Alberto Gotta, and Y. Fun Hu, “Wireless Sensor Networks: a Survey on the State of the Art and the 802.15.4 and ZigBee Standards”, Computer Communications, Vol. 30, Nr. 7, pp. 1655-1695, May 2007

[5] Michael Allen, Sebnem Baydere, Elena Gaura, and Gurhan Kucuk. “Evaluation of Localization Algorithms”, Chapter in “Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking”, Information Science Reference, 2009, ISBN: 1605663964

[6] M. Keshtgary, M. Fasihy, and Z. Ronaghi, “Performance Evaluation of Hop-Based Range-Free Localization Methods in Wireless Sensor Networks”, ISRN Communications and Networking, vol. 2011, Article ID 485486, 6 pages, 2011, doi:10.5402/2011/485486

[7] Ahmad Hatami, Bardia Alavi, Kaveh Pahlavan, and Muzaffer Kanaan, “A Comparative Performance Evaluation of Indoor Geolocation Technologies”, Interdisciplinary Information Sciences, Vol. 12, No. 2, pp. 133–146, 2006

[8] Ana-Belén, García-Hernando, José-Fernán Martínez-Ortega, Juan-Manuel López-Navarro, Aggeliki Prayati, and Luis Redondo-López, “Problem Solving for Wireless Sensor Networks”, Springer Verlag London Limited., November 2008, ISBN: 978-1-84800-202-9

[9] Eddie B. S. Tan, J. G. Lim, Winston K. G. Seah, and S. V. Rao, “On the Practical Issues in Hop Count Localization of Sensors in a Multihop Network”, Proceedings of the 63 rd IEEE Vehicular Technology Conference, pp.358-362, May 2006

[10] T. Stoyanova, F. Kerasiotis, A. Prayati, and G. Papadopoulos, "Communication-Aware Deployment for Wireless Sensor Networks", 2nd IEEE International Conference on Sensor Technologies and Applications, pp. 217-222, August 2008

[11] F. Kerasiotis, T. Stoyanova, A. Prayati, and G. Papadopoulos, “A Topology-oriented Solution Providing Accuracy for Outdoors RSS-based tracking in WSNs” , Proceedings of the 2008 Second International Conference on Sensor Technologies and Applications, pp. 239-245, 2008

[12] “TelosB mote platform”, Document Part Number: 6020-0094-01 Rev B, Crossbow Technology, Copyright 2009

Simulation of the RPL Routing Protocol for IPv6 Sensor Networks: two cases studies

Leila Ben Saad
CITI INSA-Lyon, ENS Lyon, INRIA
Université de Lyon
Villeurbanne, France
Leila.Ben.Saad@ens-lyon.fr

Cedric Chauvenet
Watteco Inc.
CITI INSA-Lyon - INRIA
La garde, France
c.chauvenet@watteco.com

Bernard Tourancheau
CITI INSA-Lyon - INRIA
Université Lyon1
Villeurbanne, France
Bernard.Tourancheau@INRIA.fr

***Abstract*—The routing protocol for low power and lossy networks (RPL) was recently designed in the ROLL working group at IETF. Few simulation tools exist that enable its evaluation in order to prepare for its real deployment. In this paper, we provide a new evaluation of this protocol with two approaches using two different simulators adapted to our needs. We first evaluated the value of mobile sinks in wireless sensor networks to extend the network lifetime using a sensor network simulator, WSNet, augmented by our own RPL module. We then focus on the performance comparison of simulated sensor networks and real powerline communication networks (PLC) using the RPL capable COOJA simulator augmented by our own PLC module. In each case, we justify the simulator choice, describe the tools implemented and present the obtained results. Our studies give two new RPL evaluations and show the interest of choosing a simulation tool adapted to the targeted study with the associated software developments. As a conclusion, we demonstrated how these two case studies can be combined in a heterogeneous network architecture to extend its global lifetime.**

***Keywords*-Network Simulation, RPL, PLC, IPv6, Mobile Sinks, Energy Optimization, WSN, 802.15.4, Interoperability, Hybrid Network.**

I. Introduction

Recently, significant studies have been conducted to enable the convergence of sensor networks with the IP world and the connectivity of smart objects to the Internet. The IETF Working Group IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) proposed an RFC [1] to enable IPv6 packets to be carried over IEEE 802.15.4. Eventually, the IETF Working Group Routing over Low power and Lossy networks (ROLL) designed a routing protocol named IPv6 Routing Protocol for Low power and Lossy Networks (RPL). RPL was proposed because none of the existing known protocols such as AODV, OLSR or OSPF met the specific requirements of Low power and Lossy Networks (LLN), see [2]. The RPL protocol targets large scale wireless sensor networks (WSN) and supports a variety of applications *e.g.*, industrial, urban, home and buildings automation or smart grid. The ROLL working group charter stipulates that the designed routing protocol should operate over a variety of different link layers, including but not limited to low power WSN. This feature requires the RPL protocol to support heterogeneity in LLN, for instance with the use of WSN and Power Line Communication (PLC) technologies.

In this paper, we evaluate the performance of the RPL protocol in two cases dealing with low power WSN and low power PLC. This paper is organized as follows. In Section 2, related work is reviewed. Section 3 presents the RPL protocol. Section 4 describes the implemented modules for the simulation of RPL on WSNet [3] and Cooja [4]. In Section 5, the performance evaluation of RPL in the case of WSN with mobile sink nodes and PLC nodes is provided. Section 6 concludes the paper and discuss our future work.

II. Related work

Recently, several RPL simulations and implementations have been provided. In the internet draft [5], the RPL performance is evaluated by considering several routing metrics (*i.e.*, path quality, delay bound for P2P routing, routing table size, control packet overhead, loss of connectivity) in real-life deployment scenarios. The simulator used in this study is OMNET++/Castalia [6]. In [7], the authors simulated RPL on OMNET++ to analyse its stability delays. In [8][9], the authors studied the multipoint-to-point performance of RPL as well as some suggested broadcast mechanisms. The simulations have been performed on NS2. In [4][10] the authors proposed a framework for RPL simulation, experimentation and evaluation. This framework is composed of three parts: the Contiki operating system [11], the COOJA [4] / MSPSim [12] simulator and the ContikiRPL implementation [10]. At Berkeley and Johns Hopkins universities, an open-source implementation of RPL in BLIP-2.0 for TinyOS 2.x [13] is under development. We provide in the following a RPL control message simulator based on the WSNet [3] / WSim [14] WSN simulator. There are also several other RPL industrial non-open source implementations.

Despite the fact that several studies and implementations have been conducted to evaluate the performance of RPL, to our knowledge, there has been no evaluation of RPL in the case of mobile sink nodes and low power PLC nodes.

III. Presentation of the RPL protocol

RPL [15] is a routing protocol designed for low power and lossy networks and targets networks with thousands

Copyright (c) The Government of France, 2011. Used by permission to IARIA. ISBN: 978-1-61208-144-1

of nodes. RPL supports the multipoint-to-point, point-to-multipoint and point-to-point traffic. The basic idea of RPL is that the nodes organize themselves by forming a Destination Oriented DAGs (DODAGs) rooted towards one sink (DAG ROOT) identified by an unique identifier DODAGID. The DODAGs are optimized according to an Objective Function (OF) identified by an Objective Code Point (OCP), which indicates the constraints and the metrics in use [16] (*e.g.*, hop count, latency, expected transmission count, energy, …). Each node is assigned a rank which determines its relative position in the DODAG. The rank increases down et decreases up.

RPL uses the concept of DAG INSTANCE, which is a set of multiple DODAGs. A node can be a member of multiple DAG INSTANCEs but can belong to at most one DODAG per DAG INSTANCE. RPL constructs and maintains the upwards routes of the DODAGs by the transmission of DODAG Information Object (DIO) messages. DIO messages contain many informations: RPL INSTANCE, DODAGID, RANK, DODAGVersionNumber. The transmission of DIO messages by a node is regulated by a trickle timer [17] to eliminate redundant control messages. Each node monitors its neighbors' DIO messages before joining a DODAG. Then, it selects a DODAG parent set from its neighbors according to the cost they advertise and eventually computes its RANK. Destination Advertisement Object (DAO) messages are aimed at maintaining downward routes. Sending a packet to the DAG ROOT consists in selecting the preferred parent with lower rank. Any node in RPL can send a DODAG Information Solicitation (DIS) message to solicit a DIO message from its neighborhood.

To repair the topology of the DODAG and allow nodes to join a new position, the DODAG ROOT increments the DODAGVersionNumber to create a new DODAGVersion. This operation is called global DAG repair. RPL also supports other mechanisms to allow local repair within the DODAG Version. For example, the node can detach from the DODAG, advertise a rank of INFINITE_RANK to inform its sub-DODAG, and finally re-attach to the DODAG.

IV. Implemented modules for RPL simulation

A. Simulator choice

Table I compares the technical features of existing simulators. We needed open-source simulators in order to easily implement our research platforms.

We chose to simulate RPL with mobile sink nodes on the event-driven simulator for wireless networks WSNet, because the addition of any new feature does not need to modify the core of the simulator and can be done easily. Moreover, a mobility module was already implemented to ease the implementation of any moving scheme, like sink nodes for instance. Notice that when this study was started, no existing RPL open source implementation in a simulator was available.

Simulator	ns2	Castalia OMNet++	TOSSIM	Cooja/MPSim	WSim/WSNet
Level of details	generic	generic	code level	all levels	all levels
Timing	discrete event	discrete event	discrete event	discrete event	discrete event
Simulator platforms	FreeBSD, Linux, SunOS, Solaris, Windows (Cygwin)	Linux, Unix, Windows (Cygwin)	Linux, Windows (Cygwin)	Linux	Linux, Windows (Cygwin)
WSN platforms	n/a	n/a	MicaZ	Tmote Sky, ESB, MicaZ	MicaZ, Mica2, TelosB, CSEM Wisenode, ICL BSN nodes, eZ430
GUI support	Monitoring of simulation flow	Monitoring of simulation flow, c++ development, topology definition, result analysis and visualization	None	Yes	None
Wireless channel	Free space, two-ray ground refection, shadowing	lognormal shadowing, experimentally measured, path loss map, packet reception rates map, temporal variation, unit disk	lognormal shadowing	multi-path ray-tracing with support for attenuating for obstacles, unit disk, directed graph	file static, disk model, free space, tworay ground, lognormal shadowing, rayleigh fading, ITU indoor model, nakagami fading
PHY	Lucent WaveLan DSSS	CC1100, CC2420	CC2420	CC2420, TR1001	CC1100, CC1101,CC2500, CC2420
MAC	802.11, preambule based TDMA (preliminary stage)	TMAC, SMAC, Tunable MAC (can approximate BMAC, LPL, etc.)	Standard TinyOS 2.0 CC2420 stack	CSMA/CA, TDMA, X-MAC, LPP, NullMAC, contikiMAC, SicslowMAC	DCF, BMAC, ideal MAC
Network	DSDV, DSR, TORA, AODV	Simple Tree, Multi-path Rings	No data	RPL, AODV	Greedy Geographic, file static
Transport	UDP, TCP	None	No data	UDP, TCP	None
Sensing	Random process with Mannasim add-on	Generic moving time varying physical process	No data	Moving nodes	Generic moving time varying physical process
Energy consumption model	Yes	Yes	With Power TOSSIM add-on	Yes	Yes

Table I
OPEN-SOURCE SIMULATORS COMPARISON

Our PLC motes development was conducted under the Contiki OS. COOJA is the simulator natively integrated into Contiki and it was a natural choice. COOJA runs as a glue between a hardware emulator (MSPsim, Avrora) and Contiki. Thus, it can directly run Contiki OS code, without modification. As a result, the ContikiRPL implementation is directly executable in COOJA. Moreover, it has a friendly GUI that made it ideal for easy learning and prototyping at the application level. Several plugins provide a fine grained vision of the simulated network. Various media and platforms are supported, forming a good starting point for our PLC components implementation.

B. WSnet simulator

A RPL module was implemented at the network layer in the WSNet simulator according to RPL draft version 5. The main features of this module are DODAGs building, rank computation and packets forwarding. The metric used to construct the DAG and determine the rank is hop count. To build the DODAGs, DAG ROOTs start by sending DIO packets containing: RPL INSTANCE, DODAGID, RANK, DODAGVersionNumber and OF. The nodes listen for DIOs and use their informations to join a new DODAG and

ISBN: 978-1-61208-144-1

compute their rank. To that end, every node scans all its candidate neighbors and selects the current best parent by considering the OF. The nodes determine their own rank by adding the preferred parent rank to a RankIncrease value. The RankIncrease may vary from 1 to 16. Then, the nodes retransmit their own DIO packets to update the DoDAG and inform other nodes about the changes. The packets are routed to DAG ROOTs by the selection of the preferred parent with the lowest rank. The global DAG repair was implemented to reconstruct the network topology in case of broken links. The transmission of DIO messages by nodes is regulated by a trickle timer to suppress redundant control messages. The trickle timer interval for emitting DIO messages was initially fixed to one second and then incremented exponentially over the simulation time as specified in [17]. The routing module was used with chipcon radio CC1100 with 250 kbps data rate and implemented over IEEE 802.15.4 MAC and PHY layer specifications.

In the energy module, the current consumption values in transmit and receive mode were respectively fixed to 16.9 mA and 16.4 mA, as stated in [18]. In the application module, all the data packets generated by the sensors are fixed to 127 bytes (IEEE 802.15.4 MTU) and destined to the DODAG ROOT. Every minute, a packet is sent to the sink according to a Constant Bit Rate (CBR) sampling. The traffic supported in the application is multipoint-to-point. Therefore, only upwards routes were considered and DAO messages advertisement was configured to be entirely disabled. The mobility model in WSNet was modified to allow sink nodes to move according to our different moving schemes.

C. *Cooja simulator*

The ContikiRPL implementation works straightforwardly in COOJA and thus does not require any modification on our new platform. The ContikiRPL implementation is based on RPL draft version 18. It handles DIO, DIS, DAO, DAO-ACK, trickle timers management, local and global repair, ETX and Hop Count metrics.

1) PLC Nodes Implementation: None of the Hardware used in our PLC components are currently implemented in COOJA. Our PLC platform implementation relies on the existing Berkeley Telos [19] platform implementation in MPSim. This platform is composed of a MSP430 micro-controller and a CC2420, 802.15.4-compliant radio transceiver. We customized it to fit our low power PLC components [20] behavior. Notice that the Telos platform uses a f1611 version of the MSP430 MCU whereas our PLC nodes use a f5438 version. RAM/ROM capability modifications have been made to fit f5438 capabilities. A new implementation has been created in MSPsim to fit these differences. Other differences have limited impact on the PLC components performances and are not considered.

The PLC transceiver is the component with the most important impact on the node behavior, so its specificities were carefully implemented to have a precise simulation. Figure 1 shows an overview of the PLC implementation architecture in COOJA. As the MAC layer of the PLC node is implemented in the transceiver itself, new MAC drivers have been implemented in the Contiki core.

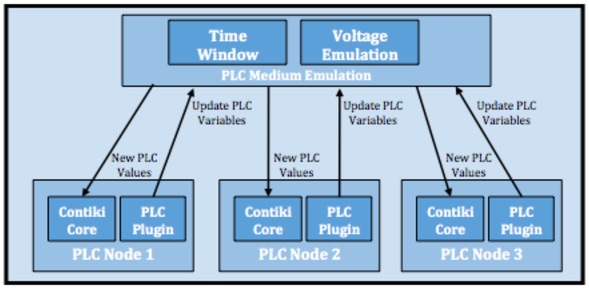


Figure 1. A simulated PLC environment in COOJA

2) Powerline medium Implementation: There is no well-adopted models for PLC simulation. We used the Directed Graph Radio Medium (DGRM) implementation of COOJA to create a PLC medium. We extended it with a node plugin in order to synchronize all simulated PLC nodes with a voltage emulation. This plugin updates the voltage emulation every 100μs on each node and triggers the computation of the communication windows on each PLC transceiver. Links are oriented, enabling to create asymmetric links, a common case in PLC networks. Every link created presents a success ratio and a delay configuration parameters. Links' delay are not relevant on PLC networks, because the speed of signal propagation on electric wires was orders of magnitude smaller than the upper networking layer delays on low power PLC. Success ratio enables to inject real link measurements into the simulator.

3) COOJA developments: Our PLC medium implementation creates a voltage emulation signal, computes the time windows where the PLC transceiver can transmit data, and updates a value in the Contiki core according to this computation. The voltage emulation consists in a sinus computation, where amplitude, frequency and phase can be set. The time window is computed according to the PLC transceiver specificities. It creates a transmitting-enable time window around the increasing zero-crossing voltage. This time window computation updates a value in the Contiki core that will impact the transceiver behavior. A PLC node plugin has been implemented to synchronize every node on the same electrical phase and trigger the time windows computation with a 100μs granularity. This plugin relies on the ”tick loop” to synchronize all simulated nodes. The PLC medium implementation triggers the PLC values computation to check if the PLC transceiver of the simulated node is able to transmit or not.

ISBN: 978-1-61208-144-1

4) Hardware Implementation: The PLC transceiver implementation in MSPsim is based on the CC2420 with data rate modification. A new chip has been created with the same architecture as the CC2420 and the symbol period has been adjusted to 16μs to fit the PLC transceiver baud rate with Hamming code correction. CC2420 can continuously transmit data whereas the PLC transceiver sends data bursts around the uprising zero-crossing of the voltage. With the hamming correction error, the PLC transceiver sends bursts of 12 bytes each 50 Hz voltage period. The implementation respects this physical indentation.

5) Contiki developments: A modified version of the CSMA implementation in Contiki has been created to handles the backoff computation and the retry mechanism of the PLC chip. Radio duty cycling (RDC) mechanisms are not used over PLC but its implementation offers useful mechanisms such as no-ack and collisions detection. For a PLC simulated node, the relevant parts of these features have been added to the original implementation to create a dedicated RDC layer. Finally, a new low level driver has been created to synchronize the transmission of the simulated PLC chips with the Contiki core variables. This reflects the time window computed into the PLC medium implementation into the COOJA simulator. This driver waits for the time window before beginning to transmit a packet. This driver also handles the chip specific CCA mechanism.

V. Performance evaluation of RPL

In this Section, we evaluate the performance of RPL on the modified simulators by considering two case studies: mobile sink nodes and PLC nodes.

A. Case of mobile sink nodes

The WSNs are often composed by a large number of battery-operated sensors, which have a limited energy supply. The sensors play at the same time the role of source nodes by generating data and relay nodes by forwarding the data of nodes farther away from the sinks. Thus, the sensors near the sinks are more likely to use up their energy much faster than distant nodes because they carry heavier workloads. Therefore, they become hot-spots. The hot-spot rapid energy depletion prevents farther nodes from relaying their data to the sinks. Consequently, the network lifetime ends prematurely. Moving the sinks even infrequently can partially solve this hot-spot problem and increase the network lifetime [21][22]. For this reason, the evaluation of the performance of RPL with multiple mobile sinks is needed to determine their best placement over time.

To evaluate the RPL performance in case of mobile sink nodes, we investigate the network lifetime (*i.e.*, the death time of the first sensor), the sensors residual energy and the packet overhead. Moreover, we make a comparative study with different mobility schemes: RPL_Static, RPL_Random, RPL_Energy, RPL_Weight. In the first scheme, the sinks are fixed. In the second scheme, the sinks are moving randomly among the sensor nodes. In the third scheme, the sinks are moving towards the nodes with the highest energy. In the fourth scheme, the sinks are moving towards the leaf node of the DODAG, which has the highest weight w_i [23]. This weight is a function of three parameters influencing the network lifetime: h_i^k is the number of hops from sensor node i to its DAG ROOT at position k, e_i is the residual energy of sensor node i and b_i is the number of its 1-hop neighbors. The exact weight calculation is as follows: $w_i = \beta h_i^k e_i + \gamma b_i$ where β and γ are coefficients of normalization. They mitigates the effect of scale since the measurement units are different. The moving schemes are performed only during the periods multiple of the periods of DAG repair. The number of sensors used in the simulation ranges from 100 to 1600 nodes whereas the number of sinks is fixed to three.

Figure 2 shows the lifetime gain as a function of the network size for different moving schemes with respect to the case of RPL with static sinks. The results shows that the lifetime improvement increases with the size of the network. This straightforwardly proves that using mobile sinks in RPL is more beneficial in large scale networks. It is also observable that the lifetime gain obtained in RPL_weight scheme is better than the other strategies independently of the size of the network. Moreover, the lifetime improvement induced is about 24% in network with 1600 nodes.

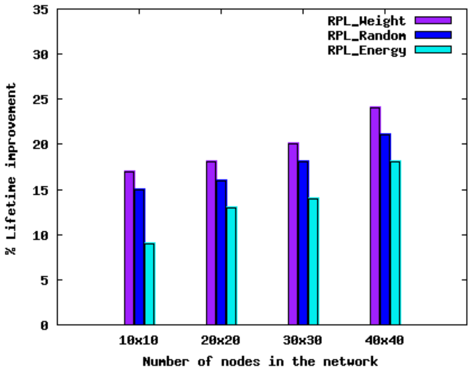


Figure 2. Network Lifetime improvement as function of network size

Figure 3 compares the percentage of sensors residual energy as function of the network size at network lifetime end. The energy left unused at the end of network lifetime in mobile sinks schemes is notably lower than in the case of static sinks. This is due to the fact that sinks mobility changes the nodes acting as relays frequently and leads to balanced energy consumption among nodes. Nevertheless, RPL_Weight results in the best distribution of the available energy on the sensors since it leaves the smallest amount of unused energy at the end of network lifetime.

In Figure 4, we analyze the amount of data packets transmitted (including forwarded) and the ICMPv6 control

ISBN: 978-1-61208-144-1

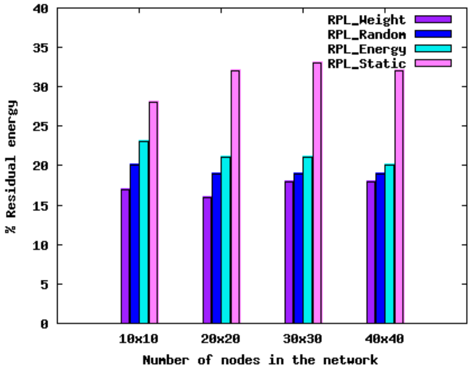


Figure 3. Pourcentage of sensors' residual energy at network lifetime end.

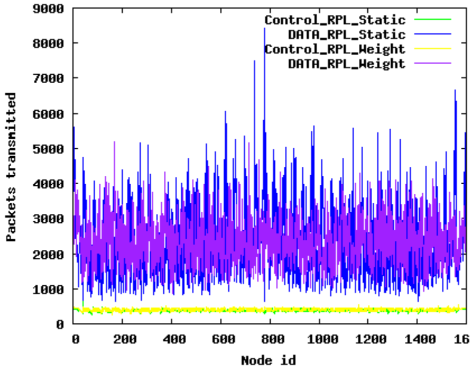


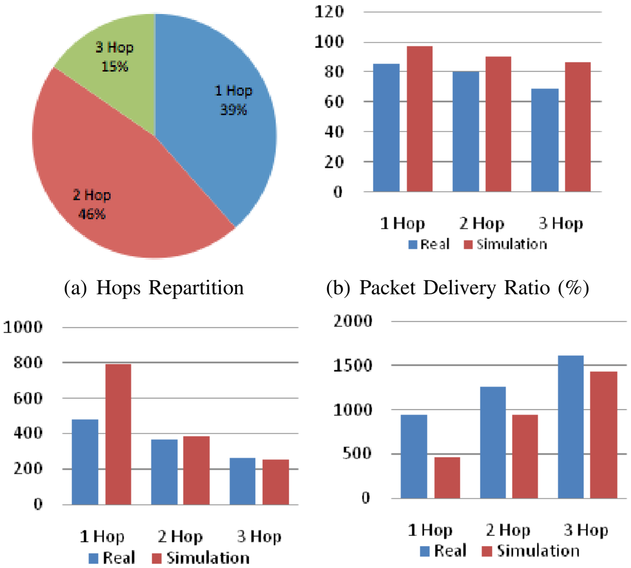
Figure 4. Packets transmitted : Control packets and Data packets (including forwarded data)

packets (DIO messages) transmitted by each node. With RPL_Static scheme, the nodes near the sinks (*e.g*, node id 778) has more data traffic than other nodes because they have to transmit their own data in addition to farther away nodes data. However, for leaf nodes (*e.g.*, 1401), the amount of data packets transmitted is smaller than middle or close to the sink nodes ones. This is because they do not have to act as forwarding nodes. By moving the sinks according to RPL_Weight, the nodes playing the role of relay nodes change and the data traffic becomes more balanced among all the nodes. As shown in Figure 4, the majority of nodes have a comparable amount of data packets transmission. Moreover, the control overhead is very small in comparison to data packets. It is also not highly increased in spite of the mobility of sinks. This can be explained by the fact that the sinks move only during the periods of DAG repair.

B. *Case of PLC nodes*

PLC nodes are not energy constrained, so that they can play the role of sinks presented in the previous Section. Relying on the IPv6 design, and the 802.15.4 adaptation over PLC presented in [24], a lightweight IPv6 hybrid stack was designed over PLC and 802.15.4 with a unique 6LoWPAN adaptation [25]. As a result, these sinks become PLC-RF bridges that form a PLC backbone to connect the wireless network. Considering the hypothesis of a limited number of sinks, we consider that a small amount of PLC nodes will be equipped with a dual physical stack. According to the previous proposition, these bridges will be moved periodically to distribute energy consumption efficiently. In such a context, we should determine the ability of the PLC network to fulfill a ”LLN backbone” role. Moreover, depending on the traffic volume and RF performance, the PLC backbone may induce losses, additional latency and/or decrease the overall throughput across the network.

In order to evaluate the performances of this backbone, we measured the performances of a real and a simulated PLC network implementing the RPL network stack. We observed hops distribution, packet delivery ratio (PDR), throughput and latency. Our test bed was a 2 floors research laboratory, composed of 25 rooms. We used 6 PLC nodes and a border router. PLC nodes were randomly plugged in outlets. The Border router was never moved. After topology establishment, the border router sends 3 series of 30 pings to each node it has in its routing table with a delay of 2 seconds per hop between each ping. Once the 3 series of 30 pings were done, we moved all the PLC nodes into a new room, and repeated the scenario. The power grid electrical network was impacted by daily life activity. The simulation platform first replayed the scenarios in the testbed topology but with ideal links in order to quantify the looseness of the PLC media. The simulated nodes used the same software as real nodes.



(a) Hops Repartition

(b) Packet Delivery Ratio (%)

(c) Throughput (bps)

(d) Latency (ms)

Figure 5. Performances of real and simulated PLC network

Figure 5(a) shows that from the border router location, the RPL protocol reached all the 6 PLC nodes in any

ISBN: 978-1-61208-144-1

room through a 3 hops maximum path. This points out the reliability, connectivity and forwarding cost reduction that is potentially available in such hybrid networks. Furthermore, this also shows that a small amount of PLC nodes may be enough to form a PLC backbone. For instance, the previous hypothesis of 3 sinks in Section V.A, shows the gain that can be obtained using 3 RF-PLC bridges for an entire small building. As expected, in Figures 5(b), 5(c), 5(d) the performances of the PLC network for PDR, Throughput and Latency decrease with path length *e.g.*, number of hops. Though, the maximum paths' length of 3 limits the performances downgrade. Throughput is less impacted by real PLC links because it is only computed for successful transmissions. Latency performance shows that real PLC links induce more link layer retries on real PLC networks. Notice that in the simulation, even with ideal links, 100% PDR is not reached because of collisions with control messages traffic.

VI. Conclusion

Our studies show that there are several possibilities for LLNs simulation. In particular, the RPL routing protocol is already supported in Contiki/COOJA. However, WSNet provides interesting capabilities for mobility management.

Our research provides new functionalities either in WSNet with the implementation of RPL for our needs in the context of sink mobility and in COOJA with the support of a new networking hardware, namely low power PLC.

With these improved simulators, the conducted experiments show the interest of RPL simulation in order to improve WSN lifetime by managing the sink mobility and to provide coherent routing in LLN heterogeneous platforms with wireless and PLC sensor networks.

References

[1] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 packets over IEEE 802.15.4 networks," IETF, RFC 4944, 2007.

[2] P. Levis, A. Tavakoli, and S. Dawson-Haggerty, "Overview of existing routing protocols for low power and lossy networks," Draft, 2009.

[3] WSnet simulator http://wsnet.gforge.inria.fr/, 2010.

[4] N. Tsiftes, J. Eriksson, N. Finne, F. Österlind, J. Höglund, and A. Dunkels, "A framework for low-power ipv6 routing simulation, experimentation, and evaluation," *SIGCOMM Comput. Commun. Rev.*, vol. 40, pp. 479–480, August 2010.

[5] J. Tripathi, J. de Oliveira, and J. Vasseur, "Performance Evaluation of Routing Protocol for Low Power and Lossy Networks (RPL)," IETF, Draft, January 2011.

[6] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Int Conf on Simulation Tools (ICST)*. ICST, 2008.

[7] M. Nuvolone, "Stability analysis of the delays of the routing protocol over low power and lossy networks," Master's thesis, Sweden Royal Institute of Technology, 2010.

[8] T. H. Clausen and U. Herberg, "Comparative Study of RPL-Enabled Optimized Broadcast in Wireless Sensor Networks," INRIA, Tech. Rep. 7296, 2010.

[9] ——, "Multipoint-to-Point and Broadcast in RPL," INRIA, Tech. Rep. 7244, 2010.

[10] N. Tsiftes, J. Eriksson, and A. Dunkels, "Low-power wireless IPv6 routing with ContikiRPL," in *Int Conf on Information Processing in Sensor Networks*. ACM/IEEE, 2010.

[11] A. Dunkels *et al.*, "Contiki-a lightweight and flexible operating system for tiny networked sensors," 2004.

[12] J. Eriksson, A. Dunkels, N. Finne, F. Osterlind, and T. Voigt, "MSPsim-an extensible simulator for MSP430-equipped sensor boards." 2007.

[13] M. Afanasyev, D. O'Rourke, B. Kusy, and W. Hu, "Heterogeneous Traffic Performance Comparison for 6LoWPAN Enabled Low-Power Transceivers," 2010.

[14] Wsim Simulator http://wsim.gforge.inria.fr/.

[15] T. Winter and P. Thubert, "RPL: IPv6 routing protocol for low power and lossy networks," IETF, Draft, 2010.

[16] J. Vasseur, M. Kim, and K. Pister, "Routing metrics used for path calculation in low power and lossy networks."

[17] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm," IETF, Draft, 2010.

[18] CC1100 datasheet focus.ti.com/lit/ds/symlink/cc1100.pdf.

[19] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Information Processing in Sensor Networks (IPSN)*. IEEE, 2005, pp. 364–369.

[20] Watteco, "Wpc-ip product brief," www.watteco.com, 2011.

[21] L. Ben Saad and B. Tourancheau, "Towards an optimal positioning of multiple mobile sinks in wsns for buildings," *Int J On Advances in Intelligent Systems*, vol. 2, no. 4, 2009.

[22] ——, "Multiple mobile sinks positioning in wireless sensor networks for buildings," in *Int Conf on Sensor Technologies and Applications (SensorComm)*. IARIA, 2009.

[23] ——, "Sinks Mobility Strategy in IPv6-based WSNs for Network Lifetime Improvement," in *Int Conf on New Technologies, Mobility and Security (NTMS)*. IFIP, 2011.

[24] C. Chauvenet, B. Tourancheau, and D. Genon-Catalot, "802.15.4, a MAC layer solution for PLC," in *AICCSA*. ACS/IEEE, 2010.

[25] C. Chauvenet, B. Tourancheau, D. Genon-Catalot, P.-E. Goudet, and M. Pouillot, "Interoperable IPv6 sensor networking over PLC and RF media," *IJBDCN*, vol. 6, no. 3, 2010.

Benchmarking for Wireless Sensor Networks

Jono Vanhie-Van Gerwen, Stefan Bouckaert, Ingrid Moerman, Piet Demeester
Department of Information Technology (INTEC)
Ghent University - IBBT
Ghent, Belgium
Email: jono.vanhie@intec.ugent.be

***Abstract*—While the number of Wireless Sensor Network (WSN) protocols steadily increases, the evaluation methods have largely remained the same. Although experimentally-supported research is gaining popularity, protocol evaluation and comparison remains difficult due to a lack of performance analysis methodologies. This work introduces a wireless-benchmarking workflow that is designed to support experimentally-driven analysis of WSN protocols. This methodology and the accompanying benchmark concepts are designed to increase the value of experimental performance evaluation compared to the current ad-hoc approaches applied by many researchers. Finally, we present a proof of concept implementation used to perform experiments based on the proposed workflow.**

***Keywords*-Wireless Sensor Networks; benchmarking; performance; evaluation.**

I. Introduction

Wireless Sensor Networks (WSNs) are evolving into a more mature research field, with more and more protocols being developed and publicly released. The use cases for WSNs are equally fast expanding into new domains, such as Wireless Building Automation [1] and Cognitive Networks [2]. Recently, the interest of the research community in experimentally-driven research in wireless networks is increasing. This observation is reflected in the topics of international conferences that increasingly welcome experimentally-driven research, and the recent interest of the European Commission in experimental facilities [3]. Still, when WSN protocols are developed, research efforts are often focused on the isolated programming of single layer protocols with little regard for other layer functionality or restrictions. This leads to protocols that exist in a vacuum and perform well on a theoretical basis, but have problems when deployed under real-life circumstances [4], [5]. Many of these protocols are furthermore validated using ad-hoc created experimental tests, specifically aimed at the strengths of a specific protocol, leaving little room for objective comparison with other protocols.

Currently, there exists no fixed set of accepted testing methods, scenarios, parameters or metrics to be applied on a protocol under test. This lack of standardization significantly increases the difficulty for a developer to assess the relative performance of their protocols compared to the current state of the art. As a solution to these problems, we introduce a benchmarking methodology for WSN research to increase the value of performance evaluations.

We define WSN benchmarking as the measurement and evaluation of wireless sensor protocols, devices and networks, relative to a reference evaluation, under well documented conditions. As a result, benchmarking differs from traditional ad-hoc experiments by following a commonly accepted methodology that covers the entire experiment life cycle, making experiments reproducible and the results directly comparable with other research results. In WSN research, this requires not only evaluating the device or protocol under test, but also measuring or modeling the wireless environment.

We have used the presented workflow in our previous work [6], where the real-life performance of five popular MAC and routing protocols for TinyOS [7] was evaluated. While [6] focuses on the results, this papers complements it by focusing on the used experimentation methods and the developed implementation to reach these results.

Our methodology is specifically aimed at real-life experimentation and testbeds. Simulations can reduce the duration of tests dramatically. Unfortunately, the real-life value of the obtained results does not necessarily increase [8]. To truly know the performance and characteristics of a network protocol, it should be benchmarked in multiple real-life environments under various realistic conditions. Within the European seventh Framework Programme, the CREW (Cognitive Radio Experimentation World) project [9] targets the development of an open federated test platform, facilitating experimentally-driven research on cognitive radio, cognitive networking, and spectrum sensing. An important aspect of CREW is the creation of a wireless benchmarking framework, which should allow fair comparison of wireless (cognitive) solutions, by creating a reproducible test environment, as well as traffic and interference models needed for emulating realistic applications and interference sources, eventually establishing a methodology for the evaluation of cognitive radio concepts.

After first reviewing the related work, the benchmarking workflow is presented in Section III, together with the concepts that drive it. Next, the proof of concept implementation of our WSN benchmarking methodology is discussed in Section IV, before presenting our future work in section V.

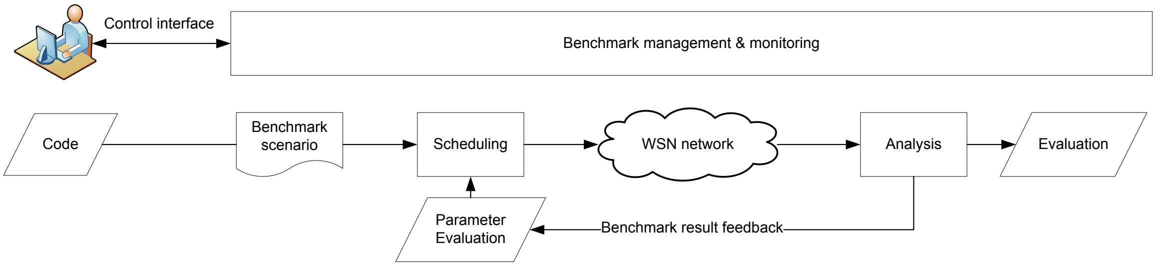


Figure 1. Benchmarking workflow

II. Related work

Although benchmarking of (wireless) networks is currently a research topic in multiple European research projects such as CREW [9], BonFIRE [10], or OneLab(2) [11], few benchmarking frameworks for wireless network protocol design are currently found in literature. Among the work that can be found, TinyBench [12] focuses solely on the internal metrics of a single sensor node; In the work of Kim et. al. [13], only a single layer in the protocol stack per benchmark execution is observed. Similarly, in [14], the authors explore black-box benchmarking of application protocols in network protocol stacks. As their focus is only on the performance of the application, they choose not to measure the performance of individual networking protocols. Benchmarks of individual characteristics of WSNs are furthermore also found in [15], where the authors develop a benchmarking methodology to evaluate the performance of sensor processors. To this end, the authors compose different basic benchmarking applications based on modular application building blocks.

In contrast with our presented benchmarking approach, the works cited above analyze only a single protocol layer. While this simplifies analysis and benchmark execution, the approach cannot provide information on layer incompatibilities between different sets of protocols, nor gives information on the overall performance of a network which is built based on the protocols under test. Benchmarking on a larger scale and scope by evaluating a complete protocol stack while observing an entire sensor network delivers a better insight into the real-life performance of a WSN application.

In other studies such as [16] and [17], the term ‘benchmarking’ *is* used to evaluate the performance of a network as a whole. In the former technical note, the authors give an overview of do’s and don’ts and hints for the performance analysis of Wireless LAN access points in terms of throughput and rate vs. range performance. In the latter document, the authors note that no third-party benchmarking solutions seem to be available and then resort to an ad-hoc benchmarking approach to compare the three wireless mesh networking platforms, mainly in function of throughput and coverage predictability. Unfortunately, while these authors do compare the overall performance of different wireless solutions, the lack of generic benchmarking solutions makes that their approach can only be used for their specific cases. Ad-hoc benchmarking solutions may lead to faster results when only needing to compare a fixed set of wireless protocols or wireless devices once, however, the reproducibility and level of details obtained through these tests is generally limited.

To the best of our knowledge, none of the related works present a sustainable benchmarking workflow, fully and flexibly supporting the benchmarking of wireless sensor solutions on testbeds. In the next sections, we present our generally applicable benchmarking solution, which was successfully used to reliably evaluate the performance of multiple full wireless networking stacks [6].

III. Benchmarking workflow

In Figure 1, we propose a benchmarking workflow responsible for defining, executing and analyzing individual experiments. By incorporating the entire experiment life cycle in our methodology, benchmark integrity and correctness is enforced. We describe the separate components in further detail to indicate their individual importance.

A. Benchmark scenario

A benchmark is fully defined by a *scenario*. A scenario is the global description of a benchmark and is based on real-life use cases or artificial test environments. For example, a scenario could describe the specific traffic patterns and network load of an intrusion detection use case. Such scenario properties are defined using three different scenario components: *(i)* criteria, *(ii)* parameters and *(iii)* metrics. It is the detailed description of these scenarios that can form a baseline for benchmark standardization.

A *criterion* defines the focus of a benchmark, and the characteristics to be examined. To establish a broad view on the performance of a sensor network, it is generally necessary to combine multiple criteria in the evaluation, depending on the requirements of a specific deployment or comparison. If the scenario is based on a real-life use case, the criteria should be chosen to reflect the most important characteristics of the use case. For example, when node lifetime in a battery powered network is important, energy efficiency is an appropriate criterion.

A scenario contains a number of *parameters*, allowing fine grained control over the execution of a benchmark. Depending on the scope of a parameter, two categories are distinguished, i.e. internal and external parameters. External parameters control the node behavior, e.g. the amount of data it sends, to whom, and at what rate. The scenario dictates in what range the external parameters can vary while staying in the same scenario, e.g. for a building automation temperature monitoring scenario the reporting interval can vary between five seconds and ten minutes. External parameters are also responsible for defining the environment, emulating interference or background traffic.

Internal parameters on the other hand affect the settings of the evaluated protocols, e.g. route table size, sleep interval, queue size. Internal parameters are most often used for protocol optimization, where a designer wants to increase the performance of his own protocol for a specific use case.

Each internal or external parameter can additionally be a function of the time, changing depending on the progression of the benchmark if the scenario requires it. This can be useful to simulate the dynamic behavior of the evaluated protocols or the environment.

Finally, the *metrics* are the aspects from the benchmark execution that are logged and evaluated. The chosen metrics should reflect the focus of the benchmark criterion and provide information to form well supported conclusions. These metrics are performance metrics and are dependent on the execution of the benchmark, e.g. throughput, packet error rate. In a final phase, all metrics are translated to a single benchmark score using appropriate weights defined in the scenario. This can also be a combination of the measured performance metrics with predefined business metrics, e.g. cost or operational complexity.

In Table I an example of the relationship between criteria and metrics is given, with examples of external parameters.

These scenarios should be standardized so that they define which benchmarks should be applied for different use cases. This includes topology and environment information to complete the external parameters of a specific benchmark. Both internal and external parameters should be well defined so that a repeated benchmark delivers the same node behavior, regardless of the specific framework implementation. A common format for benchmark definition and result description is essential to obtain relevant evaluations across different platforms. These efforts will increase the adoption and value of a benchmarking solution.

B. Scheduling

The scheduling component is responsible for executing a benchmark with the correct parameters on the WSN testbed. A benchmark performed in isolation rarely provides enough information to form well-founded conclusions. For a full protocol analysis multiple benchmarks are needed, varying parameters to form a global view. There often exists a correlation between groups of benchmarks scanning a parameter space, e.g. when evaluating the delay metric by adjusting the radio duty cycle, packet interval and network size the three parameters will be correlated. If this correlation is exploited the amount of required experiments can be drastically reduced, while increasing the value of the performed benchmarks. The scheduling and analysis components form a tightly coupled feedback loop to enhance these effects present in many benchmarks and use result feedback to adjust benchmark parameters. We consider two types of benchmark relationships:

criteria \ metrics	delivery rate	application throughput	energy use	experiment duration	footprint	end-to-end delay
energy			X	X		
reliability	X			X		X
delay	X					X
throughput		X		X		
stability	X					X
scalability			X		X	
memory					X	

parameters					
communication	broadcasting	single sink	multiple sinks	point-to-point	hybrid
traffic	packet size	% sending nodes	packet interval	event impact	variance
network	network size	topology	node density	interference	channel

Table I
EXAMPLES OF CRITERIA, PARAMETERS AND METRICS

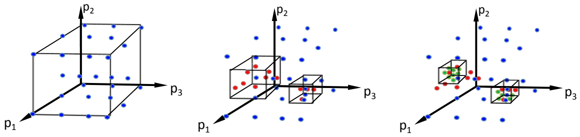


Figure 2. Parameter space reduction with three iterated benchmark series

Iterative benchmarks: are correlated by the common parameter space that is scanned. The results of a series of benchmarks can be used to tighten the parameter space in subsequent benchmarks. A series of sampling benchmarks can help to define the result space, so that new benchmarks can be selected to focus on interesting parameter intervals. This concept is illustrated by Figure 2, where the parameter space defined by three parameters is tightened around two phenomena by iterating a series of benchmarks, changing the parameter granularity and range. The dots represent individual experiments.

Conditional benchmarks: present a different type of coupling, where the successful execution of a certain benchmark is required before the coupled benchmark can be executed. This occurs when a relationship between two benchmarks is strict, i.e. a benchmark has no value if a previous benchmark did not yield the required result. To cope with sequential benchmarks a hard decision relationship is modeled between consecutive benchmarks. If these hard choices are implemented, insignificant benchmarks can be canceled or altered, saving evaluation time.

C. Analysis

The analysis component is responsible for processing measurement data during and after the experiment using the predefined metrics of the benchmark scenario. An important factor of benchmarking is the level of visibility required to perform a correct analysis. Visibility is defined in [18] as the energy cost of diagnosing the cause of a failure or behavior.

A high visibility grants insight in the internal metrics of the protocol or device under test, e.g. queue depth, processing delay and processor load. The higher your visibility requirements are, the more difficult it is to obtain that insight in a real-life testbed without altering the behavior of the protocol under test. This is also known as the Heisenbug problem [19], based on the Heisenberg uncertainty principle, where the internal observation of the program under test changes its behavior compared to an unmonitored execution.

A low visibility is the most straightforward to obtain in a testbed since no in-depth metrics are monitored when the sensor node is seen as a black-box. But if little or no information is available on the internal metrics of a protocol or sensor node, then the analysis is limited in its scope.

Visibility in benchmarking is additionally considered as the complexity cost of the diagnosis. A benchmarking framework should be capable of regulating the visibility of each benchmark to minimize the complexity of the analysis, but still achieve the desired results.

Monitoring is the live analysis of the environment and meta information of a WSN experiment, and a crucial component of any benchmarking system. To detect unwanted effects such as node crashes, clock drift and external interference, monitoring the benchmark and its environment should be an important part of a benchmarking framework. Without this feature, the quality of benchmarking results decreases as the nondeterminism of a benchmark increases. A benchmarking system should not only anticipate this nondeterminism, but instantly decrease its influence [20]. This can be done by adjusting the run length of each benchmark, depending on the measured stability and variance of the recorded metrics and the environment while repeating a given benchmark multiple times to compensate for outliers. This implies that not only the WSN devices itself should be monitored, but also the wireless medium. For this purpose a number of WSN nodes or specialized devices should be deployed to scan the environment and model the external interference to the benchmark, so that a complete coverage of the experiment area is obtained. If the measured environment does not map to the defined environment in the benchmark scenario, the results should be discarded or decreased in value, depending on the measured deviation.

D. Evaluation

In our benchmarking workflow, two forms of evaluation are identified. The first evaluation is the most straightforward, where the metrics results are translated to one or more easy to understand benchmark scores. A second evaluation is the parameter evaluation that occurs in the feedback loop between the analysis and the (re)scheduling of a benchmark, where the original parameters of a benchmark are adjusted.

Benchmark score: is the result of an additional processing of the performance metrics to more directly comparable scores. This process is always a reduction of the metrics data, so that a reliable intuitive comparison is possible between benchmarks, only based on the benchmark score. Depending on the type of benchmark a capped score can be presented in the form of a percentage or an uncapped score as a real number. For uncapped scores, the reference benchmark metrics results are mapped to a designated score, e.g. 0, 1000 or 10000.

Parameter evaluation: uses the result feedback from the analysis to refine the chosen parameters of a benchmark. This form of evaluation enables the use of correlation effects in the scheduling step as described in III-B. From the metrics of a series of previously executed correlated benchmarks interesting parameter sets are identified and additional benchmarks are scheduled with adjusted parameters to refine the benchmark results.

IV. Proof of concept implementation

In order to validate the concepts of the described benchmarking workflow, we implemented a proof of concept benchmarking system, enabling the automated performance evaluation of multiple protocol implementations. While the benchmarking framework is implemented on a specific wireless test environment, it is very loosely coupled and therefore applicable to multiple testbed environments with minor adjustments.

The specific benchmark implementation in this paper is built on the IBBT w-iLab.t testbed [21]. With a capacity of 200 sensor nodes, deployed across three floors of an office environment, the testbed provides ample measurement accuracy and size. Each Tmote Sky sensor node [22] is connected to a central database, so that every action and state of the entire sensor network can be monitored centrally.

The implemented framework consists of four cooperating components, given schematically in Figure 3. The main components are *(i)* the TinyOS code, *(ii)* the configuration software, *(iii)* the testbed and *(iv)* the result analysis. These components are briefly discussed below, and shown how they address the needs of our benchmarking system and map to the proposed workflow.

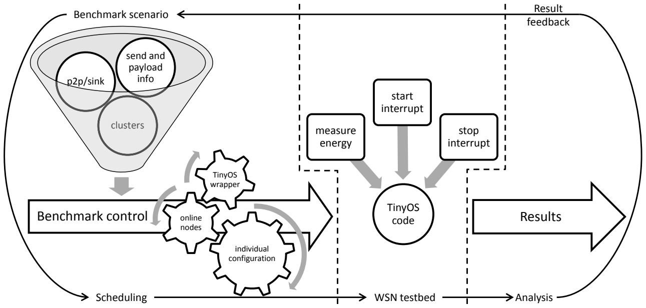


Figure 3. Components of the implemented benchmarking framework

The TinyOS benchmarking code: is the enabling factor of our framework, built on the popular operating system for wireless sensor nodes. This component allows the selection of MAC and routing protocols *at compile time*, with minimal requirements on the protocol implementation. This code has to be included in the WSN firmware so that the other components of the framework are able to communicate with the individual nodes. The code largely consists of generic timers and components, so little execution logic is embedded in the static code for the sensor nodes. Since control and execution is very loosely coupled it is straightforward to port our code to other operating systems, hardware or platforms.

The configuration software: centralizes the logic and control of our benchmarking system. This component translates the formal benchmark scenarios defined in III-A to specific configuration messages for the individual sensor nodes. It also has full connectivity through the testbed infrastructure to the WSN nodes and fulfills the functions of the scheduling component of our workflow. Each sensor node in the network only receives local configuration details and has no global knowledge of the network, corresponding with a real-life situation. Each individual configuration message is stored in a central repository, so that a benchmark can be executed multiple times with exactly the same configuration. This enhances the comparability and since the source code is written in Java, the same benchmarking logic and algorithms can easily be distributed to other systems.

The testbed: enables the execution of a benchmark on the available hardware and provides an important monitoring and control component with the "Environment Emulator" (EE). The EE is connected through a USB interface with the sensor nodes of the testbed and allows the emulation of battery level, the measurement of external metrics such as power consumption and the generation of accurate interrupts. For these features, a direct connection is made with the general purpose pins of the Tmote Sky node, allowing communication and measurement without using the blocking functions of the UART. The inclusion of the EE allows the synchronization of the benchmark and the sensor nodes. The sensor nodes connected to the testbed are configured with a management IP address. This way, each sensor node can receive a series of configuration packets at the start or during the benchmark, controlling the exact execution of the node during the benchmark.

Result analysis: is performed on the result set generated by the TinyOS component during or after the benchmark. The analysis component is fully decoupled from the rest of the framework and can work on any result set as long as the data format is provided before the analysis. The proof of concept application is provided with two analysis tools, one written in Java for real-time experiment monitoring and another composed of MATLAB scripts to generate graphs for each active metric.

Evaluation: is not yet an automated process in the current proof of concept. Both forms of the described evaluation are supported, but must be done manually. The evaluation of the metric results is aided by the automatic generation of graphs based on the types of metrics in a scenario, after which an analyst can draw the correct conclusions. The parameter evaluation after the result feedback is also a manual process, where based on the Design of Experiments [23] a parameter space is scanned, starting from coarse parameter choices to fine variations, based on the areas of interest identified by the analyst.

The benchmarking framework implementation is currently used in our research department to evaluate protocol compatibility and real-life performance. The proof of concept does currently not include all advanced features, nevertheless some interesting results are already achieved with the framework. An in depth analysis of the results obtained using this benchmarking framework is available in [6].

V. Future Work

The current proof of concept implementation is to be expanded with the more advanced monitoring and coupling aspects. An important improvement would be the further automation of the benchmarking framework, where benchmarks are automatically scheduled and refined based on previous runs. This could drastically shorten the time spent on evaluating and analyzing the raw data from the experiments, while also increasing accuracy and reducing the time needed to achieve reliable results [24].

The final results of the performed benchmarks are currently aggregated in easy to understand graphs of different scope, allowing a complete overview of the measured performance characteristics. Although complete, this method still requires an analyst to draw conclusions from the metric data. These conclusions are not always straightforward and require a certain familiarity with the subject. It would make the benchmarking system more practical if the results could be translated in a series of scores on different criteria, leading to straightforward conclusions about the performance of the evaluated system.

We are also looking to create a full integration of our configuration software in the web management system of the IBBT testbed. This will enhance the usability of the framework, while also drastically expanding the user base. This transition and the usage of a central database for all framework users will allow the possibility of easy benchmark sharing. Specific configurations and updates can easily be disseminated throughout the user community, improving the standardization and comparability.

VI. Conclusion

In this work it was argued that common problems exist in the development and evaluation of WSN protocols and applications, that lead to difficult deployments and real-life performance confusion.

The proposed benchmarking workflow tries to solve these experimentation problems, based on the methods used in our experimentally driven research. This methodology focuses on benchmark correctness and comparability, starting from a rigorous benchmark definition by scenarios. Next, the scheduling step in tandem with benchmark monitoring is responsible for the correct and most efficient execution of a benchmark. Finally, the analysis and evaluation give a clear and comparable view of the benchmark results. This has led to a benchmarking workflow that enables the qualitative comparison of WSN solutions in the least restrictive way.

By using the presented methodology, experimenters will reach more reliable results that are comparable with the results of others using this benchmark workflow. It is shown how this benchmarking methodology can be translated to a proof of concept implementation, realizing an essential subset of the proposed benchmarking concepts, capable of evaluating and analyzing network protocols and their combinations in a real-life environment.

We hope this work will be a first step to a standardized benchmarking system, assisting protocol developers and users in the selection and optimization of many existing and future protocols.

VII. Acknowledgements

Jono Vanhie-Van Gerwen thanks the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) for funding part of this research through a grant. Part of the research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n 258301 for the CREW project.

References

[1] W. Vandenberghe, B. Latré, F. De Greve, P. De Mil, I. Moerman, and P. Demeester, "A system architecture for wireless building automation," *15th IST Mobile & Wireless Communications Summit 2006, Proceedings*, 2006.

[2] E. De Poorter, B. Latré, I. Moerman, and P. Demeester, "Symbiotic networks: Towards a new level of cooperation between wireless networks," *Wirel. Pers. Commun.*, vol. 45, no. 4, pp. 479–495, 2008.

[3] European Commission, "Ict work programme 2009-10 for ict research in fp7," July 2009.

[4] M. Ali, T. Voigt, U. Saif, K. Rmer, A. Dunkels, K. Langendoen, J. Polastre, and Z. A. Uzmi, "Medium access control issues in sensor networks," *SIGCOMM 2006*, pp. 33–36, 2006.

[5] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, "The Hitchhiker's Guide to Successful Wireless Sensor Network Deployments." in *SenSys*, 2008, pp. 43–56.

[6] J. Vanhie-Van Gerwen, E. De Poorter, B. Latré, I. Moerman, and P. Demeester, "Real-life performance of protocol combinations for wireless sensor networks," in *SUTC*, jun. 2010, pp. 189 –196.

[7] TinyOS, "open-source os for wireless embedded sensor networks. http://tinyos.net (last accessed april 2, 2011)."

[8] C. J. Sreenan, S. Nawaz, T. D. Le, and S. Jha, "On the sensitivity of sensor network simulations." in *VTC Spring*. IEEE, 2006, pp. 1043–1047.

[9] CREW project, "Home page," http://www.crew-project.eu/ (last accessed March 22, 2011).

[10] BonFIRE project, "Home page," http://www.bonfire-project.eu/ (last accessed March 22, 2011).

[11] OneLab, "Home page," http://www.onelab.eu/ (last accessed March 22, 2011).

[12] M. Hempstead, M. Welsh, and D. Brooks, "Tinybench: The case for a standardized benchmark suite for tinyos based wireless sensor network devices." *LCN*, pp. 585–586, 2004.

[13] T. Kim, J. Kim, S. Lee, I. Ahn, M. Song, and K. Won, "An automatic protocol verification framework for the development of wireless sensor networks," in *TridentCom*, 2008.

[14] P. Gunningberg, M. Bjorkman, E. Nordmark, S. Pink, P. Sjodin, and J.-E. Stromquist, "Application protocols and performance benchmarks," *IEEE Communications Magazine*, vol. 27, no. 6, pp. 30 –36, jun. 1989.

[15] L. Nazhandali, M. Minuth, and T. Austin, "Sensebench: toward an accurate evaluation of sensor network processors," oct. 2005, pp. 197 – 203.

[16] Farpoint Group, "Benchmarking wirless lans: Recommended practice," Farpoint Group, Tech. Rep., 2006.

[17] Novarum Inc., "The value of smart antennas: Campus mesh network performance benchmark," Novarum Inc., Tech. Rep., January 2010.

[18] M. Wachs, J. I. Choi, J. W. Lee, K. Srinivasan, Z. Chen, M. Jain, and P. Levis, "Visibility: A new metric for protocol design," in *SenSys*, 2007.

[19] J. Gray, "Why do computers stop and what can be done about it?" Tandem, Tech. Rep. 85.7, 1985.

[20] T. Kalibera, L. Bulej, and P. Tuma, "Benchmark precision and random initial state," in *SPECTS*. SCS, 2005, pp. 853–862.

[21] S. Bouckaert, W. Vandenberghe, B. Jooris, I. Moerman, and P. Demeester, "The w-iLab.t testbed," in *Tridentcom*, May 2010.

[22] Moteiv, "Tmote Sky Datasheet," 2007. [Online]. Available: http://www.sentilla.com/pdf/eol/tmote-sky-datasheet.pdf

[23] D. C. Montgomery, *Design and Analysis of Experiments*. John Wiley and Sons Inc, 2004.

[24] T. Kalibera, J. Lehotsky, D. Majda, B. Repcek, M. Tomcanyi, A. Tomecek, P. Tuma, and J. Urban, "Automated benchmarking and analysis tool," in *valuetools*, 2006.

Enhancing The Performance Of Neural Network Classifiers Using Selected Biometric Features

Heman Mohabeer, K.M. Sunjiv Soyjaudah and Narainsamy Pavaday
Faculty of Engineering
University of Mauritius
Reduit, Mauritius
heman.mohabeer@gmail.com, ssoyjaudah@uom.ac.mu, n.pavaday@uom.ac.mu

***Abstract*— This paper describes an application which increases the overall efficiency of a neural network classifier intended for authentication whilst using fewer biometric features. Normalization of the biometric data is generally performed to remove unwanted impurities. However, in this case, when performing normalization, the statistical property for each set of data has also been taken into consideration prior to the classification process. Combination of the normalized biometric features has been performed while comparing their standard deviation. The resulting fused data has correlation value as low as possible. This gives rise to a higher probability of uniquely identifying a person in feature space. The proposed system is intended to make authentication faster by reducing the number of biometric features without degrading the overall performance of the classifier. The performance of the classifier was computed using the mean square error (MSE). The results show that redundant biometric data can indeed be excluded without degrading the performance of the classifier.**

Keywords-mean square error; normalization; biometric sample; keystroke dynamics.

I. Introduction

Biometric systems have been successfully applied as a method of authentication to replace conventional access controls [1][2]. Biometrics is the automated method of recognizing a person based on physiological or behavioral characteristics. Biometric data are highly unique to each individual, easily obtainable non-interferingly, time invariant (no significant change over a period of time) and distinguishable by human without much special training [3]. Enrollment and authentication are the two primary processes involved in a biometric security system. Enrollment consists of biometric measurements being captured from a subject. The related information from the raw data obtained from the subject is gleaned by the feature extractor, and this information is stored on the database. During authentication, biometric information is detected and compared with the database through pattern recognition techniques [4][5][6] that involve a feature extractor and a biometric matcher working in cascade.

Biometric technologies were first proposed for high security applications [6][7], but are now emerging as key elements in the development of user authentication. These technologies are expected to provide important components in regulating and monitoring access [7]. Momentous application areas include security, monitoring, database access, border control and immigration. Until now, biometric systems have been relatively expensive. In addition, they have lacked the required speed and accuracy. A family of techniques has emerged, where quality measures were used to weigh the contribution of different biometric modalities in multi-modal fusion [8]. Quality measures have also been heuristically included as meta-parameters in biometric matchers. More recently, quality measures have been interpreted as conditionally-relevant classification features and used jointly with other features to train statistical models for uni-modal and multi-modal biometric classification [9][10][11].

In this paper, we propose a novel technique for selecting biometric features for authentication purposes. The statistical property of each feature has been taken into account. The aim has been to unambiguously identifying each individual enrolled in the system while decreasing the number of features used for authentication purposes. This obviously leads to faster authentication and also an improvement in performance. Data mining technique have been used to remove unwanted impurities (noise, etc.) from the data. The variance and standard deviation of the refined data have been computed. The result shows that the statistical properties of biometric data can play an integral role in the accuracy and performance of classification. Section two provides a literature of the concept of data mining technologies and the z-score normalization technique. Section three provides the methodology of the approach used in the design of the system. Section four shows the results of the simulation and Section five provides ground for discussion and future work while section six gives an insight of the impact this research.

II. Data mining technologies

Generally, data mining means the extraction of hidden predictive information from large databases. This is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses [12]. Data mining tools predict future trends and behaviors, allowing businesses [13] to make proactive knowledge-driven decisions. They scour databases for hidden patterns, finding predictive information that experts may miss because these information lies outside their expectations. Data transformation such as normalization may improve the accuracy and efficiency of mining algorithms involving neural networks, nearest neighbor and clustering classifiers [14]. Score normalization

ISBN: 978-1-61208-144-1

refers to changing the location and scale parameters of the matching score distributions at the outputs of the individual matchers [15]. The resulting output is used to perform classification, so that the matching scores of different matchers are transformed into a common domain. The most commonly used score normalization technique is the *z-score* [17]. This calculated using the arithmetic mean and standard deviation of the given data. This scheme can be expected to perform well if prior knowledge about the average score and the score variations of the matcher is available [16]. If we do not have any prior knowledge about the nature of the matching algorithm, then we need to estimate the mean and standard deviation of the scores from a given set of matching scores. The normalized scores (S_k') are given by [17]

$$\mathbf{S_k' = (S_k - \mu)/\sigma}$$

where S_k is the raw data, μ is the arithmetic mean and σ is the standard deviation of the given data. If the input scores are not Gaussian distributed, *z*-score normalization does not retain the input distribution at the output. This is due to the fact that mean and standard deviation are the optimal location and scale parameters only for a Gaussian distribution [18].

III. Methodology

The flowchart shown in Fig.1 provides an insight of the design of the application. A hold is a distinct biometric feature from a biometric sample. The biometric data labeled 1, 2 and three consisted of different features or hold. Fig. 2 shows the histogram of the distribution of hold one which indeed follows a Gaussian distribution hence z-transform is a possible standardization for the data. It is a set of input which is similar for different individual.

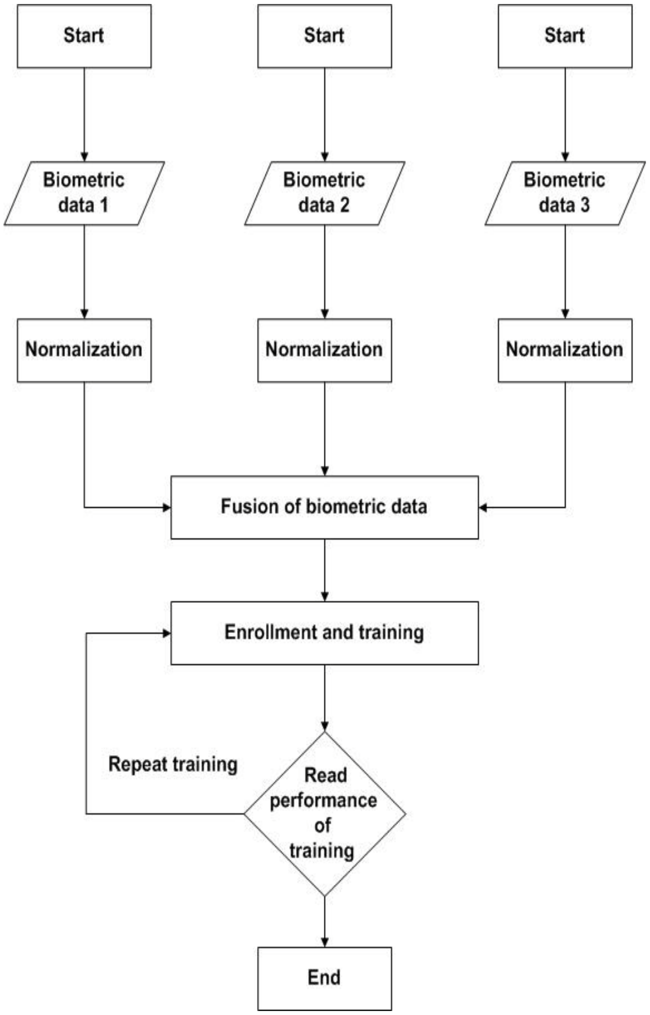


Figure 1. Flowchart showing the design of the system

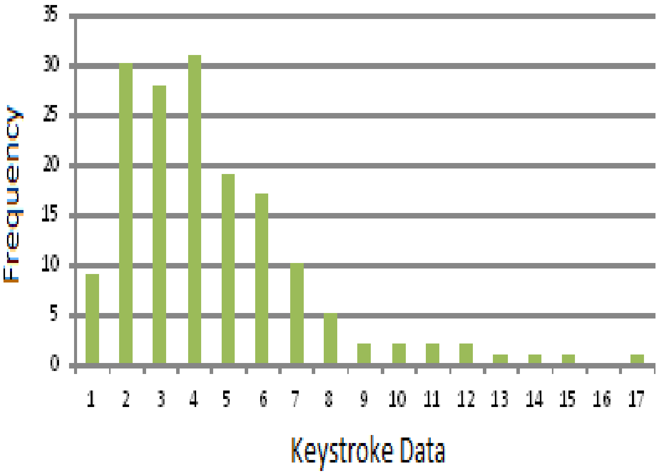


Figure 2. Plot of frequency distribution against data for Hold 1

Table 1 shows the mean and the standard deviation for each feature, in this case defined as holds. The dataset consist of nine features (hold) out of which 511 combination can be made. A subset of this amount, that is, those with greater difference in standard deviation and mean were combined for simulation purposes. This enabled a greater variance among data which decreased the correlation and thus increased the divergence from similarity for each user. This allows authentication with smaller False Acceptance rate (FAR) since there are considerable gaps between the users dataset in terms of standard deviation. It is also noticed that some features can be eliminated since they have close correlation with others and do not have a considerable impact in authentication. By eliminating these features the authentication is expected to be much faster as it makes use of fewer biometric features with more valuable information. Combining the feature vector from each biometric creates a vector that has a higher dimensionality and higher probability of uniquely identifying a person in feature space

TABLE I. MEAN AND STANDARD DEVIATION FOR EACH HOLD

Features	**Average** **(μ)**	**Standard Deviation** **(σ)**
Hold 1	9.54	2.82
Hold 2	8.34	1.83
Hold 3	8.16	1.83
Hold 4	9.07	2.82
Hold 5	10.85	4.09
Hold 6	10.04	3.01
Hold 7	9.48	3.23
Hold 8	10.43	3.36
Hold 9	8.21	1.76

IV. SIMULATIONS AND RESULTS

Simulations of the combined features were performed using neural network toolbox in MATLAB. The number of neurons, training set, and testing sets were initially chosen at random until a good and consistent result was obtained. The aforesaid parameters were eventually set to be fixed. The training algorithm used was the Levenberg Marquart algorithm and the results were computed in terms of mean square error (MSE).

Fig. 3 shows the results after the combination of two sets of data. The continuous curve is the combination with the greatest difference in standard deviation and mean while the broken curve is the combination with the smallest difference in mean and no difference in standard deviation. A horizontal line is drawn at MSE equals 0.04 to help in noting the difference between the two curves since they are closely overlapped to each other. This makes differentiation between them much easier than by mere observation of the graphs. The horizontal line is drawn as a reference to enable computation of the distinction of the two curves in a more simplified manner. The number of MSE for the red curve below the horizontal line is 50, which are about 72% of the results after simulations while the blue curve contains only 35 MSE below four representing 50% of the results both obtained upon seventy trainings

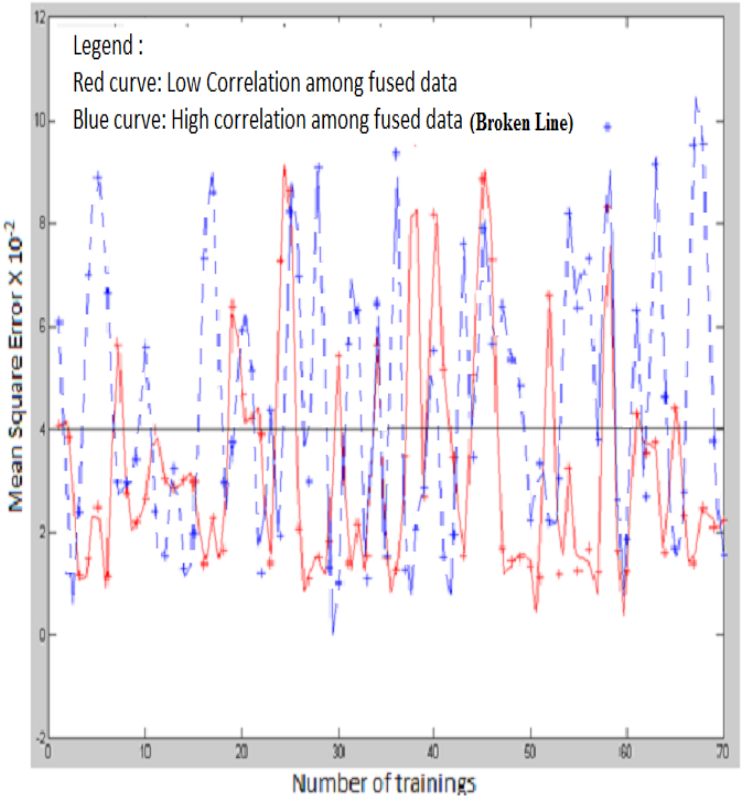


Figure 3. Performance of two sets of data combinations

Fig. 4 displays the mean square error of normalized data versus number of training for the best and worst training performances. The blue dots represents the results of the best combination whereby having greater statistical among the features. The best performance is obtained upon combining Hold 1, Hold 3 and Hold 5. This was obtained upon simulation of the combined holds. It should be noted that the correlation among these three set of data is indeed smaller compared to other combination of data. Two lines have been drawn joining the MSE for the first training and the last training. The sole purpose of the line is to show that even though the performance is continually being improved after each training, yet the combination with the lowest correlation always remained the best in terms of MSE.

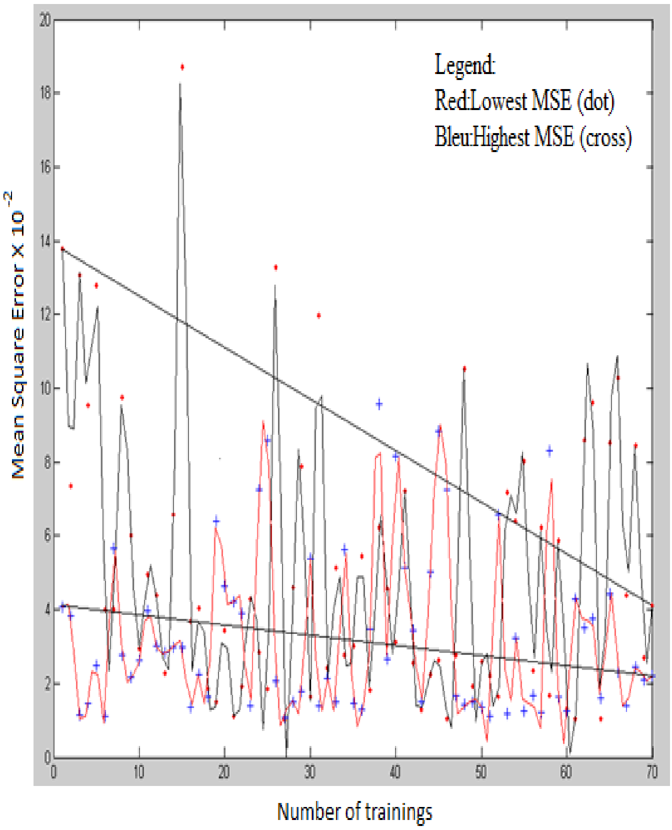


Figure 4. Mean square error of normalized data versus number of training for the best and worst training performances.

V. DISCUSSION AND FUTURE WORK

Fusion of a minimal number of biometric features in order to give better performance can be made upon a good statistical analysis of the biometric data. However, too few fused features also result in poor performance. This is because of the limitation of the variation among users thus classification becomes more error prone. For this reason there must be a balance between the statistical property and the amount of features used. The search space for the individual when performing authentication should be coherent with the number of features used as this helps distinction among individuality. While reduction of the search space remains a big challenge in biometric databases, it should not be compromised with the efficiency of the system. Minimizing the number of biometric can be regarded as a good tradeoff while keeping the search space constant. In the case of keystroke dynamics, the combination of the holds resulting in their better performance could pave way for faster authentication. It would be interesting to see the behavior of the classification process in neuroevolution of augmented topologies (NEAT). NEAT also eliminates the randomness involved in selecting the topology and weight since it enables automating the process of topology and weight optimization which is expected to give an even enhanced performance. The process of complexification from a simple topology, ensure that the final architecture is rightly suited for optimal classification process. This also results in a network that is neither too big which gives rise to over fitting nor too small, resulting in under fitting. Furthermore, selections of features were made from a single type of biometric, i.e., keystroke dynamics. It opens door for fusion of different biometrics as this will obviously result in an even more enhanced performance. The reason is due to the fact that it will create even higher divergence among the data used thus creating more uniqueness among individuals. Thus, fusion of biometrics such as iris scan and fingerprint using their statistical values can be made while keeping the number of fused features low.

VI. CONCLUSION

In this era, security has become a key element which is kept in mind when designing and developing new technologies. Biometrics has become an emergent aspect of security hereby responding to the growing need for authentication and distinction among individualities. They are gradually taking the place of traditional authentication method. Biometric authentication has become an integral part of everyday life and the trend toward a more efficient and less time consuming device is an engineer’s objective. The methodology used in this paper could inspire to build biometric systems that uses captured biometric sample and uses their statistical property to combine or remove any redundant data

ACKNOWLEDGMENT

The financial contribution of the Tertiary Education Commission is gratefully acknowledged.

REFERENCES

[1] Fernando L. Podiol and Jeffrey S. Dunn, Biometric Authentication Technology: From the Movies to Your Desktop, 2002

[2] Robby Fussell, Authentication: The Development of Biometric Access Control, The ISSA journal, July 2005.

[3] Jain LC, Intelligent Biometric Techniques in Fingerprint and Face Recognition, CRC Press, 1999.

[4] Andrijchuk, V.A. Kuritnyk, I.P Kasyanchuk, M.M Karpinski, and M.P Kasyanchuk, “Modern Algorithms and Methods of the Person Biometric Identification,” Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS 05), Sept. 2005, pp. 403 – 406.

[5] Jain A.K, “Biometrics: Proving Ground for Image and Pattern Recognition,” Image and Graphics Fourth International Conf. (IGIG07), Aug. 2007, pp. 3-3, DIO: 10.1109/ICIG.2007.195.

[6] W. Shen and T. Tan, “Automated Biometrics based person identification,” Proc. Natl. Acad. Sci. (PNAS99) , Vol. 96, pp. 11065-11066, September 1999.

[7] Simon Liu and Mark Silverman, “A Practical Guide to Biometric Security Technology,” www.lfca.net, April 2001, accessed on 20/04/2011.

[8] L. Hong, A. Jain, and S. Pankanti, “Can multibiometrics improve performance?,” Proc. of AutoID, 1999, pp. 59-64.

[9] J. Fierrez-Aguilar, Y. Chen, J. Ortega-Garcia, and A. K. Jain, “Incorporating image quality in multi-algorithm fingerprint verification,” Proc. of the ICB, Jan. 2006, pp. 213-220.

[10] J. P. Baker and D. E. Maurer, “Fusion of biometric data with quality estimates via a Bayesian belief network,” Biometric Consortium Conf. Arlington, 2005, pp. 21-22 .

[11] K. Nandakumar, Y. Chen, S. C. Dass, and A. K. Jain, “Quality-based score level fusion in multibiometric systems,” Proc. of ICPR, vol. 4, Aug. 2006, pp. 473–476.

[12] Berry M.J.A and Linoff, Data Mining Techniques: For Marketing, Sales, and Customer Support, John Wiley & Sons, 1997.

[13] http://www.thearling.com/text/dmwhite/dmwhite.htm, accessed on 20/04/2011.

[14] Han J. and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2001.

[15] Luai Al Shalabi, Zyad Shaaban, and Basel Kasasbeh, “Data Mining: A Preprocessing Engine,” Journal of Computer Science 2 (9), 2006, pp. 735-739.

[16] A.K. Jain, K. Nandakumar, and A. Ross, "Score Normalization in Multimodal Biometric Systems," Pattern Recognition, Vol. 38, No. 12, December 2005, pp. 2270-2285.

[17] F. Alsade,N.zaman,M. Z. Dawood, and S. H. A. Musavi, “Effectiveness of score normalisation in multimodal biometric fusion,” Journal of ICT,Vol 3,No 1, 2009, pp. 29-35.

A Performance Evaluation Tool for EPCIS

TaiHyun Ahn, Gihong Kim, Bonghee Hong
Department of Computer Engineering
Pusan National University
Busan 609-735, Republic of Korea
{anta77, buglist, bhhong}@pusan.ac.kr

Joonho Kwon
The Institute of Logistics Information Technology
Pusan National University
Busan 609-735, Republic of Korea
jhkwon@pusan.ac.kr

Abstract— **EPCglobal has provided a standard layered architecture built around the EPC (Electronic Product Code) for the global RFID data sharing. Among these layers, EPCIS (EPC Information Service) plays an important role for tracing and tracking individual items. However, there is no well-known solution for testing EPCIS. In this paper, we propose a performance evaluation tool for EPCIS. This tool provides fast data insertion method, rapid test data generation method, and performance evaluation method. Finally, we performed experiments to analyze our tools' test data generation speed. Through these functions, our suggested tool can generate a large number of EPCIS event data rapidly, almost 40 times faster than existing tools.**

Keywords - RFID; Middleware; Test Data Generation; EPCIS

I. Introduction

Nowadays, since the price of RFID (Radio Frequency Identification) tag drops and the size of RFID become smaller, RFID technology is becoming essential in many areas such as pharmaceutical and supply chain management [1]. In the field of supply chain management, RFID technology, replacement technology of barcode, enables us to track individual items. Therefore, an RFID system should store an enormous amount of data.

EPCglobal [2] has suggested a standard architecture for EPC (Electronic Product Code) to support the use of RFID. The standard architecture consists of several layers such as ALE (Application Level Event), Capturing Application, EPCIS (EPC Information Service), ONS (Object Name Service) and EPCDS (EPC Discovery Service) [3]. Each layer communicates with each other using a Web Service (XML Web Service).

ALE is filtering and grouping a number of raw tag data from several readers. A Capturing Application changes raw tag data to event data and sends it to the EPCIS. An EPCIS stores entire event data that contains of raw tag information and business information.

EPCIS needs high performance to store and process a large volume of data. However, there is no well-known solution for evaluating EPCIS. And deployment real system to the real application for a test is waste of time and man power. To solve this problem, in this paper, we propose a method to generate test data for evaluating the performance of EPCIS.

The remainder of this paper is organized as follows. Section 2 explains the related work of our paper and introduces previous work on RFID middleware testing. Section 3 discusses the motivation of our approach and Section 4 presents a detailed description of our performance evaluation tool for EPCIS. In Section 5, we analyze our experimental result. We conclude our work in Section 6.

II. Related Work

A. EPCIS

EPCIS is one of the standard architectures proposed by the EPCglobal [3]. Its main function is storing and searching event data. Event data consists of raw tag data and business information. The EPCIS Standard V.1.0.1 [4] proposed the basic architecture.

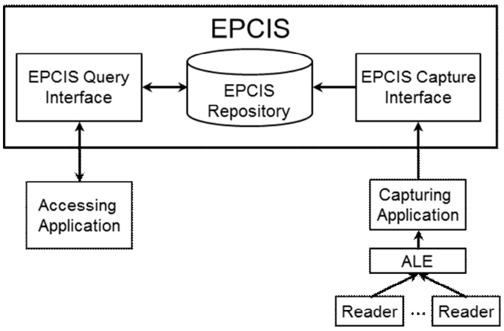


Figure 1. Simple EPCIS architecture

Figure 1 illustrates a simplified EPCIS architecture [5]. The Capturing Application receives raw EPC data from the Application Level Event and adds business information to make event data. After that, a Capturing Application sends event data to the EPCIS through the EPCIS Capture Interface. The EPC Repository changes form to store relational database such as Oracle and MySQL. When a user query from the Accessing Application is sent to the EPCIS Query Interface, the EPCIS Repository analyzes the query and changes it to the SQL and searches the data. After searching, the EPCIS Query Interface returns the result of the user query.

In the EPCIS standard data model, there are four event types: ObjectEvent, AggregationEvent, TransactionEvent and QuantityEvent [4]. Each event consists of various fields like event time, EPC list, action, bizlocation, and so on.

An ObjectEvent captures information about an event pertaining to one or more physical objects identified by EPCs. An AggregationEvent describes events that apply to objects that have been physically aggregated to one another. A QuantityEvent captures an event that takes place with respect to a specified quantity of an object class. A TransactionEvent describes the association or disassociation of physical objects to one or more business transactions.

B. Existing RFID Middleware Test Tool

Several papers [6, 7, 8] suggested test method for RFID middleware. These papers suggested that create virtual test data using virtual reader technique [6], performance test for ALE using several virtual readers and virtual accessing application [7], and ALE scalability verification to use several virtual readers and virtual path technique [8]. Figure 2 shows a simple idea of virtual reader and virtual tag. This method's main idea is to virtualize the real reader and to tag and make a real time test data for ALE.

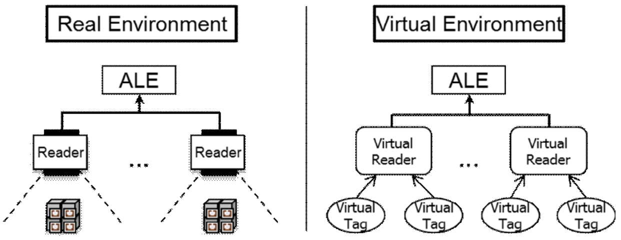


Figure 2. Example of Virtual reader

But these methods are focused on ALE. Therefore, these methods have problems for testing EPCIS. So in this paper we propose the performance evaluation tool for EPCIS.

III. PROBLEM DEFINITION

Since EPCIS handles a large amount of data, a high work load is expected in EPCIS. Figure 3 shows an example of the event occurs in a production factory. In producing place every item will be attaching a tag. Thus every item generates a one event. After producing, in packing part generates for group of items to one event. After packing, storing and transporting place also generate the same number of events.

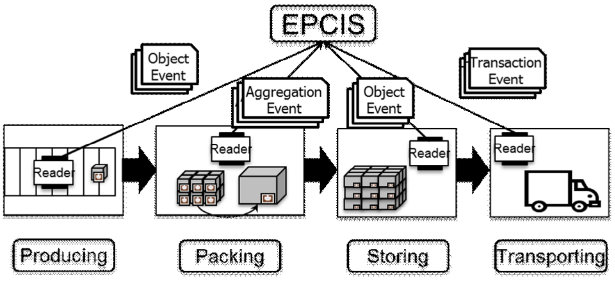


Figure 3. Example of real event generation

In this example, if a factory producing 200,000 items for a day and packed in 10 items, then the number of generated events in one month is shown as Table I.

TABLE I. EXAMPLE OF EVENT GENERATION

Business Process	Event Type	Event Number
Producing	Object Event	6,000,000
Packing	Aggregation Event	600,000
Storing	Object Event	600,000
Transporting	Transaction	600,000
Total		7,800,000

The first problem is the speed of data insertion. If test data is generated from ALE, that data must pass the Capturing Application. Since the Capturing Application adds business information to raw tag data to make EPCIS event data. But, as shown in Figure 4, each layer communicates through the web service using XML file in RFID middleware. But, marshaling operation, object conversion to XML file to be sent by the web service, and de-marshaling operation, XML file convert to object to be used in the program, are very slow operations [9]. Therefore, if a large volume of test data, such as Table I, generated from ALE will take long time to arrive at the EPCIS. And also, existing methods could generate noise data or redundant data cause of network communications.

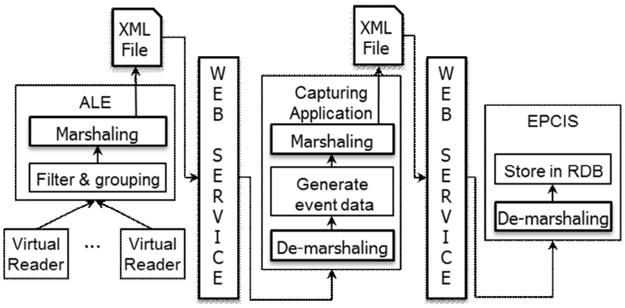


Figure 4. Example of web service communication

The second problem is the data generation latency. ALE processes only real time data. Therefore, the real environment's scenario, like Figure 3, has several seconds waiting time between product event and packing event or several hours waiting time between storing event and transporting event. Then, these testing tools have to wait same time for generating next event data since existing test tool only make a real time event for ALE. And, if the test data generating tool ignores the scenario and randomly generates event data, then meaningless data will be generated. Since that test data did not reflect the characteristics of the target environment.

To solve these problems this paper introduces a performance evaluation tool for EPCIS. This tool provides virtualization the passage of time for a high speed test data generation, high speed data insertion technique, and performance measurement function.

IV. DESIGN AND IMPLEMENTATION OF TESTING TOOL FOR EPCIS

This section proposes an architecture of the performance evaluation tool for EPCIS. Furthermore, we explain this tool's functions, parameters, and implementation.

A. Architecture

Figure 5 depicts the architecture of the performance evaluation tool for EPCIS. It consists of the configuration parser, event generator, tag generator, time controller, result generator, database connector, file connector, and EPCIS access controller module.

Configuration parser module analyzes user's input data such as tag information, event information, time information, and output information. Time controller module uses tag generator, which is instead of ALE's function in EPCglobal standard architecture, and event generator module, which is instead of capturing application, to generate business event data and tag data. Time controller module also gives time data to result generator module. Result generator module gathers the separated business, tag, and time data and generates event data for EPCIS performance evaluation.

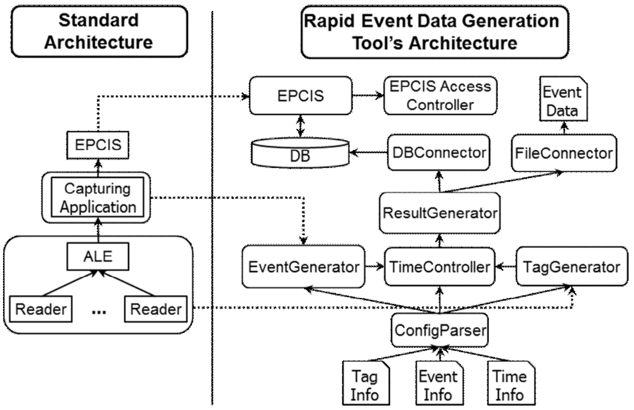


Figure 5. Architecture of the test data generation tool

Depends on users setting, generated EPCIS event data will insert database directly or store in the file using a database connector and file connector module. EPCIS accesses the controller module send user defined query to EPCIS. And if the EPCIS controller module got a reply from EPCIS then store query response time and number of stored event data to the database.

B. Function

In this section, we will explain each module's functions more detail.

1) Virtualizing passage of the time

Time controller module plays virtualizing passage of the real time function.

Existing RFID testing tools [6, 7, 8] are focused at ALE which uses only real-time data. Therefore, these testing tools are not considered scenario's waiting time of target environment. But EPCIS handles both real-time data and historical data. Therefore, test data generation tool for EPCIS has to control passage of the time to reflect test environment's scenario.

2) Tag, event data generating function

Tag generator and event generator module provide generating tag and business event data function. And result generator plays join of these two generated data function to make EPCIS event data.

EPCIS event data include both business information and raw tag data. According to this reason, unlike existing test tool, the performance evaluation tool for EPCIS has not only tag data generation function but also business event data generation function, instead of ALE and capturing application.

3) Fast Evnet data insertion techniques

Database connector and file connector module make output of our tool's result.

Insertion method of event data to EPCIS is web service communication using XML file, which is suggested by EPCglobal Standard. But web service communication's performance is too low to create a large number of event data [9]. So our test data generation tool provides direct insertion method to relational database for fast insert of event data for EPCIS. And also we provide file output method too.

4) Performance Measurement function

EPCIS access controller module plays send a user defined query to EPCIS query interface and store response time of query and number of stored event in EPCIS.

After creating the EPCIS test event data, our tool provides performance measurement method for EPCIS. EPCIS queries will define by a user using our tool's interface cause of the performance test of EPCIS is highly depended on target environment. After query definition EPCIS access controller sends generated query to EPCIS query interface. And, if the EPCIS access controller receives a result of a query sent from EPCIS, then store response time of query result and number of event, which is stored in EPCIS.

C. Parameters

Test data generation tool have to know about a real installation environment to generate meaningful test event data. Since the user should input real environment information parameters for test data generation. Parameter can be divided into three XML files.

1) Tag Information

The first parameter is tag information. Tag information contains a code type of EPC, company, item, serial number, which is the use at real environment. And also tag information includes a number of tags, generation cycle. Table II shows more detail information of tag information parameters.

TABLE II. PARAMETERS FOR TAG INFORMATION

Parameter Name	Description
Company	Code type and companyprefix
item	Item prefix
serial	Serial prefix
Max	Number of tag will generate
Delay	Delay for generation cycle
Tag	Representative one tag
TagList	Representative list of tag

Configuration parser module read this information and sends to tag generator then the tag generator makes raw tag data using this information.

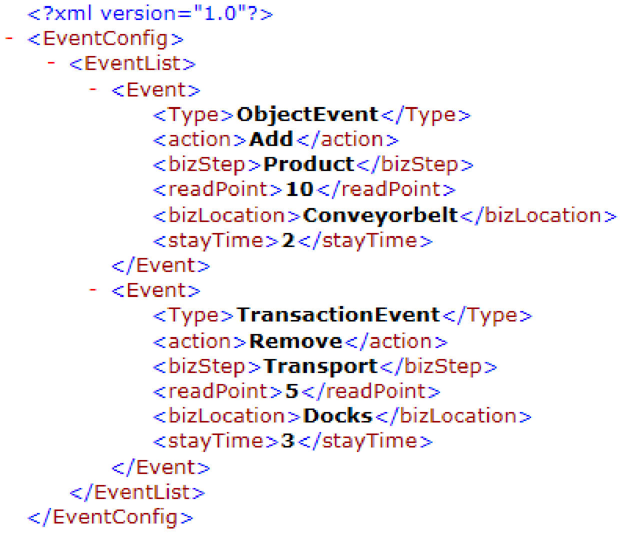
2) *Event Information*

The second parameter is event information. Event data will be completely different depend on deploying environment's features. So setting of event data is a necessity for reflecting the characteristics of the target environment's scenario. Event information parameter contains a type of event, type of action, business step, read point, business location, and the waiting time for the next event. Table III explains more detail of event information parameters.

TABLE III. PARAMETERS FOR EVENT INFORMATION

Parameter Name	Description
Type	Set event type
action	Event relates to the lifecycle of the entity being described[4]
bizStep	A vocabulary whose elements denote steps in business processes[4]
readPoint	Identify the most specific place at which an EPCIS event took place[4]
bizLocation	Designate the specific place where an object is assumed to be following an EPCIS event until it is reported to be at a different Business Location by a subsequent EPCIS event[4]
stayTime	Staytime for next event
Event	Representative one tag
EventList	Representative list of event

Configuration parser module reads event information and sends to event generator module then the event generator makes business event data using this information. Figure 6 shows an example of event parameters.



```
  <?xml version="1.0"?>
- <EventConfig>
  - <EventList>
    - <Event>
        <Type>ObjectEvent</Type>
        <action>Add</action>
        <bizStep>Product</bizStep>
        <readPoint>10</readPoint>
        <bizLocation>Conveyorbelt</bizLocation>
        <stayTime>2</stayTime>
      </Event>
    - <Event>
        <Type>TransactionEvent</Type>
        <action>Remove</action>
        <bizStep>Transport</bizStep>
        <readPoint>5</readPoint>
        <bizLocation>Docks</bizLocation>
        <stayTime>3</stayTime>
      </Event>
    </EventList>
  </EventConfig>
```

Figure 6. Example of event parameter

3) *Time information and output information*

The third part is starting time and output information. The test data generation tool will emulate passage of time, since the start time setting is required. And test data generation tool provides two parameters for store method. One is directly inserting method to the relational database, and another one is the file writing method. Table IV shows more detail information of time and output information parameters.

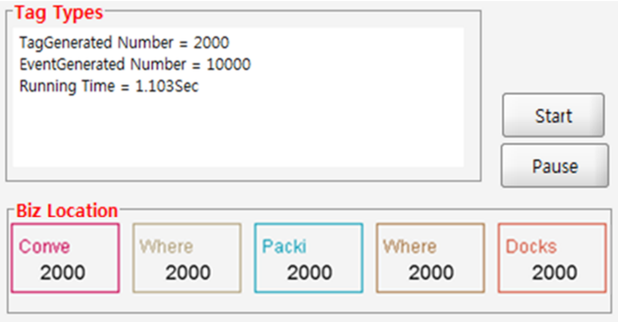
TABLE IV. PARAMETERS FOR TIME AND OUTPUT

Parameter Name	Description
Year	Year of start time
Month	Month of start time
Day	Day of start time
Minute	Minute of start time
Second	Second of start time
Unit time	Program unit time for one second
File out	Value for file out
DB insert	Value for database insert

Configuration parser module reads time and output information and sends to time controller then the time controller makes time data. After that result generator gathers the tag data, business event data and time data from the tag generator, event generator and time controller thereafter result generator generates event data.

Data generation processing is finished then the database connector, and file connector module will make an output. If a user sets the Database connector module to use then this module insert directly to the database. If a user set file connector module to use then this module write a file for EPCIS event data.

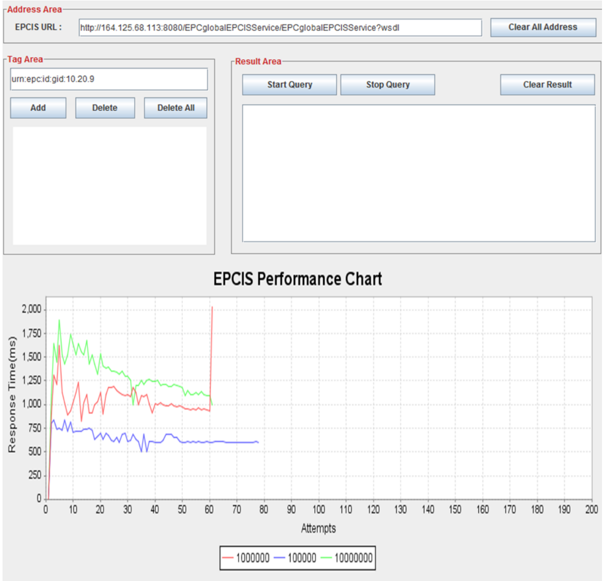
D. *Implementation of test data generation tool*



TagGenerated Number = 2000
EventGenerated Number = 10000
Running Time = 1.103Sec

Figure 7. Test data generation tool

The performance evaluation tool for EPCIS is separated two parts. The first part is an event data generation part. This part shows the process of generating event data. If user click start button then analyzes a user input and draw biz location. After that tool starts generating event data directly and show progress of generation. Figure 7 shows implemented generation part.



EPCIS URL : http://164.125.68.113:8080/EPCglobalEPCISService/EPCglobalEPCISService?wsdl

urn:epc:id:gid:10.20.9

Figure 8. Performance Evaluation tool

The second part is performance test part. Figure 8 shows performance evaluation tool. First, user sets address of EPCIS and tag information for query. After that click start query button then evaluation tool start sending query to EPCIS. If the EPCIS reply query response then recording number of stored data and query response time to the database. Performance evaluation function provides not only point query, which search just one value, but also range query. Measured information shows a graph directly to the user. This function provides to users can easily test the performance of EPCIS.

V. PERFORMANCE OF TEST DATA GENERATION

We evaluated the effectiveness of our tool. For this regard, we compared our tool with an ALE test tool [8] by checking the time of event test data generation.

Since the ALE tool cannot generate EPCIS event data, we added the Capturing Application to transform tag data into event data. Figure 9 shows the architecture for test environments.

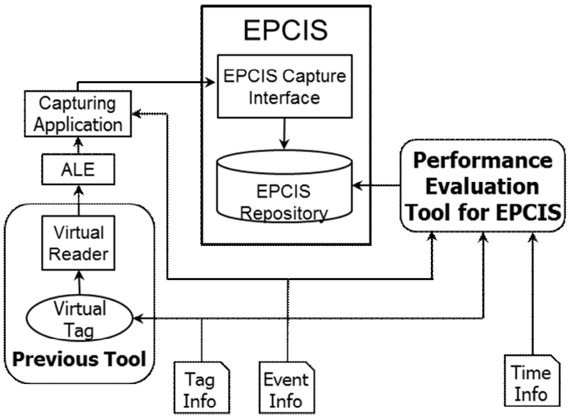


Figure 9. Test System Environment

All experiments were performed on an Intel Core 2 Quad Q6600 machine with 3.25 GB memory running Windows XP. We used an LIT-EPCIS which is the EPCglobal-certified software for EPCIS and ORACLE11g for EPCIS repository.

The parameter setting consists of one business location, one reader and one type of tag.

We have generated test event data from 1,000 to 500,000. We conducted this experiments 10 times and averaged experimental results.

Figure 10 shows the creating time for event data. Our tool shows a better performance than ALE test tool: it is 40 times faster than ALE test tool.

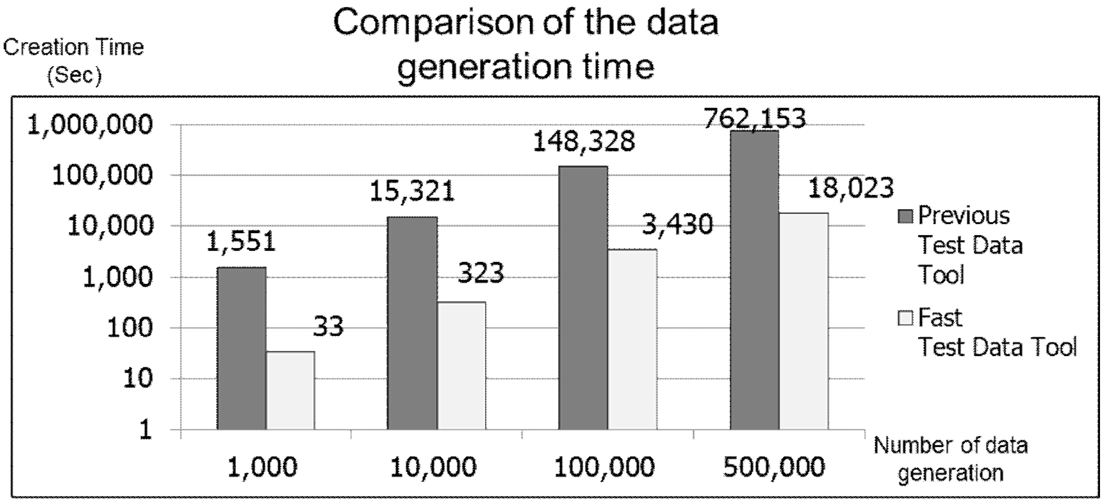


Figure 10. Result of Testing

VI. Conclusions and Future Research

EPCIS handles the large amounts of data since it can cause a performance problem. But there is no tool for testing EPCIS. Thus we propose an EPCIS performance evaluation tool. Its main functions are as follows: quick event data generation, fast insertion to EPCIS repository and performance evaluation. We have conducted an experiment to see the effectiveness of our tool. In our future work, we plan to consider the problem of optimizing RFID tags deployments and EPCIS based on our performance evaluation tool.

Acknowledgment

This work was supported by the Korean Ministry of Education, Science and Technology Grant" (The Regional Core Research Program/Institute of Logistics Information Technology)

References

[1] Chun Hee Lee and Chin Wan Chung "Efficient storage scheme and query processing for supply chain management using RFID" Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 2008, pp. 291-302.

[2] EPCglobal Inc., http://www.gs1.org/epcglobal, 05.29.2011

[3] EPCglobal Inc., "The EPCglobal Architecture Framework EPCglobal Final Version 1.4 Approved 15 December 2010"

[4] EPCglobal Inc., "EPC Information Services (EPCIS) Version 1.0.1 Specification"

[5] Himanshu Bhatt and Bill Glover, "RFID Essentials", O'Reilly, January 2006

[6] Haipeng Zhang, Wooseok Ryu, Bonghee Hong, and Chungkyu Park, "A Test Data Generation Tool for Testing RFID Middleware", in Proceedings The 40th International Conference on Computers & Industrial Engineering(CIE40), pp. 1-5.

[7] Jongyoung Lee and Naesoo Kim, "Performance test tool for RFID middleware: parameters, design, implementation, and features", Proceedings of the International Conference Advanced Communication Technology 1 (2006), pp. 149–152.

[8] Jekwan Park,Wooseok Ryu, Bonghee Hong, and Byeongsam Kim, "Design of toolkit of multiple virtual readers for scalability verification of RFID middleware", in Proceedings The Second International Conference on Emerging Databases (EDB 2010), pp. 56-59.

[9] Hazem M. El-Bakry and Nikos Mastorakis "Performance Evaluation of XML Web Services for Real-Time Applications", INTERNATIONAL JOURNAL OF COMMUNICATIONS Issue 2, Volume 3, 2009, pp. 25-33

Energy Efficient Localization Framework for Mobile Applications

Jamal A. Madni
University of California, Los Angeles
Bioengineering Department
Los Angeles, California, USA
jmadni@ucla.edu

Rahul Rao Basava
University of California, Los Angeles
Computer Science Department
Los Angeles, California, USA
brahulrao@ucla.edu

Ethan Chen
University of California, Los Angeles
Computer Science Department
Los Angeles, California, USA
ethanc@cs.ucla.edu

***Abstract*—Smart phones are increasingly becoming powerful sensor devices, which can be used to continuously sense the user's environment such as location coordinates. Numerous smartphone applications such as augmented reality and navigation requires accurate stream of location data. One of the major challenges in location sensing is energy efficiency. GPS is the primary source of location information and the continuous use of GPS has a major impact on the device's battery life. In this paper, we describe an energy efficient framework, which can provide a continuous stream of location data to the applications. We describe an adaptive duty-cycling strategy combined with a location estimation mechanism. We show that our location estimator provides reasonable accuracy while providing high-energy efficiency.**

I. INTRODUCTION

Mobile phones are powerful platforms with a wide variety of sensors, which enable the applications to be continuously aware of the user's environment. An ever growing number of applications requires continuous stream of the user's location to provide sufficient context to the information they process. A few (among the numerous) applications of continuous location sensing on mobile phones include participatory sensing [5], augmented reality [1], social networking [2], maps and navigation.

One of the main challenges in enabling continuous location sensing on mobile phones is energy efficiency. The power usage is different for different and localization primarily uses the GPS hardware which has a major impact on battery life [4]. Modern mobile operating systems such as Android provide WiFi based mechanisms to detect the location. Even though this mechanism is relatively energy efficient (compared to GPS), it sacrifices the location accuracy to a great extent and it limits its applicability. As a result, it is essential to use GPS for applications, which require higher accuracy.

In this paper, we describe a framework for energy efficient location detection for continuous sensing applications. Various mechanisms have already been mentioned in the literature for energy efficient localization [6, 7, 8]. Inspired from these mechanisms, our framework is implemented based on the following principles: 1) Reduce the use of GPS by using a combination less power hungry sensors (whenever possible) to detect and estimate location data, 2) Detect user movement and turn off GPS when the user is idle 3) Duty-cycle GPS and other sensors used in the framework.

One of the problems with duty-cycling is the impact on data accuracy. If the application requires location data between two duty-cycles, it will only receive the GPS coordinate from the previous duty-cycle. If the duty-cycle interval is large, the user can move a significant distance between two duty-cycles resulting in reduced accuracy. This accuracy can be improved by reducing the interval but only at the cost of energy efficiency. In this paper, we also describe a scheme to improve the accuracy without reducing the duty-cycle interval by using location estimation. We describe the design, implementation and evaluation of the of the energy efficient localization framework containing the location estimator.

Various mechanisms for location prediction have already been proposed. For example, [6] describes a mechanism called EnLoc, which used human mobility patterns as a heuristic to estimate the user location. In our scheme, we estimate the speed using the information from the previous duty-cycle and current movement pattern (which is calculated using the accelerometer). Our estimation scheme is based on user speed prediction using accelerometer samples. We propose the use of a caching mechanism, which dynamically learns the correlation between the value calculated from the accelerometer and the actual speed of the user.

Our implementation is based on an assumption that the user will continue to move in a straight line between two consecutive duty-cycles. Even with this assumption we show that our framework can provide reasonable data accuracy and high energy efficiency.

This paper is organized as follows. Section 2 describes the design details of the energy efficient localization framework and section 3 talks about the location estimator component of the framework, Section 4 mentions a few important details from the implementation and section 5 details the experimental results followed by the conclusion in section 6.

II. DESIGN

Our group chose to design a framework to provide location data to possible clients. The philosophy followed largely stems from the design philosophy used in sockets, in which the multiple potential streams and sources of sensor data used to provide a single location are abstracted away, presenting a single stream of

ISBN: 978-1-61208-144-1

location data to client programs. We divide up our sensors into modules roughly correlating with our three localization schemes. We break up our modules into a GPS based scheme, a WiFi based scheme, and a cache and WiFi based scheme. Our three schemes have differing power consumption figures as well as accuracy, due to the sensors and algorithms used within. This allows the framework to switch between the various schemes for better power saving, increased accuracy, and redundancy. The schemes used are named for the primary sensor used to obtain location. From these three schemes, the framework itself caches the last several location samples, for use in the last significant module in the system, the location estimator module.

The framework provides location data to clients, but the rate it polls its own modules is independent of the rate at which the clients require location data. Additionally, since the framework must allow module polling to be adjusted based on power concerns as well as redundancy concerns, the framework must incorporate an estimation module to fill in data at time points in between module samples. The estimation module is responsible for providing location data between module location fixes, as well as smoothing out location data based on location values cached by the framework itself. It uses location data coupled with bearing and velocity to return a predicted location.

The framework connects all of the aforementioned modules, decides which location module to use based on the set of sensors available for use, the remaining amount of system power, and the power savings settings specified by the user. If the client polls the framework at such a time far enough from a location module poll, the estimator module is used to provide a best guess location instead. As estimation algorithms go, accuracy degrades very severely very quickly, so the framework must rely on the next location module poll interval to lock location to an “accurate” value.

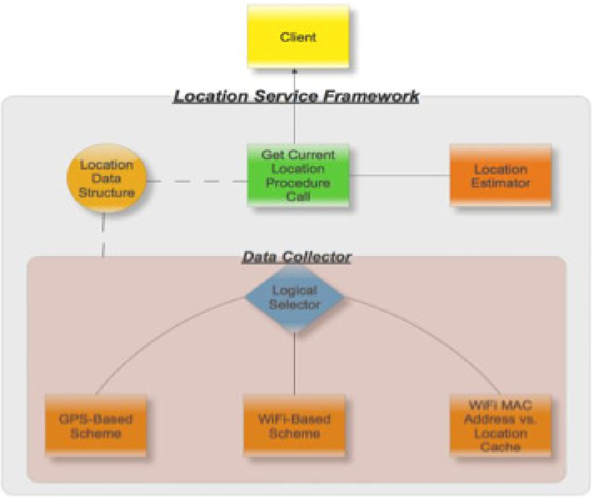


Figure 1. Overview of Design

The design lends itself to the stated goal of power efficiency by allowing the system to modify the set of sensors currently in use for a more power efficient set, but to adjust the rate at which it samples those sensors through the use of an estimation module.

A. GPS

The GPS module, so named because the primary source of location is the GPS, relies on sensing idle time to reduce its power consumption to the minimal it can. This is significant as the GPS sensor’s power usage is one of the highest, as illustrated in this chart. We bring GPS power usage in this scheme down by using the 802.11 WiFi and accelerometer sensors to detect idleness, instead reporting back the last known GPS location, only using GPS for location when under periods of motion and location change.

While Figure 1 shows that one of the few sensors that does in fact consume just as much power as the GPS is the 802.11 WiFi chip, Figure 1’s data shows power consumption for WiFi in a data transfer state. Our scheme uses WiFi only to sample the MAC addresses of the base stations broadcasting within range. These values are cached in the GPS module, to be used later as a fingerprint identifying the particular area it was sampled in. Since base stations may or may not be visible in repeated scans of the same area, the cache includes a similarity comparator, which will match area fingerprints provided the base station MAC addresses seen are within some percentage of similarity of each other. By comparing a scan’s fingerprint against the cache, we can establish whether or not the current location is new. This allows the GPS sensor to establish idleness with relatively good power consumption.

While 802.11 WiFi can be used to establish whether the current location has changed, in order to ascertain whether the system is in motion, the accelerometer is used. When WiFi scans reveal an unknown location fingerprint, the accelerometer is polled for a movement factor. This movement factor will be discussed in more detail later, but for now it is sufficient to say that the movement factor is a sum value of acceleration in 3D space. The GPS module uses this movement factor to decide whether the system is in motion, and from there whether to poll the GPS sensor for a more accurate fix.

By taking advantage of the usage profile of mobile phones, which we believe tends towards long stationary periods punctuated by short periods of movement, the GPS module designed has the potential to reduce power consumption by a significant amount. However, of the three sensor modules designed, the GPS module is still the most power hungry. On the scale of power consumption versus accuracy, the GPS module still tends towards accuracy.

B. WiFi

The WiFi module is named for the fact it acquires location by polling for WiFi signatures and comparing against a 3rd party online WiFi location database. There are several of these services available, including Skyhook [3] and Google's own WiFi location implementation available on its Android platform. This module uses the WiFi location sources in an analogous manner to the GPS sensor. As noted before in Figure 1, the WiFi chip's power consumption is high, but the assumption again holds that this is for data transfer. The opportunity for power savings here comes from the fact that the WiFi location is based around signal strength and base station MAC addresses. By avoiding constant trips to the location database when idle, power can be saved. However, we do not expect this approach to yield as significant savings as the scheme that uses GPS due to the need to periodically use the WiFi to scan the current location.

As noted before, the WiFi module is very similar to the GPS module in terms of design. They share the same general algorithm of polling the location source under movement while scanning using the WiFi and accelerometer for fingerprints and movement factors, respectively. The WiFi scans provide a test for idleness, while the movement factor obtained from the accelerometer is used in the decision to poll the main location sensor. Power usage is primarily minimized by the scan for idleness.

Since the WiFi module is so similar to the GPS module, its intended use case is that of a backup to the GPS module. Should the GPS be unavailable due to a physical lack of GPS hardware, user GPS use restriction, lack of GPS signal, or even GPS sensor damage, the WiFi module can be used as a replacement. Obviously the WiFi module's accuracy falls compared to GPS, as systems such as Skyhook report accuracy of approximately 10-30 meters [3]. Additionally, we anticipate that a comparison of energy efficiency against the GPS module will result in modest gains at best. Again, this is why we have designed the WiFi module from the outset as a backup or replacement for the GPS module.

C. WiFi & Cache

This module is named for the fact that it gets location based on WiFi fingerprints already present in the cache. As mentioned before, both the GPS and WiFi modules incorporate a cache of WiFi location fingerprints. The idea behind this module is to use the cache of location fingerprints directly to look up locations. The accelerometer and compass is then used to provide a bearing for use by the estimator module. In essence, this module is in many ways a micro-implementation of a WiFi location scheme, such as Skyhook. Because this scheme relies on local sensors only, it is the most limited and inaccurate, and it is only used in the case that GPS and WiFi location have failed, have become too power- intensive, or have become user-restricted.

The reason we believe this scheme can be functional again lies with user usage profiles. Typical mobile phone usage tends towards repeat visits to a similar set of locations. From this observation we feel it is likely that cached locations will be visited again, and an estimated location can be generated from previously observed patterns. Additionally, as it relies most heavily on a lookup of data, power usage is minimal. However, it again is worth noting that this scheme is has many more drawbacks compared to GPS and WiFi location, including potentially poor accuracy, fragility, and failure to return data, and as such is intended for use in situations where any other localization modules are not available, not as a primary location source.

III. LOCATION ESTIMATION

The location estimation method is a threshold- based scheme between GPS duty cycles. Conceptually, our group partitioned GPS duty cycles into equal length time instances. A T' time threshold was then determined in an adaptive manner. Here T' was defined as the elapsed time from when our framework polled GPS to gather the exact location and the current time. If a client polled our framework with a get current location event inside of our T' value, our estimator would simply return the prior GPS duty cycle GPS sample, with the justification that the prior sample is still "fresh" enough to maintain relative accuracy. However, if a client polled our framework outside the T' value, then our location estimator method would be invoked.

Assuming the time threshold was set to T1, the location returned to satisfy the request of event E1 would simply be the location value when actually polling GPS at C1. However, the location estimator would be invoked for event E2 and return an approximate location based on the GPS sampled location at C1 as the method's baseline location. It is important to note that the location estimator method is an iterative process and thus has a cumulative error characteristic. For instance, if a get current location event E2* occurs between time instances T2 and T3, the location estimator will base its approximation computation for satisfying the event by using the location at the last satisfied event, namely the location returned at E1, as the baseline. Similarly, if a get current location event E2** occurs between T2 and C3, the estimator will satisfy this request by using the location generated for E2* as its baseline.

It is also important to note that these characteristics are then reset at the next GPS duty cycle. Thus, event E3 is satisfied in a similar manner as event E1, now using the GPS location at the next duty cycle, C2. Furthermore, based on this new GPS polled value, events E4, E4* and E4** will be dealt with in the same manner as E2, E2* and E2** respectively in the prior GPS duty cycle.

The algorithm for the location estimation scheme begins by calculating the system time and comparing this Delta T value with the previously defined T' threshold. If the Delta T is less than or equal to the T' threshold value, this means that the get current location event request lies within our threshold parameter and the previously sampled GPS value can simply be returned as it is "fresh" enough. If, however, the Delta T value is greater than the T' threshold value, then the event request is outside this freshness boundary and an approximation must be done on location to satisfy the request. Thus, the scheme enters an accelerometer module whereby the sum of the net acceleration of the mobile device is calculated and averaged over N very closely timed samples. Here N is again adaptive and the extremely minute time instances between samples are for the purpose of simulating a continuous time interval of the mobile phone's movement behavior. This behavior was qualified as the Current Movement Factor of the phone and is defined as the "movement state" of the phone.

If this Current Movement Factor is zero, we assume that velocity is also zero and thus we return the current location stored (either the location from the previous GPS sample or an approximate location generated for a prior event). If the Current Movement Factor is equal to the Previously Recorded Movement Factor, then we use the corresponding speed of the previously recorded movement factor as an input to our formula estimator. If non-equality holds between the above two movement factors, then the scheme enters a Movement Factor vs. Speed Hash Map, which is described in greater detail below. Upon entry into the Hash Map, if a corresponding speed exists for the Current Movement Factor, we use this speed as an input to the formula estimator, however, if no such corresponding speed exists, we simply use the previously recorded speed (which again corresponds to the speed of the previously recorded movement factor) as our input. Once the scheme enters the formula estimator, a new location is generated based on the movement factor, speed, bearing, time and location stored. The variable values are then updated and the scheme is then prepared for subsequent event requests.

The formula estimator calculates a new location based on the Haversine formula, which is a well-known trigonometric and iterative equation for navigation. Our version of the Haversine formula calculates a new location based on: 1.) the current location in our system, 2.) the distance determined as a product of the elapsed time from the derived location of the prior event request and the probable speed (derived from the Movement Factor vs. Speed Hash Map), and 3.) the bearing, which is assumed to be constant since the last GPS sampled location.

The Movement Factor vs. Speed Hash Map is a necessary data structure since directly calculating velocity as the integration of acceleration is exceptionally error- prone due to shaky movements. Thus the purpose of the Hash Map is to maintain a history of Movement Factor vs. Speed correlations for GPS samples collected during past duty cycles. The map tries to match the Current Movement Factor with a cache of speeds. One of the main problems with this mechanism is that there is a many-to-one correspondence between speed and movement factor, and thus the movement factor gets frequently overwritten. For instance, if an individual is walking 2 m/s while keeping the mobile device extremely stable in his/her hand, and then enters a car that will travel 40 m/s (but continues to keep the mobile device in exactly the same stable manner in his/her hand as in the walking case), then the movement factor has two corresponding velocities. Similarly, this property can be extrapolated to many velocities. Our system deals with this by autocorrecting the movement factor at every new GPS sample, and thus the Hash Map serves as an intelligent mechanism that attempts to learn, over time, the relationship between Movement Factor and Speed.

IV. IMPLEMENTATION DETAILS

In this section, we describe a few important aspects of the framework implementation.

As mentioned before, the GPS hardware is duty-cycled in our framework to reduce energy consumption. We use an adaptive duty-cycling scheme where the interval between two duty-cycles is dynamically calculated using the user's speed. This is done by making the distance traveled between two duty-cycles constant. In our current implementation, this value is defined as 100m and the duty-cycle interval is calculated by dividing this value with the current speed. The problem with this approach is that the user's speed could increase in large amount within the same duty-cycle interval, which in turn reduces the accuracy. But the location estimator mitigates this problem by estimating the locations in between regardless of the speed variations.

In order to detect the movement factor, the accelerometer is used, which is also duty-cycled at a constant rate. In our current implementation, this interval has been chosen to be 10s. At every duty-cycle, a few accelerometer samples are collected and the net acceleration on each sample is calculated. These net accelerations are averaged to get the movement factor.

In order to detect movement when the user is idle, WiFi signatures are continuously scanned to check if they have been changed. This is also done at a constant rate every 10 seconds. We also discard weak signals for better stability.

V. EXPERIMENTAL RESULTS

The evaluation was done on a Motorola Droid X using ANDROID Platform- 2.2. The evaluations were done to measure two factors, battery consumption and accuracy.

Accuracy is measured as the deviation from GPS coordinates as obtained from the sensor directly. It should be noted that in some cases the GPS coordinates directly obtained from the sensor may have significant deviation from the ground truth. Thus we are only measuring the deviation of the values as estimated by the framework with values as given by GPS sensor. We are not comparing the values with real path taken by the subject. For accuracy measurements we poll both the framework and the GPS sensor at the same time at a fixed rate. This information is recorded as pairs of locations obtained at the same time. Since the GPS sensor is continuously used, there is a significant drain on the battery. In the evaluations below, the experiments for battery consumption and accuracy measurement have been kept separate for this reason.

Accuracy was evaluated for two cases. The first is a set of measurements gathered when the subject is walking or running. The user will walk interspersed with random amounts of time spent in running or waiting at stop signs or road crossings. The second is a set of measurements gathered when the subject is driving.

A. *Test Scenario 1*

This scenario involved walking short distance with near constant speed. The subject does not run or stop at any time when the sample was collected. The walk was in a near straight line. The locations were sampled every 5 seconds. Figure 7 shows a plot of GPS coordinates (in yellow) and the locations returned by the framework (in blue).

The locations are marked with numbered symbols. The numbers indicate from which pair the point was plotted on the map. Some of points returned by the framework will not be visible since they coincide with the GPS locations and thus are rendered under the symbol for the GPS locations. This will be observed in all the test scenarios for accuracy measurement.

B. *Test Scenario 2 & 3*

These scenarios involve walking long distances with variable speed. The subject occasionally runs and stops at road crossings. The walk was in a near straight line. The locations were sampled every 5 seconds. Figure 8 and 9 shows a plot of GPS coordinates (in yellow) and the locations returned by the framework (in blue) for each of the scenarios respectively.

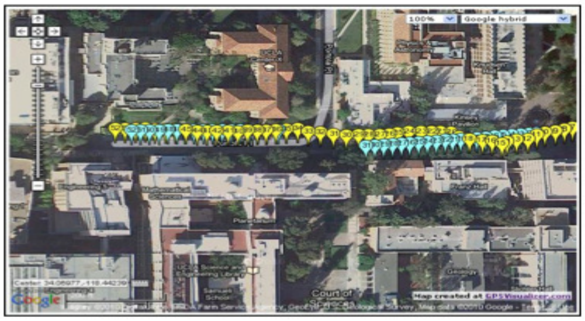


Figure 2. Test Scenario 1

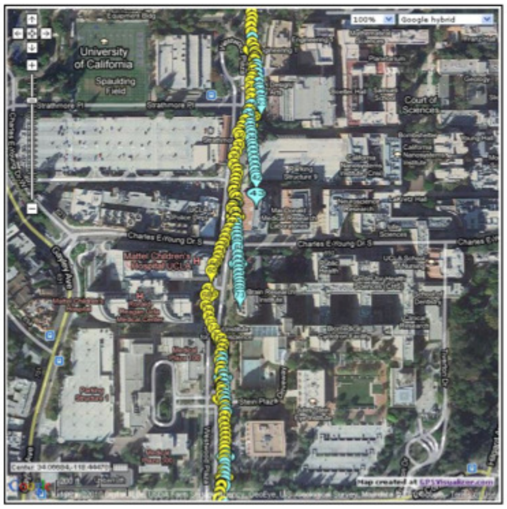


Figure 3. Test Scenario 2

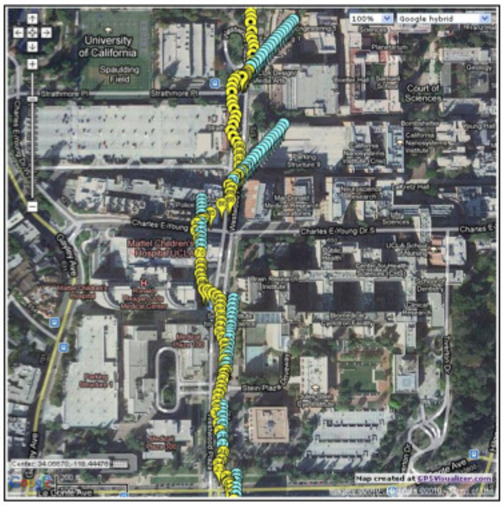


Figure 4. Test Scenario 3

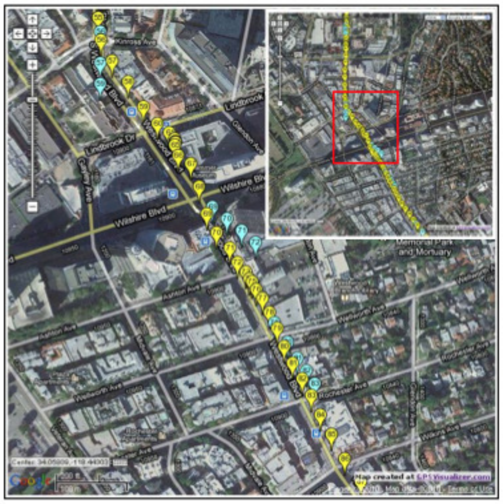


Figure 5. Test Scenario 4

C. Test Scenario 4

This scenario involved driving with variable speed. The subject accelerates at random times and stops at signals. The drive route was a near straight line except for one major change in direction. The locations were sampled every 3 seconds. The figure 10 shows a plot of GPS coordinates (in yellow) and the locations returned by the framework (in blue). The figure 10 (inset) shows the how the majority of the points are plotted. The figure 10 (main) shows a specific part of the plot where a change in direction occurred. The average speed was 6.146 m/s and the variance in speed was 29.22 m/s. These are measured using the speed information returned by the GPS location API.

D. Discussion

In our implementation the bearing of the subject was assumed constant and the results of the test scenarios reflects the same. As can be observed from the figures, the blue symbols, when not aligned with the yellow symbols, are at tangents to the path formed by the yellow symbols. This is directly due to the fact that the bearing is assumed at the point of divergence and that point onwards estimated points occur in a straight line whereas the GPS location changed direction. It can be observed that the errors are cumulative in nature and errors in previous estimations add to the current estimation until the framework synchronizes back with the GPS locations from the sensor. This can be markedly observed in Test Scenario 3 where the tangents are significantly off the mark. However in the case of driving (Test Scenario 4), the tangents are not so marked. This could be attributed to the adaptive cycling used for GPS sampling (which reduces the sampling interval for higher speeds) thus leading to a faster synchronization of the framework to the actual GPS coordinates.

E. Power Measurement

The power consumption measurements are obtained by running separate runs for a scripted route. The drop in battery power for the new framework at various instances of time with respect to the reference start point of measurement is recorded. Similarly drop in battery power for a direct GPS polling at various instances of time with respect to the reference start point of measurement is recorded. The interval for which the battery API in Android returns a change event is phone dependent. The Droid X showed only 10% battery changes.

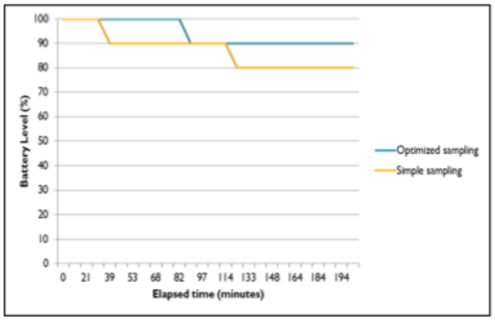


Figure 6. Battery Usage

Figure 11 shows the battery consumption for sampling using our framework in comparison to sampling using simple GPS sensor polling at a 30 second interval. The framework was sampled every 5 seconds. These measurements were done for over a 3 hour duration. The scripted scenario included walking and periods of no movement. Clearly it can be seen that the framework gives much better battery usage. However, increasing test time can provide much better results. Also we have to evaluate the extent to which this gain is due to entering a sleep state when the device is not in motion.

VI. CONCLUSION

The experimental results show that reasonable accuracy can be obtained by using location estimation mechanism even with the assumption that the user must only move in a straight line. One drawback of our approach is that successive estimations within the same duty-cycle will result in compounded error. Thus, if the application does not poll the location a large number of times, reasonable accuracy can be guaranteed.

We also show that adaptive duty-cycling combined with user idle detection can save a great deal of energy. This energy saving is directly proportional to the idle time of the user. With mechanisms such as WiFi signature detection in place, quick user movements could be detected so that GPS sampling can be restarted before the user covers a large distance.

We would like to acknowledge that a practical application of this mechanism would require detection of bearing changes between two different duty-cycles. This can be done using the compass and the orientation sensor and the investigation of this mechanism is left to our future work.

REFERENCES

[1] http://www.layar.com/

[2] http://cenceme.org/

[3] http://www.skyhookwireless.com/

[4] F.B Abdesslem, A. Phillips, T. Henderson, “Less is more: energy-efficient mobile sensing with senseless”, MobiHeld, 2009.

[5] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, M.B. Srivastava, “Participatory sensing”, SenSys, 2006.

[6] I. Constandache, S. Gaonkar, M. Sayler, R.R. Choudhury, L. Cox, “EnLoc: energy-efficient localization for mobile phones”, Infocom, 2009.

[7] M.B. Kjaergaard, J. Langdal, T. Godsk, T. Toftkjaer, “EnTracked: energy-ecient robust position tracking for mobile devices”, MobiSys, 2009.

[8] Y. Wang, J. Lin, M.Annavaram, Q. A. Jacobsen, J. Hong, B. Krishnamachari, N. Sadeh, “A framework of energy efficient mobile sensing for automatic user state recognition”, MobiSys, 2009.

Architecture for Extreme Low Power Sensing in Wireless Sensor Network Devices

Jerker Delsing, John Borg, Jonny Johansson

Luleå University of Technology
EISLAB
Luleå, Sweden
jerker.delsing@ltu.se, johan.borg@ltu.se, jonny.johansson@ltu.se

***Abstract*—When discussing powering wireless sensor network nodes, there are a few major energy consumers: communications, microcontroller and the sensor. We propose a wireless sensor network platform architecture minimizing the energy consumption of sensing. The architecture proposed herein is based on a reactive approach to sensing. A number of possible hardware approaches are evaluated and compared. This comparison indicates that analog storage between the sensing element and the sensor electronics can be a feasible method for reducing the energy consumption of the system.**

Keywords-low power WSN sensing; WSN node architecture; wireless sensor network node.

I. Introduction

One of the most common questions regarding wireless sensor networks, WSN, is what the power consumption at the sensor node must be. Much work has been done on low-powered sensor nodes and their communication abilities: see, for example, [1]–[7]. Some specific examples are schemes handling the reduction of communication [8], effective routing and multihop schemes, and the reactive partial waking up of WSN nodes [9].

Most often, the sensing element itself is disregarded from an energy budget point of view. The current state of the art for sensor interfacing is to convert the sensor data to digital form. The most common approach is using an A/D converter. Other well known approaches involves letting the sensor data influence a digital pulse train of which we easily can measure parameters like frequency, pulse width or duty cycle. None of these approaches puts the power consumption of the sensor into focus. All of these approaches are based on the assumption that all data should be transfered to some computational stage on the WSN node or any device higher up in the system architecture.

In this paper, we propose an architecture for low power interfacing of a sensing element to a WSN node. The architecture exploits the idea of detecting no or discardable changes in data. Such detection should then inhibit further processing of sensor data as early as possible. The proposed architecture is based on a reactive wake-up chain starting at the sensing element itself.

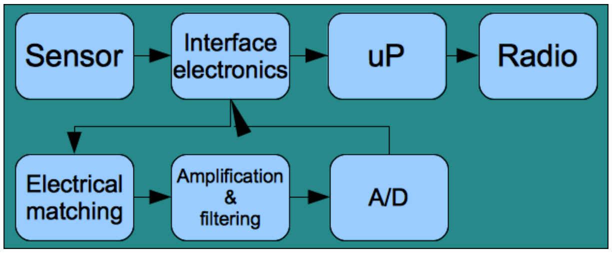


Figure 1. Traditional WSN node architecture, A1

A. WSN node architecture

The basis for this WSN node architecture is the determination of changes in sensing element data as early as possible. In many real systems data changes are small and are most often not of interest to the surrounding system. From an energy consumption point of view, we like to keep as much of the WSN node asleep as possible when determining whether a data sample constitutes a significant change compared with the previous sample.

Consider a WSN node architecture as in Figure 1. The most frequently used approach is to read the data into the μP store and compare it with the previous data. Energy is spent on sensing element sampling, signal amplification and filtering, A/D conversion and data comparison in the μP.

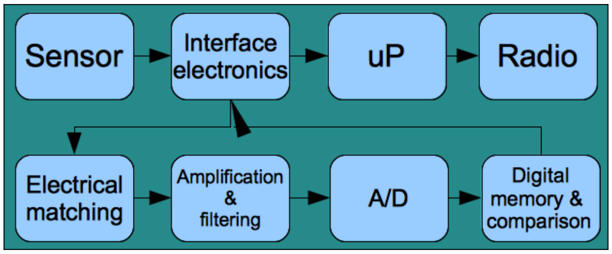


Figure 2. WSN node architecture, A2, with external digital sensor memory and comparison logic.

The next opportunity for comparison is by adding a digital memory and comparison function after the A/D conversion, as shown in Figure 2. Energy is spent on sensor sampling, signal amplification and filtering, A/D conversion and storage and comparison of the data in digital memory. Enabling the μP to sleep while performing sensor data sampling if there are no changes or small changes in sensor data will use energy only for the HW cost of digital memory, some configurable logic (to allow for the setting of change limits) and sending a wake-up signal to the μP.

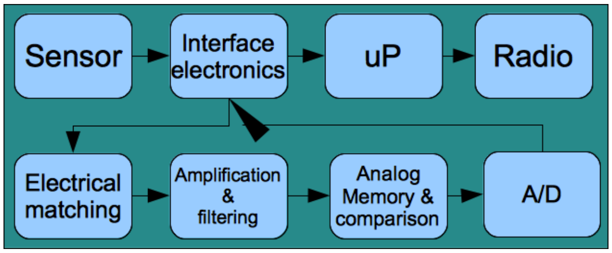


Figure 3. WSN node architecture, A3, with external analog memory and comparison logic before the A/D conversion.

In terms of energy cost here, the A/D conversion is the most power-hungry process. Can we bypass A/D conversion as well? A possibility for this is given in Figure 3. Here, we introduce analog memory with the capability of storing and comparing at least two values with associated logic and signaling to the μP. HW developments supporting this architecture using charge-coupled device technology, or ccd, were made available in the seventies [10]. Recently, new findings presented by Borg and Johansson [11] indicate that ccd technology implemented in standard CMOS processes exhibits properties supporting the proposed architecture. Energy is spent on sensor sampling, signal amplification and filtering and analog storage (ccd) and associated logic. Thus, A/D conversion and the μP can sleep during sensing element data sampling if there are no changes or small changes in the sensed data.

II. Energy analysis of the architecture

To analyze the energy consumption of the presented architectures, we will use two different sensing situations:

- Temperature sensing using a PT100 element as the sensing element
- Ultrasound pulse echo measurement using piezo ceramic transducers as sensing elements

This analysis is conducted using published state-of-the-art data (from our group and others) and data from commercial devices. In this way, we can build an accurate picture of the total energy consumption of the proposed architectures. No circuit simulations are made here, nor have we built any complete devices.

The energy analysis is based on the following model. The total electrical power P_{tot} consumed by a WSN node can be described as:

$$P_{tot} = P_{sens} + P_{cond} + P_{A-mem} + P_{AD} + P_{D-mem} + P_{\mu P} \quad (1)$$

The total energy usage is then obtained by introducing the time needed for each operation (after which it can be turned off). To make the analysis reasonably simple, we assume that the architecture supports turning off E_{sens}, E_{cond}, once data has been stored either in analog or digital form.

For data sampling from one sensor, we assume a sensing and conditioning time $t_{sens_{c}ond}$, an analog storing time t_{A-mem}, an A/D time t_{AD}, a digital memory time t_{D-mem} and a μP time $t_{\mu P}$. Thus, we obtain the total power E_{tot} used for data sampling from one sensor as:

$$E_{tot} = P_{sens} * t_{sens} + P_{cond} * t_{cond} + P_{A-mem} * t_{A-mem} + P_{AD} * t_{AD} + P_{D-mem} * t_{D-mem} + P_{\mu P} * t_{\mu P} \quad (2)$$

Provided that we have some understanding of the real values of these energies and times, we can calculate the total power consumption. In the following equation, we will do this for two sensor types, a PT-100 sensor and an ultrasonic pulse echo sensor, for each of the three architecture types A1, (see Figure 1), A2 (see Figure 2), and A3 (see Figure 3).

A. Energy analysis PT-100 sensor

In this case, we assume the power consumption and time needed for a PT-100 sensor and its associated electronics, according to table I. For each of the three different architectures in Figures 1-3, we then calculate the expected span of energy consumption (see Figure 4).

Device	Energy consumption [mW]	Time awake [μs]
PT100	0.1-1 [12], [13]	10
Conditioning electronics	0.01	10
Analog memory	0.1-50 [10], [14], [15]	1
A/D (10-14 bit)	0.05 - 2 [16], [17]	1
Digital memory	0.01-0.1	1(storage time)
μP	1-10 [18]	30

Table I
Energy consumption and timing for a PT100 WSN sensor node;, observe the logarithmic scale on the Y axis.

B. Energy analysis ultrasound sensor

Here, we discuss the power consumption and time needed for a piezo electric transducer in an ultrasound pulse echo system. A typical pulse echo measuring situation with typical sound signals is shown in Figure 5. In table II, the power consumption and related timing for the sensor and associated electronics areis given. For each of the three different architectures A1-A3 of Figure 1-3, we then calculate the expected span of (maximum and minimum) energy consumption (see Figure 6).

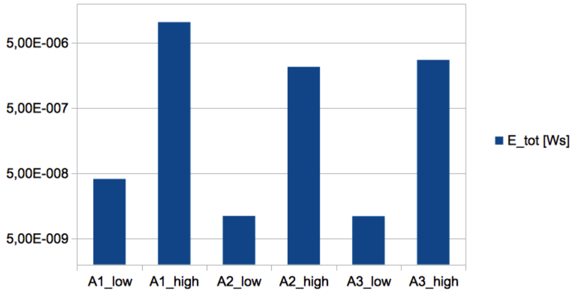


Figure 4. Energy usage for different WSN node architectures and a PT100 temperature WSN node.

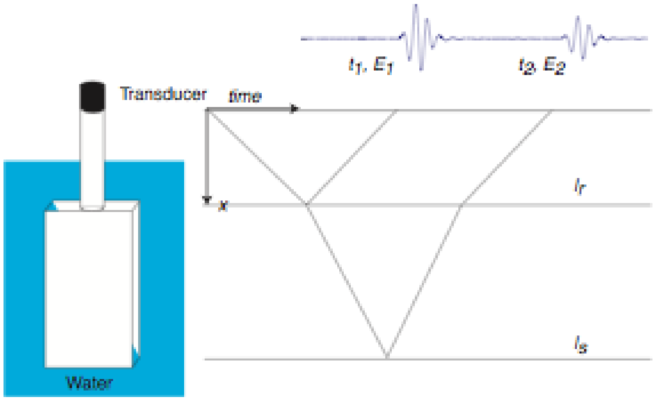


Figure 5. Ultrasound pulse echo measurement with associated acoustic signals

Device	Energy consumption [mW]	Time awake [μs]
Piezo excitation	0.01 [19]	1 (1 μs long pulse excitation
Amplifier and filtering	1-100 [20]–[22]	10-20 (signal duration + startup time)
Analog memory	0.1-100	10-15 (signal duration + startup time)
A/D	0.1-10	10-15 (sampling + startup time)
Digital memory	0.01-0.1	10 (storage time)
μP	1-10	30 (300 clock cycles at 10MHz)

Table II
ENERGY CONSUMPTION AND TIMING FOR AN ULTRASOUND WSN SENSOR NODE;, OBSERVE THE LOGARITHMIC SCALE ON THE Y AXIS.

III. RESULTS AND DISCUSSION

Under the assumptions that we made, we have compiled energy consumption data for two sensing scenarios. These data are shown in Figures 4 and 6. It is obvious that the μP uses a large amount of energy. Thus, any architecture that can avoid waking up the μP has clear advantages from an energy consumption point of view.

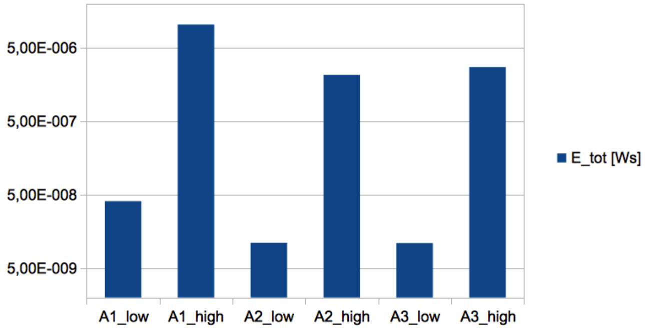


Figure 6. Energy usage for different WSN node architectures for a piezo electric based ultrasound pulse echo WSN node.

It is also clear that in a sensing situation with a dynamic sensor signal, such as ultrasound, avoiding A/D conversion is a promising approach. If analog memory and comparison techniques can be developed similar to what we have seen for A/D converters, analog storage architecture will be a strong contender forin future WSN designs.

IV. CONCLUSION

The reactive architecture proposed here for minimal energy consumption of sensing on WSN platforms is promising. Based on an analysis of current state-of-the-art sensor interface electronics, an approach using analog storage provides interesting data when compared with more mature technology such as ADC:s and memory logic.

Future work will reveal whether if a reactive architecture based on an analog storage approach will show improvements in energy consumption similar to those of advanced ADC. If such improvements are shown, the analog storage approach has clear merit for use in future WSN node designs.

ACKNOWLEDGMENT

The authors would like to thank the ESIS project supported by EU structural funds for financial support.

REFERENCES

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless sensor networks,” in *Proc. Hawaii Intl. Conf. System Sciences*, Hawaii, Jan 4-7 2000, pp. 3005–3014.

[2] J. Eliasson, M. Lundberg, and P. Lindgren, “Time synchronous bluetooth sensor network,” in *Proc- IEEE Concumer Communication and Networking Conference, CCNC*, 2006.

[3] M. Lundberg, J. Eliasson, L. Svensson, and P. Lindgren, “Context aware power optimization of wireless embedded internet system,” in *Proceedings IEEE IMTC*, 2004.

[4] J. Eliasson, P. Lindgren, J. Delsing, S. J. Thompson, and Y.-B. Chen, "A power management architecture for wireless sensor nodes," in *Proc IEEE Wireless Communication and Networking Conference, WCNC*, 2007.

[5] V. Loscri, G. Morabito, and S. Marano, "A two level hierarchy for low energy adaptive clustering hierarchy (TL-LEACH)," in *Proc. 62nd IEEE Vehicular Technology Confere)nce (VTC-Fall*, Dallas, 25-28 September 2008, pp. 1809–1813.

[6] Y. Xu, J. Heidman, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proc. Mobicom*, 2001, pp. 70–84.

[7] J. Delsing and P. Lindgren, "Sensor communication technology towards ambient intelligence, a review," *Meas. Sci. Technol.*, vol. 16, pp. 37–46, 2005.

[8] J. Lu, F. Valois, M. Dohler, and M.-Y. Wu, "Optimized data aggregation in wsns using adaptive arma," in *Proceeedings Sensorcomm 2010*, 2010, pp. 115–120.

[9] S. G. Hong, N. S. Kim, C. S. Pyo, and W. W. Kim, "Hybrid sensor module and data processing using low-power wakeup in wsn," in *Proceeedings Sensorcomm 2010*, 2010, pp. 191–195.

[10] S. D. Rosenbaum and J. T. Caves, "8192-bit block addressable ccd memory," *IEEE JOURNAL OF SOLID-STATECIRCUITS*, vol. SC-10, no. 5, pp. 273–280, Oct. 1975.

[11] J. Borg and J. Johansson, "A method for experimental separation of charge trapping and incomplete transfer in ccds," *I E E E Transactions on Electron Devices.*, 2011.

[12] Pentronic, "Resistance thermometer theory." [Online]. Available: http://www.pentronic.com/Theory/Pt100sensor/tabid/188/language/en-GB/Default.aspx

[13] ——, "The effect of 2,3 or 4 wire connection using pt100/rtds." [Online]. Available: http://www.pentronic.com/Portals/0/PDF/En/Useful%20links%20pdf/The_effect_of_2_3_4_wires_on_Pt100_060210.pdf

[14] D. Gerna, M. Brattoli, E. Chioffi, G. Colli, M. Pasotti, and A. Tomasini, "An analog memory for a qcif format image frame storage," in *Circuits and Systems, 1996. ISCAS '96., 'Connecting the World'., 1996 IEEE International Symposium on*, vol. 1, May 1996, pp. 289 –292 vol.1.

[15] R. Carmona, S. Espejo, R. Dominguez-Castro, A. Rodriguez-Vazque, T. Roska, T. Kozek, and L. Chua, "A 0.5 μm cmos cnn analog random access memory chip for massive image processing," in *Cellular Neural Networks and Their Applications Proceedings, 1998 Fifth IEEE International Workshop on*, Apr. 1998, pp. 271 –276.

[16] M. van Elzakker, E. van Tuijl, P. Geraedts, D. Schinkel, E. Klumperink, and B. Nauta, "A 10-bit charge-redistribution adc consuming 1.9 muw at 1 ms/s," *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 5, pp. 1007 –1015, May 2010.

[17] B. Murmann. (2011) Adc performance survey 1997-2011. [Online]. Available: http://www.stanford.edu/~murmann/adcsurvey.html

[18] "Data sheet m16c." [Online]. Available: www.renesas.com

[19] J. Johansson and J. Delsing, "Energy and pulse control possibilities using ultra-tight integration of electronics and piezoelectric ceramics," in *Proc. UFFC*, vol. 3, 2004, pp. 206–2210.

[20] J. Borg and J. Johansson, "Optimization of the design of an integrated ultrasonic preamplifier," in *Proceedings of the International Congress on Ultrasonics*, Vienna, April 9-13 2007.

[21] ——, "An ultrasonic transducer interface ic with integrated push-pull 40 vpp, 400 ma current output, 8-bit dac and integrated hv multiplexeri," *IEEE Journal of Solid State Circuits*, vol. 46, no. 2, pp. 475–484, 2011.

[22] E. M. I. Gustafsson, J. Johansson, and J. Delsing, "A cmos amplifier for piezo-electric crystal interfaces," in *Proceedings of 11th MIXDES conference*, Szczecin, Polen, 2004.

Cluster-based Performance Analysis of Sensor Distribution Strategies on a Wireless Sensor Network

Majid Bayani Abbasy, José Pablo Ulate
Universidad Nacional de Costa Rica, Escuela informática
Universidad Nacional de Costa Rica, Escuela informática
mbayani@una.ac.cr, jose.ulate.castro@est.una.ac.cr

Abstract— **Since the placement of sensor nodes has a direct effect on its performance, this paper explores three predefined configurations in order to compare their power consumption performance. The experiments assume an obstacle free, rectangular field, with a cluster-based routing protocol and the event location in the field. The results demonstrated the different significant behaviors of the placement strategies in a cluster-based scenario.**

Keywords-Wireless Sensor Network (WSN); Sensor placement strategy; WSN performance; Cluster-based.

I. INTRODUCTION

Wireless sensor networks (WSNs) consist of a group of sensor nodes, distributed over a field, in order to sense and collect the data of an event. Among the constraints that affect the WSNs performance such as the low availability of energy, low bandwidth, sensing limitations and sensor lifetime, *energy consumption* is the most challengeable [1].

In order to improve the WSN performance huge number of techniques has been developed before the writing of this paper. Some of them concentrate on routing protocols such as Directed Diffusion, Leach [24], BVR, VRR and Gear [2], to decrease the power consumption of sensor nodes. Using ultra-low power data storage [7], new low-power processors [5], and transmitters [4], the efficiency of WSNs has significantly improved. Applying different positioning strategies for sensors is another complementary approach to increase the efficiency of a WSN performance that has been considered by some researchers lately [3].

Two main factors are considered and evaluated in this research. The first being the sensor location strategy and its effect on the WSN performance, and the second factor of this investigation is the cluster-based approach.

The focus in this paper is mostly on the second factor or the cluster-based approach.

The results showed that the first factor (sensor placement) in a cluster and query based environment has a direct impact on WSN performance. However, in many query-based cases, the main constraint is the cost of manually locating and replacing sensors nodes. Therefore, it was considered essential to evaluate the clustering scenarios of the WSN and their effect on their energy consumption performance.

A. Related Work

Sensor deployment is a topic, which has been studied by many researchers. Chen et al. studied the placement of a given number of sensors to maximize WSN lifetime per unit cost [4]. Another approach in this route was discussed in [8] considering the joint optimization of sensors for data gathering, where a given number of nodes needs to be placed in a field in such a manner that sensed data can be reconstructed at a sink, while minimizing the energy consumed for communication. A constrained multi-variable nonlinear programming problem is formulated in order to determine the optimal location of the nodes in [6]. Also, [3] a comparison of random allocation and two different geometric placements in order to find the optimized sensor placement strategy among the defined strategies and consequently, increasing the WSN performance in terms of power consumption has been undertaken. The [3] experiments are realized in a flat-based environment using Directed Diffusion Protocol.

The main objective of this paper is the study of geometric configuration of the sensor nodes in a clustered environment.

The main idea is based on finding the impact of the node placement strategy among the defined strategies and consequently, increasing the WSN performance in terms of power consumption.

Overall performance under different placements depends on the pattern of queries requested from the network, which in turn might depend on the physical segmentation of the filed. For example, queries for a volcanic surveillance will be carried out from a small number of sites at some distance from a (central) crater, to those sensors, which are found closer to the location of the volcanic activity. In this particular occasion, the filed around the target can be segmented into heterogeneous and homogeneous regions. Theoretically, the clustering the heterogeneous part of the scenario can be a realistic option. This paper emphasizes the use of a geometric clustered-based approach to obtain a significant increase of the energy consumption performance. The strategies, which are deployed, are compared to one another to determine which one optimizes power consumption. This paper presents a realistic simulation-based research whereas all requests for information originate at random points in of the field, and are directed to the selected clustered pieces of the area.

II. METHODOLOGY

This section describes in detail, the methodology used in the simulation-based experiments. The strategies for sensor allocation are described subsequently.

A. Sensor Placement Strategies

The two main strategies used are random and plan-based. Both are the Cluster-based types.

The two planned strategies which were used are: Uniform and Circular. The Random Strategy is a default common strategy and the Uniform Strategy is the theoretical optimal placement to reduce the number of nodes on a field without obstacles, and the Circular Strategy was an arbitrary plan choice. All strategies were implemented on a small clustered filed in a 50x50. Sensors can only be located at grid intersections.

The experiments are divided into two main categories: with a leader and without a leader:

With a leader: for all kinds of the sensor placement, a sensor is selected as the leader and located on the center of experimented clustered piece.

Without a leader: in each cycle of communication each one of 16 sensors may act as a leader.

1) Random Placement

In Random placement, the sensor nodes are randomly scattered over a previously allocated intersection on the grid and a random distribution scenario is generated. Figure 1 is an illustration of a random distribution for 16 sensor nodes.

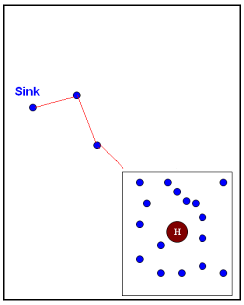


Figure 1. Random Clustered-Distribution of 16 sensor nodes.

2) Plan-Based Placement

In the plan-based strategies, sensors are placed in the field following a predefined geometric plan or using a predefined distribution algorithm.

a) Uniform Distribution

In the uniform strategy, a sensor was placed at each intersection on a selected clustered area. Given the number of sensors used in the experiments, and the predefined area of coverage of each sensor, this type of strategy provides 100% connectivity. Figure 2 illustrates this distribution for 16 sensors with a leader on the center.

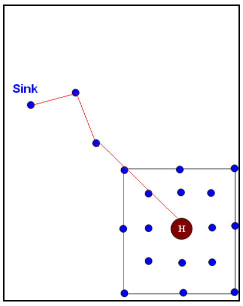


Figure 2. Uniform Clustered-Distribution of 16 sensor nodes.

b) Circular Distribution

The Circular Distribution Strategy is an arbitrary option; the sensors form a predefined circle of nodes distribution that radiates symmetrically from the center. This distribution can be considered as an alternative arrangement of distributing the sensors. Figure 3 shows a Circular distribution for 16 sensors with the head cluster in the center.

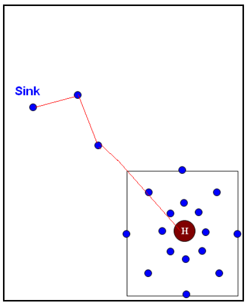


Figure 3. Circular Clustered-Distribution of 16 sensor nodes.

B) Power Consumption Measurement model

As part of the research objectives, the performance analysis was done on power usage at different abstract levels (network, query, individual sensor). In order for the simulation to behave in the same way as it would in reality, battery characteristics have to be defined.

In order to achieve this goal, a particular type of sensor was chosen, and its main characteristics were mapped to the simulation model (battery life, transmission range, and so on). The sensor chosen was the MICA2DOT which is part of the Mica family [12] due to its popularity in WSN design. Its transmission speed is 250 Kbps, and its outdoor range is 75-100 m. The maximum battery capacity is 2000 m.A.h. Some of the relevant energy characteristics of this type of sensor are summarized on Table I.

TABLE I. ENERGY USAGE BY ACTION TYPE

Action	**Energy used (µ joules)**
Reception of data message	100
Reception of control message	3
Transmission of data message	100 + 200 x $\boldsymbol{d}^2$
Transmission of control message	3 + 64 x $\boldsymbol{d}^2$

As an assumption, the simulation required some simplification over the traditional battery consumption behavior. The total power consumption by each node was based on the estimated consumption rate in “wake up” mode. Another important assumption is that the “wake up” mode includes only transmit and receive modes only, while energy consumed in both the idle and sleep states is ignored. Each time a message (data or control) passes through a node, the energy spending for that node is reduced in the amount necessary to transmit all the messages.

C) Experimentation Design

A C-Sharp Interface was implemented in order to simulate the experiments.

When a sink injects a query from any point of the scenario, a possible leader (head cluster) receives the message and sends the control message to the first sensor node in the cluster, consequently, the target node will send back the DATA to the leader. The leader receives the DATA, consumes energy as well as the sender, and will send another control message to the other nodes and repeat the process. The procedure of sending and receiving the control messages between leader and all 16 sensors was defined as a *Query Cycle.* The sensors can die as they do not have sufficient energy to receive or transmit the DATA or control a message in different query cycle.

As mentioned above, the experiments are cluster-based and are implemented under two categories: with a leader and without a leader.

In the case of experiments with a leader, a particular sensor allocated in the center called the Leader (head cluster). In the without leader option, each of the nodes can be act as a leader (head cluster) and when it has died another sensor takes the role of conducting the process.

The details will be explained in the next subsections.

1) With leader

In a typical scenario as shown in Figure 4, 16 sensors are deployed. The leader of the cluster (HC) is located in the center, sequentially it injects a query into the WSN and a flooding algorithm [3] propagates the query (interest message) over the WSN. When the sink injects a query, all those sensors with enough power to receive and transmit the sensory data back to the leader. All sensors that participate in the process of data delivery consume energy. ET_i (µj)) represents the total energy consumed by the message and data delivery is calculated as the sum of the energy consumed when the i-th query cycle starts, the network has already used some energy for the processing of *i-th* queries.

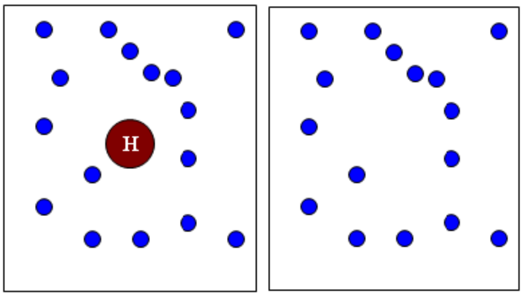


Figure 4. An experimental scenario With and Without Leader.

2) Without a Leader

In this scenario each of the 16 sensor nodes acts as the leader of the cluster (HC). As an assumption, the sequence of the change is ignored in this experiment because, all sensors are charged the same amount of energy and what is calculated is the sum of energy consumed at the end of the sequence. The rest of the process is the same as the previous section (with a leader).

III. RESULTS

The results obtained under normal and stress conditions in the simulation are organized in different categories. The analysis starts with the total energy consumption under non-stress critical point conditions. Under normal conditions, no sensors can die of energy exhaustion, so unsuccessful queries, that create energy consumption variability, are due to non-reachability issues in the Random strategy.

A) Analysis of the results

At the most critical point, whereas the first sensor was dead, the behavior of the three strategies is varied. The result shows a significant difference between three strategies. The Uniform that theoretically, because of scattering of the nodes, uniformly, total energy consumption, is lower than the Circular and Random strategies. In the Uniform Distribution Strategy more sensors than the other configurations will be involved in the process.

In the particular case of Circular scattering (Figure 5), it demonstrates a reasonable response in terms of the total power consumption (µj) compared to the Random. This fact is shown in the Figure 5.

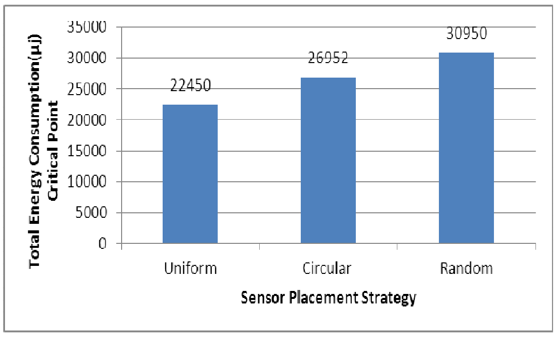
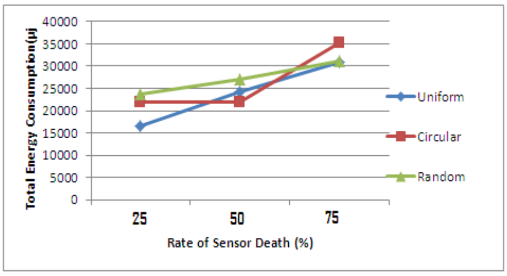
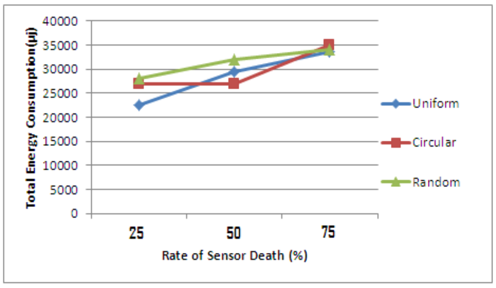


Figure 5. Total Energy Consumption in the Critical Point.

Figures 6(a), 6(b) show total energy consumption for different strategies under stress situation. The results were obtained for three stress situations. The total energy consumed by all sensor nodes is calculated when 25%, 50% and 75% of the sensors are dead. As the results show, the best result is that of the Uniform Strategy. The Random and Circular cases show the closed outcome in terms of the total energy consumption. When a high rate of the nodes was dead, the Random Strategy was close to that of the Uniform Strategy. This is to be expected as more sensor nodes are involved in any query in the Random Strategy case. Based on the Figures 6)a ,6)b , the result did not demonstrate any significant difference between the results for both with leader and without leader cases.



6 (a) Total Energy Consumption (with leader) in different rate of sensor death.



6(b) Total Energy Consumption (without leader) in different rate of sensor death.

Another useful result that is revealed in Figure 7, is the total energy consumption for all strategies in different cycles of the query in the normal condition where none of sensors dies. As Figure 7 demonstrates, all strategies have the similar answer. The Uniform Strategy shows a more active strategy when compared to the others. The Random Strategy shows a lower activity in opposition to Circular Strategy and the Uniform Strategy. Based on these aggregate statistics for normal conditions, the rate of increasing energy consumption for all cases is smooth and very similar.

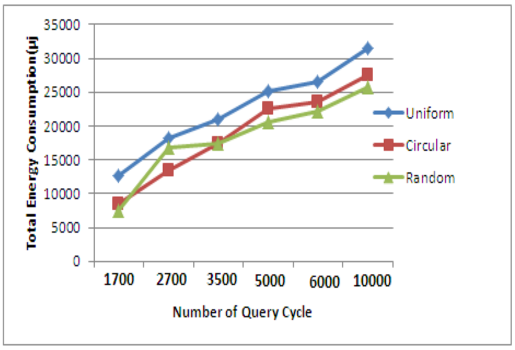


Figure 7. Total Energy Consumption for different Query Cycle.

Figure 8, exhibits another essential piece of data obtained in the experiments. As shown in Figure 8, the number of sensor that died in the process during different cycles is varied. In the Uniform strategy, the higher number of nodes died as a result of higher activity that was expected from the Uniform distribution. When starting the query cycles, a lesser number of the nodes were dead. As the cycles increased, more sensors died during the communication process in the Circular Strategy clustered area as well as the Uniform Strategy. The death rate of the nodes for Uniform Strategy is almost two times the same rate for the Random Strategy. This is due to the fact that the sensor nodes were participating in the communication process in the Uniform Strategy and were more active than the Random Strategy case. It is acceptable to state that when the sensors are distributed in a cluster area based on a geographic plan they will be more active than when they were scattered randomly in a WSN clustered filed.

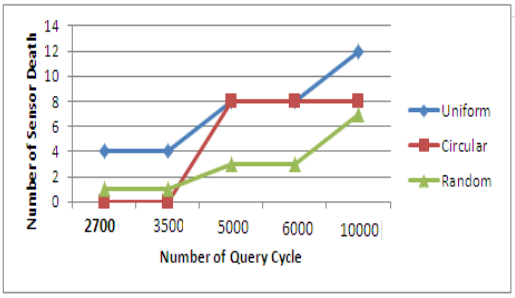


Figure 8. Rate of Sensor death for three Sensor Placement Strategies.

As a final point, Figure 9 shows the different amounts of the average power consumption for different query cycles. Based on the Figure 9, the average energy consumption pattern for all strategies is similar whereas the Uniform Strategy spends less total energy for almost all queries and in most cycles, as expected.

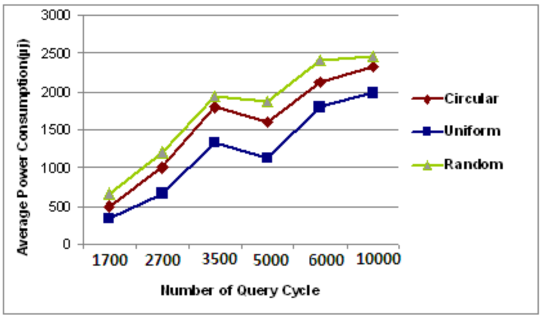


Figure 9. Average Power Consumption for different Query Cycle.

The explanation for this fact, as discussed before, relates to the ability of the Uniform layout of constructing the closest-to-straight path between leader and target nodes. However, the Uniform Strategy has the highest rate of sensor death, which means that with each subsequent query cycle, the paths are of lesser and lesser quality. This is apparent in the fact that the differences between the Uniform Strategy and the other strategies are fewer than under normal conditions. Although, the plan-based Circular Strategy shows a very similar answer to the Random, on the whole, the results were located in a lower level of energy consumption than de Random case by the sensors.

Lastly consideration is: for all cases, the result of the experiments with HC and without leader hasn't shown any significant difference.

6) CONCLUSION AND FUTURE WORK

Simulation results obtained under stress situation for clustered sensor placement have confirmed that plan-based strategies use less total energy than Random deployment strategies. In particular, Plan-based sensor distribution strategies demonstrate better response in terms of the total energy consumption in the system.

Under normal conditions, the Uniform Strategy is the best-proposed strategy for applying in terms of the participating of the sensor nodes and high covering the cluster area. In this case the energy consumption behavior of the all strategies relatively, is similar.

Another significant issue that was observed is, in the critical point that first sensor is dead, the results are very reasonable. As was theoretically expected, the plan-based strategy and the Uniform Strategy, in particular, is the most optimized strategy in terms of the total energy consumption by all sensors that are scattered over the cluster.

Finally, the rate of sensor death was different for all strategies. The first death of the sensor occurs in the Uniform due to the high rate of sensors participating in the process. In different query cycles, this rate varied. The circle Strategy starts with a delay but as the cycles increases the participating level of the nodes in this strategy increases, as well. The Random always keep the same rate compared to the plan-based strategies.

Due to the fact that the selection of deployment of a sensor strategy has a significant effect on WSN performance, a strategic focus for future work could be the task of discovering optimized cluster-based sensor placement strategies on the different WSN scenarios using a particular sensor placement strategy. This investigation is a query-based research.

An additional line of research could focus on the modeling of possible Circular sensor distributions for volcanic monitoring is an example of ***a real implementation*** focus for researchers of this paper.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks", IEEE Communication Magazine, Vol. 40, No. 8, August 2002, pp. 102-114.

[2] J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey", IEEE Wireless Communications, Vol. 11, Issue 6, December 2004, pp. 6-28.

[3] M. Bayani Abbasy, "Performance Analysis of Sensor Placement Strategies on a Wireless Sensor Network",IEEE Fourth International Conference on Sensor Technologies and Appalications , July 2010.

[4] Y. Chen, C. Chuah, and Q. Zhao, "Sensor placement for maximizing lifetime per unit cost in wireless sensor networks", IEEE Military Communications Conference, MILCOM 2005, October 2005, pp. 1097-1102.

[5] L. Ciaran and F. O'Reilly, "Processor Choice for Wireless Sensor Networks", Workshop on Real-World Wireless Sensor Networks REALWSN'05 Sweden, June 2005.

[6] G. Deepak and B. Beferull-Lozano, "Power-efficient Sensor Placement and Transmission Structure for Data Gathering under Distortion Constraints", ACM Transactions on Sensor Networks (TOSN), Vol. 2, Issue 2, May 2006, pp. 155 – 181.

[7] G. Deepak, P. Desnoyers, and P. Shenoy, "Ultra-low Power Data Storage for Sensor Networks", In Proceedings of the Fifth International Conference on Information Processing in Sensor Networks, April 2006, pp. 374- 381.

[8] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocols for Wireless Micro Sensor Networks", Proceedings of the 33rd Hawaii International Conference on System Sciences, Vol. 8, January 2000, pp. 20-29.

[9] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", In Proceeding of Mobicom'2000, August 2000, pp. 56–67.

[10] F. Silva, H. Heidemann, and R. Govindan, "An Overview of Directed Diffusion", to appear in Frontiers in Distributed Sensor Networks, ISI-TR-2004-586, February 2004.

Liquid Level Sensor in Automotive Design

Mehmet Emre Erdem
Institute of Science and Technology
Istanbul Technical University
Istanbul, Turkey
emre_erdem@rocketmail.com

Dogan Gunes
Energy Systems Engineering Department
Istanbul Bilgi University
Istanbul, Turkey
dogan.gunes@bilgi.edu.tr

***Abstract*—New materials and technological developments in electronics and computers have changed all of the industries, as well as the world itself. These technological advancements have evolved automotive industry by redefining the concepts like: performance, efficiency, fuel consumption, driving dynamics, ergonomics etc. to a level far beyond expected. In this paper, the development stages; including the finite element analysis of the current sensors, using finite element analysis to design a new sensor, is elaborated. The new model designed in this study is computationally validated for production purposes and is planned to be experimentally tested. Also, in this paper a novel liquid level sensing device development strategy is presented in detail.**

Keywords – sensor; finite element analysis;modal analysis; durability analysis; whispering gallery mode (WGM).

I. INTRODUCTION

Oil sensors and analyzers are used in automotive and industrial applications to gather or send valuable information to ensure that the level of the engine oil does not become dangerously low without being noticed. The sensor monitors the oil level continuously during the entire engine operation, which means that the oil level can be prevented from falling below the minimum level during operation, which in turn means the oil film is not interrupted (which would lead to engine damage). Secondary influences such as the slope of the vehicle's lateral and longitudinal accelerations are compensated by the vehicle control unit calculating a mean value [1].

There are many oil level sensors in automotive, but most common approaches are ultrasound and resistance sensors [29]. Sensors using a resistance wire work with the principle of changing resistance and temperature of the wire between under and over the liquid. Sensor sends a current to the wire, and the output voltage differs by changing liquid levels.

The ultrasonic level sensor, however, works on the principle of measuring the time-frame between transmitting and receiving of ultrasound waves. The ultrasound wave travels through material [2]. The sensors emit high frequency (20 to 200 kHz) acoustic waves that are reflected back to and detected by the emitting transducer affected by the changing speed of sound according to moisture, temperature, and pressures. Correction factors can be applied to the measurement level to improve the accuracy of measurement [3]. Ultrasonic sensors have advantages in dynamic measurements; nevertheless they are more complex and more expensive when compared with the resistance wire type level sensors.

At this point, new approaches were searched and a possibility of applying Whispering Gallery Mode (WGM) theory into liquid level sensing is considered [14] [19].

II. LIQUID LEVEL SENSOR DEVELOPMENTS IN HISTORY

There are many different techniques in literature for measuring the liquid levels. Although, the concept development of sensors is parallel with the technological developments in electronics, some applications are much older than the electronics, like the Archimedes’ principle.

Archimedes’ principle also works for liquid level measurement. A cantilever beam is mounted into the liquid tank, and Archimedes’ principle raises the beam through the liquid, and load cell measures the deflection [4].

Another method which uses ultrasonic lambda waves, was developed in Russia (2002). The principle of this study is based on the changing characteristics of lambda waves in the liquid environment [5].

Ultrasonic level sensor for liquids working under high pressure [6] was developed and performed by NASA Langley research center and Old Dominion University. The basic principle can be explained as follows; the ultrasonic waves sent by a transducer propagate and are reflected from the liquid surface, and the time between transmission and reception is converted to distance.

An ultrasonic wave propagation method was developed in 2001 by BFGoodrich airplane advanced sensors department [7]. The principle is the same with the study before, sending and collecting ultrasonic waves, and finding the liquid levels according to the wave transfer times.

A totally different method is the usage of fiber optics in liquid level measurement [8]. Fiber optic cables reflect the laser/light signal when in the air. However, when the density difference reduces, it transfers the laser light to the liquid environment, and thus, the sensor detects the liquid environment.

Fiber optics is used again in another study [8]. The methodology used in [8] is as follows. While in air, most of the light rays reflect back, but in water light rays continue their way. The decrease in the level of reflected signals indicates that the optical fiber tip is inside water. The simplicity of the methodology implies that within the material property boundaries (between -20C and +70C) usage of optical fibers is easy, accurate and inexpensive.

ISBN: 978-1-61208-144-1

In the case for automotive engine applications, however, although all of the above methods have different advantages, a new method is required. The design must have high precision, and needs to be small enough to be located into the engine and must be durable enough for working under flammable liquids or gasses safely.

III. New Oil Level Sensor DEsign

A. *Hella Sensor Design*

This study was first started with the idea of generation and development of a new oil level sensor which is integrated inside an oil pan (Fig. 1). The initial plan was to study with the engineers in Hella Company's (Hella KGaA Hueck & Co.) automotive sensors division [26].

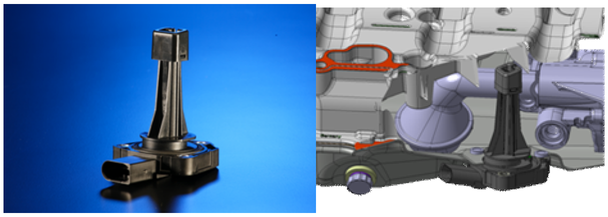


Figure 1. The current Hella oil level sensor, the picture in the right shows the sensor in it's assembled position and Target of OLS Design [25].

According to the mutual study plan, available information is shared. The information include previous three dimensional (3D) computer aided design (CAD) models, design and finite element analysis (FEA) studies, as well as the test results of the previous ultrasonic oil level sensor. For this aim, Hella Company has been visited, and during this visit details of the ultrasound sensor, its development, production and validation stages have been evaluated. Design validation tests of the sensor are thoroughly investigated to setup a relevant computational model.

The target of the Hella cooperated study is to work on the mechanical designs (material, sealing, mounting) in order to integrate the ultrasonic oil level sensor inside the vehicles oil pan.

For design verification purposes, a FEA simulation model is developed to study overall thermo-mechanical performance and vibration influence. Following the computational verification of the model developed, rapid prototypes of the model are planned to be produced to perform the first functional tests by Hella.

The main details of the FEA analysis are; the vibration profile and mechanical properties of the new sensor design under various temperature ranges, which are between -40C up to +150C. The endurance of the new design will be evaluated by using computer aided engineering (CAE) methods.

Following the verification of an acceptable design, the physical samples made out-of rapid prototyping will be prepared, and tested submerged in oil within the real world experimental setup. The target is to reach an accuracy of -/+ 0,2mm for sensing the oil levels.

In the present the study, several sensor models have been analyzed, and a new, reliable design proposal is developed (Fig. 2)

IV. Finite Element Analysis and evaluation of Results

During the FEA analysis, material properties of PA66-GF35 [25] is assigned for the main sensor parts, while fixings are CuZn39Pb3 and grounding is CuNi10Fe1Mn [24]. Several meshing structures are tested and an optimum mesh structure is reached. Final mesh structure consists of over 76.000 tetrahedral elements and around 22.000 nodes. Following the "meshing" stage of parts, all the individual parts are assembled with an attention to the rigidity of each part and the model. During the analysis it is assumed that the oil level is low, which also demonstrates the worst case by minimizing the damping effect of oil in the sump.

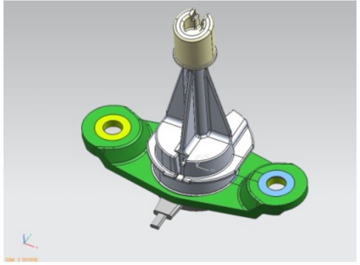


Figure 2. New oil level sensor design according to the results of FE analysis and the requirements of Hella GmbH.

A. *Modal Analysis*

The goal of modal analysis [27] is to determine the natural mode shapes and frequencies of an object or structure during free vibration. Since the model developed is of complex structure it is customary to use the finite element method (FEM) to perform this analysis,

The types of equations which arise from modal analysis are those seen in Eigen systems. The physical interpretation of the eigenvalues and eigenvectors, which are determined by solving the system, correspond to the natural frequencies and related mode shapes. Usually, the modes that are dominant are the lowest frequencies which are the most prominent modes at which the object will vibrate, dominating all the higher frequency modes [23].

A sample of the results evaluated is presented in Fig 3. The results indicate very high strain and stress values only in 7th, 8th and 9th modes, where the frequencies are 1635 Hz, 2187Hz, and 2413Hz.

All of the above frequencies are higher than the testing limits: (According to DIN EN 60068-2-6) [26], and our design is completely safe according to natural frequencies found in the modal analysis.

The above conclusion is demonstrated in depth in the next section, which summarizes the stress analysis results under the testing conditions.

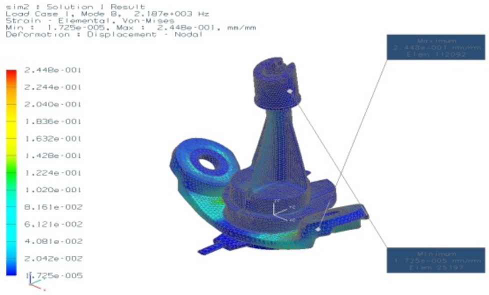


Figure 3. The mode 2 shape and the elemental strain values of the FE analysis.

B. *Stress-Strain Analysis in Freqency-vs-Acceleration*

Frequency vs. acceleration values obtained from the Volkswagen Automotive test spec [26] in Table 1., are applied to the new sensor model designed and meshed as summarized in Part A.

TABLE I. THE FREQUENCY VS ACCELERATION VALUES

Frequenz [Hz]	**Amplitude der Beschleunigung [m/s^2]**
100	100
150	150
200	200
240	200
255	150
440	150

Results of the stress-strain analysis under test conditions (Fig. 4) show that the oil level sensor faces a stress value of not more than 0.21 MPa, which makes no deformation when compared with its Yield Stress and Ultimate Tensile Strength values which ranges within 150-200MPa. Also the results show that there are no significant strain values.

In summary, computational results indicate that, in these test conditions, the new design is assessed to be safe.

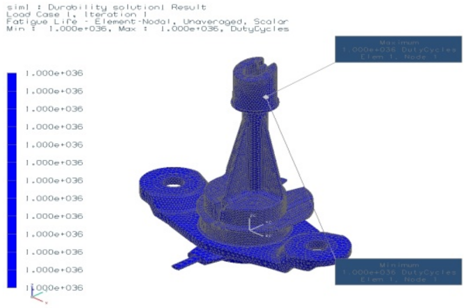

sim1 : Durability solution1 Result
Load Case 1, Iteration 1
Fatigue Life - Element-Nodal, Unaveraged, Scalar
Min : 1.000e+036, Max : 1.000e+036, DutyCycles
1.000e+036
Maximum
1.000e+036 DutyCycles
Elem 1, Node 1
Minimum
1.000e+036 DutyCycles
Elem 1, Node 1

Figure 4. Fatigue Analysis Sample Results

C. *Fatigue Analysis*

Post processing of the results obtained show that, the stress values may not harm the design. However, the new model needs to be analyzed according to fatigue conditions to see the effects of cycling loads.

For this aim Fatigue Strength Coefficient, Fatigue Strength Exponent, Fatigue Ductility Coefficient and Fatigue Ductility Exponent values have been found and inserted into the materials' database of the computational model [24].

Same loading conditions are repeated for many times to see the life-time of the model under the test conditions. The results show that expected lifetime is above 10^36 cycles. Safety factors have minimum values of 25.76 for fatigue and 2.796 for strength. According to these findings it can be suggested that, the new design is absolutely safe according to both stress-strain and fatigue properties under cycling loading of test conditions.

V. WHISPERING GALLERY MODE PHENOMENON

A. *Theory*

Following the above study, exploring the capabilities of Southern Methodist University Mechanical Engineering Faculty’s Micro Sensors Lab. a totally new type of liquid level sensor idea is activated.

This new sensor is planned to detect pressure/force by using whispering gallery mode (WGM) phenomenon, and capable of working nearly in everywhere including underwater.

The target is to develop sensor prototypes using the existing WGM theory. The test design includes equipments such as; laser as the signal source, micro sphere as the sensing element, fiber optics, and a liquid container as the test medium (Fig. 5). It is planned to perform the functional tests by using the Micro sensors laboratory of Southern Methodist University.

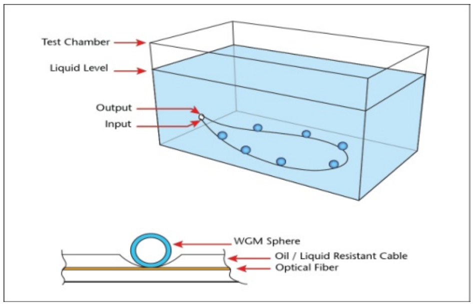


Figure 5. Planned experimental setup of liquid level sensor using WGM method

Fig. 5 and Fig. 6 show the proposed WGM pressure sensor experiment setup.

Fig. 5 shows the general experiment setup, while Fig. 6 shows detailed proposal of how to encapsulate the WGM sphere to make the sensor resistant to the medium including liquids [10].

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

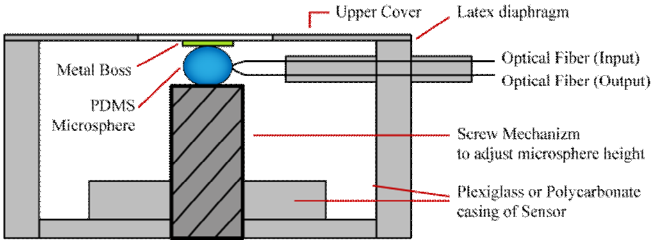


Figure 6. WGM sensor should be resistant to fluids therefore a latex type membrane will be planned and used in the experiments

As a summary; results of the present work is aimed to include not only a new level sensor, but also a new perspective into the WGM phenomenon will be evaluated to create a completely new sensor to be used in automotive as well as in many other branches of industry.

B. *Equations*

In the whispering gallery mode theory, light undergoes total internal reflection, and because it is trapped inside the sphere, WGMs are observed under certain conditions.

The details of the theory, the change in WGM spheres under pressure is as follows [19]:

For r >>λ, resonance condition (approximate):

$$2\pi\ r\ n0 \approx \ell\ \lambda \qquad (\ell = \text{integer}) \tag{1}$$

n = Refractive index of the micro-sphere
λ= Wavelength
r = Micro-sphere radius

$$\frac{\Delta n}{n_0} + \frac{\Delta r}{r} \sim \frac{\Delta\lambda}{\lambda} \tag{2}$$

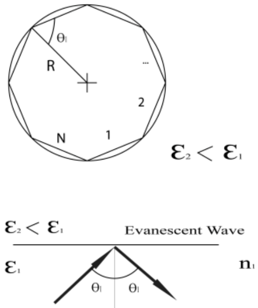


Figure 7. Total internal reflections of the sensor, according to basic optical physics.

When we look at Fig. 7 above, it can be seen that each time the light bounces off the inner surface of the sphere due to total internal reflection, the reflected wave experiences a “phase delay”,ϕ. This phase delay is a function of the light wavelength, λ; and incidence angle; as well as the sphere-to surrounding refraction index ratio, n_1/n_2.

As it can be seen easily, the laser light comes into the microsphere in its contact point in the tapered film, and the light undergoes total internal reflections in the sphere, which causes the phase shifts or in other words WGM shifts.

As illustrated in Fig. 8, while light starts re-circulating in the sphere a resonance shift occurs in the light when compared with the reference light beam.

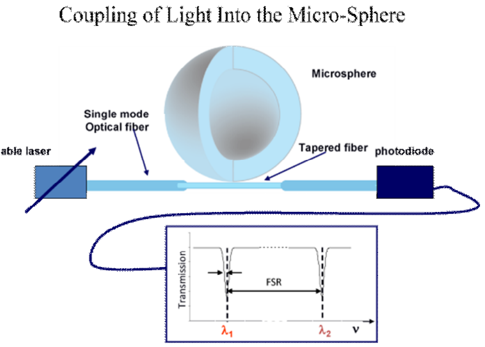


Figure 8. The WGM phenomenon, and the phase shift (or the WGM shifts) caused by the total internal reflections occured in the microsphere [19].

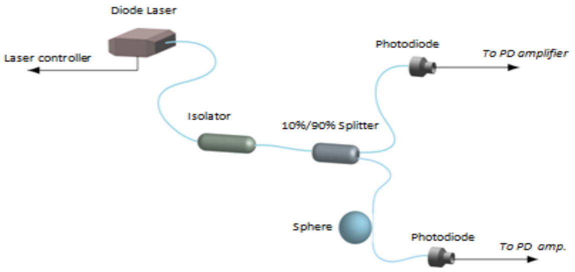


Figure 9. The experiment setup in it’s general form. The light coming from the laser splits into two [19].

Fig. 9 presents the experiment setup in its general form. The light coming from the laser splits into two, leaving a small percentage for reference going directly to the photodiode, while keeping a high percentage (90% in this scenario) for WGM shift, which goes directly to the microsphere. So after the WGM phase shift occurs, it has been noticed by checking the differences between the reference light and WGM part of it

When force is applied to the resonator, a change occurs in both the shape and index of refraction of the resonator (Fig. 10). The formulas below demonstrate that we can measure the force by detecting the changes occurred in the shape and the index of refraction [19].

$$F = f(\Delta\lambda) \tag{3}$$

$$\frac{\Delta n}{n_0} + \frac{\Delta r}{r} = \frac{\Delta\lambda}{\lambda} \tag{4}$$

$$F = g(\Delta\lambda) = f\left(\frac{\Delta R}{R}, \frac{\Delta n}{n_0}\right) \tag{5}$$

ISBN: 978-1-61208-144-1

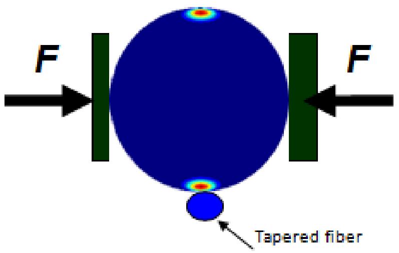


Figure 10. When force is applied to the resonator, a change occurs in both the shape and index of refraction of the resonator [16].

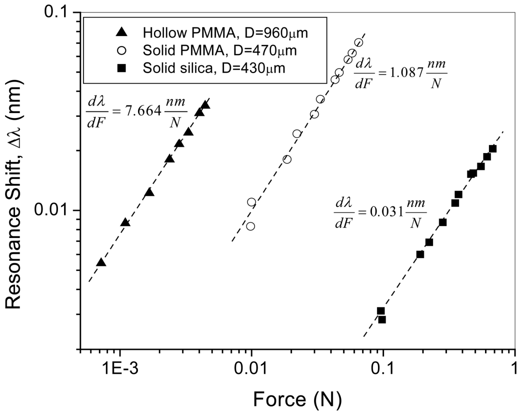


Figure 11. Force versus Resonance results for different spheres [16].

In this manner, there are different kinds of sphere materials showing different responses to the measurements. Fig. 11 shows the different sphere materials and their specific values like Young's modulus, index of refraction and Elasto-optical constant which all are related with the WGM measurement results.

It can be observed that, the force shifts the resonance, but more importantly according to different materials, resonance shift also differs.

Now if we use WGM phenomenon to develop a pressure sensor we can see that pressure changes of the medium surrounding the sphere will induce WGM shifts.

This means by using the same formula [19]:

$$\frac{\Delta n}{n_0} + \frac{\Delta a}{a} = \frac{\Delta \lambda}{\lambda} \qquad (6)$$

We can reach to the pressure as a function of the differences of sphere radius and index of refraction [19]:

$$P = g(\Delta\lambda) = f\left(\frac{\Delta a}{a}, \frac{\Delta n}{n_0}\right) \qquad (7)$$

If the stress and strain of a sphere is investigated, first the coordinate system should be defined (Fig. 12)

After many calculation steps, the formulas below can be achieved [16].

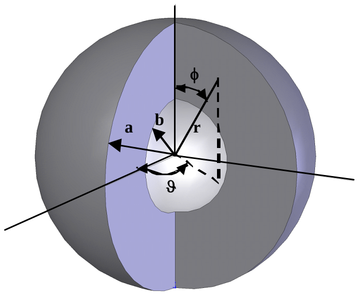


Figure 12. Coordinate System used [16].

$$\frac{dn}{n} = \frac{n_r - n_{0r}}{n_{0r}} = \frac{n_\vartheta - n_{0\vartheta}}{n_{0\vartheta}} = \frac{n_\phi - n_{0\phi}}{n_{0\phi}} = \frac{C(\sigma_{rr} + \sigma_{\vartheta\vartheta} + \sigma_{\phi\phi})}{n} \qquad (8)$$

By inserting the appropriate expression for the three principle stresses in $\frac{\Delta n}{n_0} + \frac{\Delta a}{a} = \frac{\Delta \lambda}{\lambda}$ we can obtain the dependence of the WGM shift on external pressure P_0 for a PMMA spherical resonator as follows [19]:

$$\frac{\Delta\lambda}{\lambda} = \frac{1}{2G} \frac{P_0}{1-\left(\frac{b}{a}\right)^3} \left(\frac{\frac{1-2\nu}{1+\nu} + \frac{1}{2}}{2G\left[\left(\frac{a}{b}\right)^3 - 1\right]\frac{z}{3} + \frac{1}{2}\left(\frac{a}{b}\right)^3 + \frac{1-2\nu}{1+\nu}} \left(\frac{1-2\nu}{1+\nu} + \frac{1}{2} + \frac{6GC}{n_0}\right) - \frac{1-2\nu}{1+\nu} - \frac{1}{2}\left(\frac{b}{a}\right)^3 - \frac{6GC}{n_0} \right) \qquad (9)$$

This equation can be simplified - if the effect of p_i (Which denotes Inner Pressure) is ignored [19]:

(Please note that p_0 denotes External Pressure)

$$\frac{\Delta\lambda}{\lambda} = -\frac{1}{2G} \frac{1}{1-\left(\frac{b}{a}\right)^3} \left(\frac{1}{2}\left(\frac{b}{a}\right)^3 + \frac{1-2\nu}{1+\nu} + \frac{6CG}{n_0} \right) \cdot P_0 \qquad (10)$$

This equation indicates the dependence of WGM shifts on external pressure P_0.

Fig. 13 demonstrates the experimental results of a previous study [16] showing that the WGM resonances shift by applying force on the sphere.

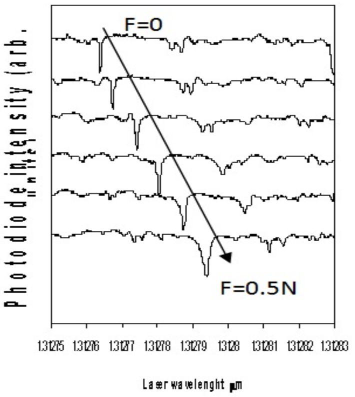


Figure 13. The experimental results can be seen as force terms in the resonance shift [19].

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

As a summary, WGM shifts can now be calculated according to the pressure changes of the medium, or in other words the external pressure changes.

VI. Conclusion and Future Work

In this study, development and verification stages of a new, state of the art, oil level sensor was presented. Finite element analysis and modal analysis results indicate that the new design may be safely used within the oil pan of the engine. Safety factors, 25.76 for fatigue and 2.796 for strength, indicate that the model is ready to initiate production of prototypes for experimental verification purposes.

Research done within the scope of this study initiated an idea for developing a totally new liquid level sensor. The new sensor design is expected to detect pressure/force by using whispering gallery mode (WGM) phenomenon. This attempt is a new perspective into the WGM phenomenon.
For this aim, the experimental set up outlined in this paper will be built and operated in SMU laboratories.

Acknowledgment

Authors would like to thank Torsten Eggers for his valuable comments.

Also, special thanks to Prof. Volkan Otugen for his valuable contributions, and hosting the experiments in Micro Sensors laboratory of Southern Methodist University; lastly Ulas Ayaz (M.Sc.) for his valuable cooperation & contribution.

References

[1] Oil Level Sensors http://www.hella.com/MicroSite/soe/sites/default/files/downloads/J00137_Oelniveausensoren_GB_TT_korr1.pdf> 20.04.2011

[2] Level Measurement <http://www.scribd.com/doc/2836046/Level-measurement-the-basics> 24.04.2011

[3] Level Sensor <http://en.wikipedia.org/wiki/Level_sensor>15.02.2011

[4] A. Kurhani, R. N. Karekar, and R. C. Aiyer, “Liquid level sensor”, University of Prune, India, October 2005

[5] V. E. Sakharov, S. A. Kuznetsov, B. D. Zaitsev, I. E. Kuznetsova, and S. G. Joshi, "Liquid level sensor using ultrasonic lamb wawes", 2002

[6] A. J. Zuckerwar, D. S. Mazel, and Donald Y. Hodges, "Ultrasonic level sensor for liquids under high pressure", 1986

[7] D. D. Hongerholt, Greg A. Seidel, Charles G. HUss, and Eric D. Haugen, "Ultrasonic technology for nonintrusive level measurement on commercial aircraft", 2001

[8] P. Nath, P. Datta, and K. Ch Sarma, "All fiber-optic sensor for liquid level measurement", Gauhati University, India, 2007

[9] M. Bottacini, N. Burani, M. Foroni, F. Poli, and S. Selleri, "All-plastic optical fiber level sensor", University of Parma, 2005

[10] T. Ioppolo, N. Das, and M. V. Ötügen “Species concentration sensor concept based on whispering gallery modes of micro-spheres” 2010, Submitted to J. Appl. Phys

[11] T. Ioppolo, M. V. Ötügen, and K. Marcis “Magnetic field-induced excitation and optical detection of mechanical modes of micro-spheres” 2010, Submitted to J. Appl. Phys

[12] V. P. Stepaniuk, T. Ioppolo, M. V. Ötügen, and V.Sheverev “Attenuation of single-tone ultrasound by an atmospheric glow discharge plasma barrier” 2010, Submitted to J. Appl. Phys

[13] T. Ioppolo, U. K. Ayaz, and M. V. Ötügen, "Tuning of whispering gallery modes of spherical resonators using an external electric field", Optics Express, vol. 17, 19, pp. 16465-16479 (2009).

[14] T. Ioppolo, U. K. Ayaz, and M. V. Ötügen, "High Resolution Force Sensor Based on Morphology Dependent Optical Resonators Polymeric Spheres", J. Appl. Physics, vol. 105, 013535 (2009).

[15] N. Q. Nguyen, N. Gupta, T. Ioppolo, and M. V. Ötügen “Whispering gallery mode-based micro-optical sensors for structural health monitoring of composite materials,” J. Mat. Sci., vol. 44, pp. 1560-1571 (2009) (DOI: 10.1007/s10853-3163-3).

[16] T. Ioppolo, M. Kozhevnikov, V. Stepaniuk, M. V. Ötügen, and V. Sheverev, "A micro-optical force sensor concept based on whispering gallery mode resonators", Appl. Optics, vol. 47 , 3009 (2008).

[17] G. Adamovsky and M. V. Ötügen “Morphology-dependent resonances and their applications to sensing in aerospace environments”, J. Aerosp. Comp. Comm. & Info, Vol. 5, pp. 409-424 (2008).

[18] V. P. Stepaniuk, T. Ioppolo, M. V. Ötügen, and V. Sheverev, "Measurement of gas temperature and convective velocity profiles in a dc atmospheric glow discharge", J. Appl. Phys., vol. 102, 123302 (2007).

[19] T. Ioppolo and M. V. Ötügen, "Pressure tuning of whispering gallery mode resonators" J. Opt. Soc. Am. B, Oct. 2007, vol. 24, 10 (2007).

[20] W. Su, V. Stepaniuk, and M. V. Ötügen, "Demonstration of a Laser Vorticity Probe in Turbulent Boundary Layers" Review of Scientific Instruments, vol. 78, 095106 (2007)

[21] C. Tarau, M. V. Ötügen, V. Sheverev, G. Vradis, and G. Raman, "The effects of thermal barriers on sound wave propagation" International Journal of Aeroacoustics, vol. 6, No. 3, pp. 287-308 (2007)

[22] E. Esirgemez, S. W. Newby, C. Nott, S. Ölçmen, and V. Ötügen, "Experimental study of a round jet impinging on a convex cylinder", Meas. Sci. Tech. vol. 18, pp. 1800-1810 (2007) (doi: 10.1088-0957-0233/18/7/002).

[23] P. Guillaume, "Modal analysis", Department of Mechanical Engineering,Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussel, Belgium.

[24] Product Technical Data <http://www.dwu-unterrichtsmaterialien.de> 10.02.2011

[25] Reinforced PA66 Product Technical Data Sheet Available: <http://kestro.com/pdf/Shanghai%20Kestro%20Polychem,%20Inc.-TDS-A370P.pdf>10.02.2011

[26] HELLA KGaA Hueck & Co. <http://www.hella.com/hella-de-de/index.html>10.01.2011

[27] Modal_analysis <http://en.wikipedia.org/wiki/Modal_analysis_using_FEM> 10.02.2011

[28] The Toyota Website < http://www.toyota.com> 20.12.2011

Localization for Electromagnetic Radio Underwater Sensor Networks

Petar Djukic, Yifeng Zhou, Mylène Toulgoat
Communications Research Centre/Centre de recherches sur les communications
3701 Carling Ave., P.O. Box 11490, Station H, Ottawa, ON, K2H 8S2, Canada
{*petar.djukic,yifeng.zhou, mylene.toulgoat*}*@crc.gc.ca*

***Abstract*—Electromagnetic (EM) radios have been recently proposed for underwater sensor networks. Due to the short transmission range of EM radio transmissions underwater, new localization techniques are required for underwater sensor networks using these radios. We propose and evaluate three localization approaches for EM-based underwater sensor networks: (1) multilateration with the aid of autonomous underwater vehicles (AUVs), (2) multi-dimensional scaling (MDS) without the AUVs, and (3) a hybrid approach combining MDS and multilateration. We compare performance of the three approaches with simulations. The main disadvantage of AUV-aided localization with multilateration is that it does not always provide complete localization coverage of the network. On the other hand, MDS provides complete localization coverage, but its performance decreases in sparse networks. The hybrid approach provides complete network coverage and improves the performance of MDS in sparse networks.**

***Keywords*-Underwater Localization; Multi-dimensional Scaling; Multilateration; AUV-aided Localization.**

I. Introduction

The importance of underwater sensor networks (UWSNs) for environmental monitoring is becoming more important as demonstrated by man-made disasters such as the recent oil spill in the Gulf of Mexico, natural disasters such as Tsunamis, as well as ongoing concerns about climate change. The monitoring capabilities of UWSNs are essential to limit the impact of disasters and to predict the future of Ocean resources. An important service required for effective environmental monitoring is localization of environmental sensors, which ensures that the environmental information can be mapped. Due to the unavailability of the Global Positioning System (GPS) underwater, UWSN localization is a very challenging problem. In this work, we tackle the localization problem for UWSNs using electromagnetic (EM) based underwater radios.

The EM underwater radio is a recent technology, which promises a cost-effective and reliable underwater communications [1]. Current underwater acoustic communication technologies suffer from serious drawbacks: they are unreliable due to their susceptibility to environmental noise and require expensive signal processing to deal with the multipath underwater acoustic channel. Underwater, EM-based signals do not suffer from multi-path arrivals and are not susceptible to environmental noise, so they do not require expensive processing for signal decoding.

The major drawback of using the EM-based radio underwater is its limited communication range due to the rapid attenuation of EM waves in the water. This short communication range requires a large number of nodes to provide connectivity across large areas. However, since the cost of EM-based radios is significantly lower than that of acoustic based radios we do not expect the cost of these networks to be large. In fact, a network with hundreds of EM-based nodes would be less expensive than a network with only a few nodes using acoustic modems.

Localization of nodes in a network is a highly desired capability for sensor network applications. Location stamps are used to tag sensor measurements, ensuring that observations can be mapped to the location where they were taken. Even though UWSNs do not have access to the GPS, localization can be achieved in acoustic-based UWSNs [2]. Due to their short range, localization of EM-based UWSNs needs to be handled differently than localization in acoustic-based UWSNs.

We propose and evaluate three different localization algorithms for UWSNs using EM-based radios:

(i) Multilateration using an autonomous underwater vehicle (AUV) to act as a mobile beacon. The AUV may not reach all nodes, so this localization approach does not in general cover the entire network.

(ii) Multi-dimensional scaling (MDS) using neigbourhood distance information. MDS requires all inter-node distances, which may not be available due to the limited transmission range of the nodes. Missing distances are estimated with a shortest path algorithm, which introduces localization errors.

(iii) A hybrid approach, which uses MDS, but where instead of finding the missing inter-node distances with shortest-path algorithms, the missing distances are calculated from the positions estimated with multilateration. The hybrid approach improves the performance of MDS in sparse networks.

An exhaustive review of localization techniques for acoustic UWSNs is available in [2]. Here we briefly point out why many of these techniques cannot be used for EM-based UWSNs, due to their shorter communication range.

Current localization approaches in underwater acoustic networks rely heavily on the use of anchors [2]. Surface anchors have a known location through the GPS [3]–[5],

ISBN: 978-1-61208-144-1

while dive'n'rise (DNR) anchors get their position on the surface from the GPS and then dive underwater for localization [6], [7]. Underwater anchors get their position from the surface anchors or from the DNR anchors. Given locations of the anchors and distances to the anchors measured by directly communicating with them, UWSN nodes can localize themselves [4]–[7]. For large networks, the anchors may not be able to communicate directly with all nodes. A solution is to use already localized nodes as anchors to localize other nodes in the network, which are then used as anchors, until the entire network is localized [3]. Another approach is to use mobile anchors [8], which can move and reach every node in the network, so that all nodes are covered and localized.

The approaches using direct communication with anchors would not work in EM-based UWSNs since they would require an enormous number of anchors to reach all nodes in a network. In the case of surface anchors, the approach may not work at all if the UWSN is located at a high depth. The incremental approach may also require a large number of anchors to provide desired performance, since the localization errors propagate as new nodes become anchors. Mobile anchors [8] can get closer to the nodes in the UWSN, which is the reason why we propose their use in this work. Moreover, the MDS approach used in this paper is especially good at eliminating the propagation of localization errors in large sensor networks.

II. System Architecture

The underwater network considered in this paper is used for environmental monitoring similar to the floats used in the ARGO project [9]. The nodes are dropped off as a swarm to monitor and collect environmental information. Normally, the nodes dive below the surface of the ocean to collect data. Periodically, they rise to the surface and report their data to the sink over a satellite link or are picked up from the water at the end of the monitoring missions and their information is physically retrieved. In either case, the collected information is used for off-line analysis of a given environmental phenomena.

The network may include a roaming AUV for the purpose of augmenting the localization process. The AUV may also be used for other purposes, such as facilitating delay tolerant networking (DTN) if satellite links are not available [10]. In DTN, the data is picked when the AUV is near a sensor and transmitted to the satellite at a later time by the AUV.

All nodes are assumed to be equipped with EM-based radios for peer-to-peer underwater communications [1]. Since EM waves suffer extremely high attenuation in the underwater medium, the radio range of the nodes is several orders of magnitude shorter than that of an acoustic-based networks. We provide more details on node range in Section IV.

The nodes are required to stamp their data with the location of where the data is collected. As the nodes may be moving due to currents, the location where the information is retrieved (by satellite or by ship) may be very different from the location at the time the environmental data was collected. However, since the collected information is analyzed off-line, the sensor nodes do not actually require the knowledge of their location at the time that observations are made – the location estimates are only needed during the analysis of the observed data. Hence, location estimation can be done off-line, in tandem with the analysis of the observed data. This means that, the localization related information does not need to be sent to a central processor on the fly, even for centralized localization algorithms, significantly reducing the communication cost of sensor localization.

As we show later, geometrical information about the network (distances between nodes and between nodes and the AUV), is sufficient for localizing the nodes. In our system, the sensor nodes tag the observations with the geometrical data instead of estimating their own location and tagging the observations with the estimated location. The observation data and the geometrical data are later transferred to the sink for off-line processing. During off-line processing, the location estimates are obtained from the geometrical data and the observed data is then mapped with the location estimates.

While it may seem that this amount of geometrical data is large (in the order of the number of nodes in the network), in practical situations each node is only connected to a few neighbours at any given time. In our simulations, a typical number of neighbours is only 20 even for networks with several hundred nodes. With the current advancement in computer hardware, this storage requirement can be achieved with off-the-shelf hardware.

III. Localization Algorithms

We now discuss three localization approaches tailored for EM-based underwater networks: multilateration, MDS and a hybrid approach. All three approaches use distance information to localize the nodes. For notational purposes, we assume that there are n nodes in the network and that they are distributed on a plane, i.e., two-dimensional localization is considered. The extension to three dimensions is trivial.

Multilateration uses distance measurements from the AUV to a node i, which are the m distance measurements $d_1^{(i)}, \ldots d_m^{(i)}$ to the AUV and the corresponding set of m AUV positions at which the measurements were taken $\left(x_1^{(i)}, y_1^{(i)}\right), \ldots, \left(x_m^{(i)}, y_m^{(i)}\right)$. MDS uses the set of distances from each node to each of its neighbours. This information is represented by an $n \times n$ Euclidian distance matrix $\boldsymbol{D}$ in which an entry D_{ij} is the distance between nodes i and j. The hybrid approach uses both sets of geographical information: the distance measurements to the AUV and the positions of the AUV, and the Euclidean distance matrix $\boldsymbol{D}$.

ISBN: 978-1-61208-144-1

A. A Two-way Ranging Protocol

Distances between pairs of nodes, or a node and the AUV, are obtained with a two-way ranging protocol. If node i sends a ranging packet at time t_1, which is received by node j at time t_2, and at time t_3 node j responds with a ranging packet received by node i at time t_4, the propagation time can be found with

$$\begin{aligned} t_{ij}^{(prop)} &= \frac{1}{2}\left[(t_4 - t_3) + (t_2 - t_1)\right] \\ &= \frac{1}{2}\left[(t_4 - t_1) + (t_3 - t_2)\right] \end{aligned}$$

and the distance between i and j can be computed with

$$d_{ij} = t_{ij}^{(prop)} v_{EM}$$

where v_{EM} is the propagation speed of electromagnetic waves in the water. Note that node j can be the mobile AUV.

The two-way ranging protocol time-stamps the ranging packets and also sends the clock differences required to find the propagation delays. For example, node j sends the difference $t_2 - t_1$ in its ranging packets, while time-stamping the packet with t_3. The time-stamp and the clock difference is sufficient information for node i to determine the distance to node j. In a two-way ranging exchange with the AUV, the AUV also sends its current location in addition to the time difference.

The exchange of ranging packets is performed periodically in order to provide the current distance information in the presence of node movement. We note that this type of packet exchange is found in networks for synchronization purposes [11]. If the network supports synchronized medium access control, the ranging process comes for ”free” and does not require extra communications between nodes.

Since the ranging packets are time-stamped with the computer time and not the actual time, the distance estimate contains errors due to inaccurate clock reading and clock skew. However, since the propagation speed of EM waves in the water is relatively low (10^5 m/s) and the actual propagation time in EM-based networks is in the order of milliseconds, the clock errors contribute a very small amount of perturbation to the distance estimate (in the order of centimeters).

B. Multilateration from AUV Measurements

The multilateration localizes each individual sensor node using the distance measurements from the node to the AUV. The set of distance measurements to the AUV and the positions of the AUV associated with those measurements are used to form a system of m non-linear equations

$$\left(x_j^{(i)} - x_i\right)^2 + \left(y_j^{(i)} - y_i\right)^2 = \left(d_j^{(i)}\right)^2, 1 \le j \le m$$

where (x_i, y_i) is the unknown position of the node i. In order to localize a node on a plane, $m \ge 3$ distance measurements from non-collinear AUV locations for each system of equations.

The unknown square terms can be eliminated by subtracting the last equation from the first $m-1$ equations to arrive at a linear set of equations for each node

$$\boldsymbol{A}_i \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \frac{1}{2}\boldsymbol{b}_i, \tag{1}$$

where

$$\boldsymbol{A}_i = \begin{bmatrix} (x_1^{(i)} - x_m) & (y_1^{(i)} - y_m) \\ \vdots & \vdots \\ (x_{m-1}^{(i)} - x_m) & (y_{m-1}^{(i)} - y_m) \end{bmatrix}$$

and

$$\boldsymbol{b}_i = \begin{bmatrix} \left(d_m^{(i)}\right)^2 + \left(x_1^{(i)}\right)^2 + \left(y_1^{(i)}\right)^2 \\ \vdots \\ \left(d_m^{(i)}\right)^2 + \left(x_{m-1}^{(i)}\right)^2 + \left(y_{m-1}^{(i)}\right)^2 \end{bmatrix} - \begin{bmatrix} \left(d_1^{(i)}\right)^2 + \left(x_m^{(i)}\right)^2 + \left(y_m^{(i)}\right)^2 \\ \vdots \\ \left(d_{m-1}^{(i)}\right)^2 + \left(x_m^{(i)}\right)^2 + \left(y_m^{(i)}\right)^2 \end{bmatrix}.$$

An estimate of the node position is a least squares solution to (1), which minimizes the Euclidian norm

$$\left\| \frac{1}{2}\boldsymbol{b}_i - \boldsymbol{A}_i \begin{bmatrix} x_i \\ y_i \end{bmatrix} \right\|.$$

This solution is given by

$$\begin{bmatrix} \hat{x}_i \\ \hat{y}_i \end{bmatrix} = \frac{1}{2}\left(\boldsymbol{A}_i^T \boldsymbol{A}_i\right)^{-1} \boldsymbol{A}_i^T \boldsymbol{b}_i. \tag{2}$$

Geographical information collected through the ranging process with the AUV is sufficient for each node to use multilateration to estimate its location without any exchange of information with its neighbours. However, there are several problems with the AUV approach, which may prevent its effective use. First, it is unlikely that the AUV can reach each node in the network from multiple non-collinear locations, meaning that some nodes may not have sufficient number of measurements to localize themselves. Second, since the AUV is underwater, it does not have access to the GPS satellites and must use dead-reckoning techniques to estimate its location. Dead-reckoning only provides a very rough estimate of the AUV’s location and has the disadvantage of error propagation (errors grow with time), which ultimately affects the localization performance. Third, the AUV takes time to visit every node in the network, by which time some nodes may have moved away. So, the location estimates obtained with the AUV may become outdated quickly.

ISBN: 978-1-61208-144-1

C. Multi-dimensional Scaling

The MDS localization algorithm [12] simultaneously finds the position of all of the nodes in the network. Unlike the multilateration approach with the use of AUV, this approach provides complete localization coverage of the network. In addition, MDS is known to be relatively resilient to distance measurement errors due to the over-determined nature of the solution. The output of the algorithm is the estimated relative positions of the nodes

$$\tilde{\boldsymbol{P}} = \begin{bmatrix} \tilde{x}_1 & \tilde{y}_1 \\ \vdots & \vdots \\ \tilde{x}_m & \tilde{y}_m \end{bmatrix}$$

The algorithm works on the full Euclidian distance matrix $\boldsymbol{D}$, which contains the distances between all pairs of nodes. First, the MDS algorithm calculates the square distance matrix $\boldsymbol{\Delta}^{(2)}$ in which each entry corresponds to a square entry in the distance matrix

$$\Delta_{ij}^{(2)} = (D_{ij})^2$$

Then, the MDS algorithm calculates an estimate of the Gram matrix $\tilde{\boldsymbol{B}} = \tilde{\boldsymbol{P}}\tilde{\boldsymbol{P}}^T$ by applying double centering to the square distance matrix

$$\tilde{\boldsymbol{B}} = -\frac{1}{2}\boldsymbol{J}\boldsymbol{\Delta}^{(2)}\boldsymbol{J}, \tag{3}$$

where $\boldsymbol{J} = \boldsymbol{I} - 1/n\boldsymbol{1}\boldsymbol{1}^T$, $\boldsymbol{I}$ is an $n \times n$ identity matrix, and $\boldsymbol{1}$ is an $n \times 1$ column vector of 1s.

Finally, the position matrix is recovered from the Gram matrix with the use of eigenvalue decomposition. Given the eigendecomposition of $\tilde{\boldsymbol{B}}$

$$\tilde{\boldsymbol{B}} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T, \tag{4}$$

where $\boldsymbol{Q}$ is the matrix of eigenvectors and $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues, the estimate of the position matrix is given by

$$\tilde{\boldsymbol{P}} = \boldsymbol{Q}_+\boldsymbol{\Lambda}_+^{1/2}, \tag{5}$$

where $\boldsymbol{Q}_+$ is an $n \times 2$ matrix obtained from $\boldsymbol{Q}$ by retaining the two eigenvectors corresponding to the two largest eigenvalues and $\boldsymbol{\Lambda}_+^{1/2}$ is an $n \times n$ matrix obtained by retaining the columns of $\boldsymbol{\Lambda}$ corresponding to the two largest eigenvalues and taking their square root and making all other entries in the matrix 0.

The last step in the MDS algorithm minimizes the ”strain” error between the position matrix and its Gram matrix [12]

$$\left\| \tilde{\boldsymbol{P}}\tilde{\boldsymbol{P}}^T - \tilde{\boldsymbol{B}} \right\|.$$

The MDS approach assumes that the distance matrix $\boldsymbol{D}$ is fully populated. However, this is only the case if all nodes can communicate directly with each other. In general, the distance matrix is sparse and missing distances should be approximated or estimated. To fill in the missing entries, we use the standard method where a shortest path algorithm [13] estimates the missing distances from available distance measurements. In our simulations we use the all-pair Ford-Fulkerson algorithm to fill in the missing entries in the distance matrix.

D. Hybrid MDS-Multilateration Localization

Even though the MDS algorithm estimates the positions of all nodes in the network, its major drawback is that it relies on estimates of inter-node distances obtained by the shortest path algorithm. When a network is dense and has a regular shape, the shortest path distance corresponds well to its Euclidean distance. However, if a network is sparse or has an irregular shape, a shortest path distance will not match its Euclidean distance, resulting in localization errors.

To improve the performance of MDS localization in sparse UWSNs, we propose a hybrid approach, which combines the multilateration estimates and the MDS algorithm. In the hybrid approach, the position estimates from multilateration are used to calculate the missing inter-node distances in the distance matrix. These estimates have the potential to improve the shortest path estimates as long as the error from multilateration is small.

IV. Simulation Results

To analyze and compare the performance of the proposed localization algorithms, we perform a set of Monte-Carlo simulations using Matlab. In each Monte-Carlo run, sensor nodes are uniformly distributed over a disk with a 1000 m radius. The transmission range of the nodes is assumed to be 100 m, corresponding to the range of EM radio signals in water [1]. The AUV path and the locations of beacon transmissions are the same for every Monte-Carlo run. For each run, we calculate the inter-node distances and the distances from each node to the AUV. We pass the distance smaller than the maximum range to the localization algorithms.

The AUV moves in a spiral pattern with a trajectory given by the coordinates in time $x_{uuv}(t+t_0) = At\cos(t+t_0)$ and $y_{uuv}(t+t_0) = At\sin(t+t_0)$, where $A = 10$ and $-19\pi \leq t \leq 19\pi$ are chosen to ensure that the AUV can visit every node in the network and t_0 is the uniform random variable chosen from the interval $[0, 2\pi]$. The AUV sends out a beacon every 10 s, nodes in the 100 m radius of the AUV at the time of the beacon transmission can perform two-way ranging with the AUV.

Since the AUV moves underwater and only occasionally updates its coordinates using the GPS, most of the time it uses dead-reckoning to determine its coordinates. We model the error due to dead reckoning by nudging the AUV away from its nominal path with a random perturbation. The AUVs coordinates are randomly sampled from a uniform disk with a given radius centered at the point on the AUVs

ISBN: 978-1-61208-144-1

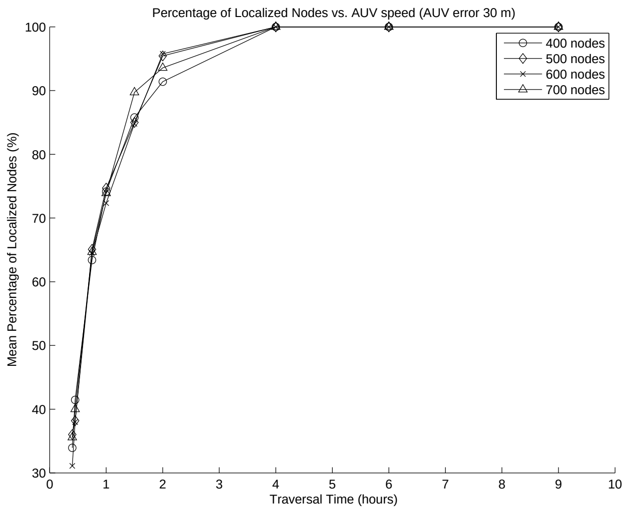


Figure 1. Mobile Beacon Coverage.

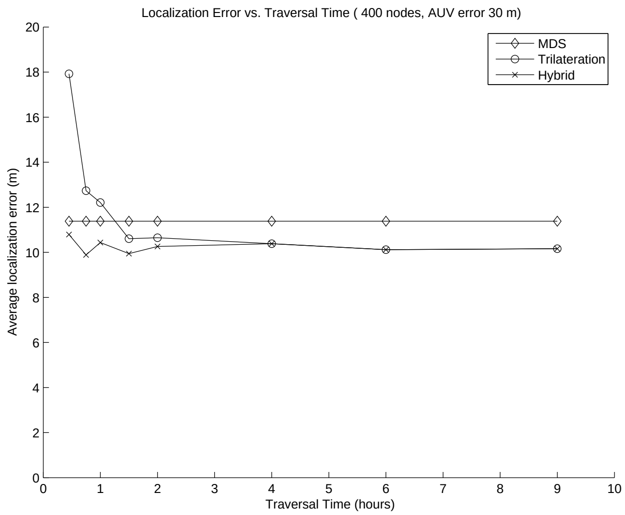


Figure 2. Localization Error (sparse networks)

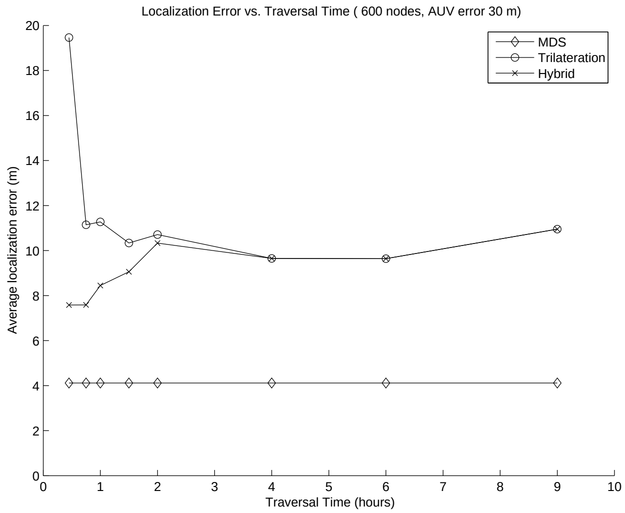


Figure 3. Localization Error (dense networks).

nominal path. This radius of this circle is indicated as ”AUV error” in our plots.

We vary the speed of the AUV to achieve different amount of node coverage. We use speeds from about 10 m/s to about 0.5 m/s, corresponding to the network traversal time from about 30 minutes to about 9 hours, respectivelly. At a high speed the AUV can only send a few beacons before it traverses its entire pre-programmed path. The consequence of high speed then is that many nodes may not receive a sufficient number of beacons to localize themselves. At a lower speed the AUV sends out more beacons while traversing its path, thus increasing the number of nodes that received more than 3 non-collinear beacons.

Figure 1 shows the localization coverage of the mobile beacon approach as the total time to traverse the network (AUV speed) changes. We see that at high speeds the coverage is very low (about 30 %). The coverage can be 100 % at lower speeds, albeit at the cost of longer time to cover the entire network. These results are consistent with previously published results [8] for acoustic based networks. The figure also shows that the coverage does not improve at higher node densities.

Figure 2 and Figure 3 compare the performance of the three localization algorithms. Figure 2 shows the performance for a relatively sparse network (400 nodes), while Figure 3 shows the performance results for a relatively dense network (600 nodes). In both cases, the AUV error is 30 m. The estimated position error is the average across all node errors for all runs.

For the sparse network scenario (Figure 2), we see that multilateration performs better than MDS. However, multilateration cannot localize all nodes for traversal times of less than 4 hours. At higher AUV speeds the nodes that can be localized by multilateration also have fewer ranging measurements with the AUV, than at lower speeds. This decrease in the number of measurements accounts for a larger multilateration error at higher speeds. The hybrid approach is able to localize all nodes, since it uses MDS. In addition, it is able to provide improved localization performance when the AUV is at both high and low speeds. So, using the position estimates from multilateration decreases the error of the MDS position estimates for that scenario.

For the dense network scenario (Figure 3), the MDS approach always outperforms the multilateration approach. The error from estimating inter-node distances with the shortest path algorithm is lower than the error due to the uncertainty of AUV’s location. For higher speeds, the hybrid approach can be thought of as a MDS refinement of the

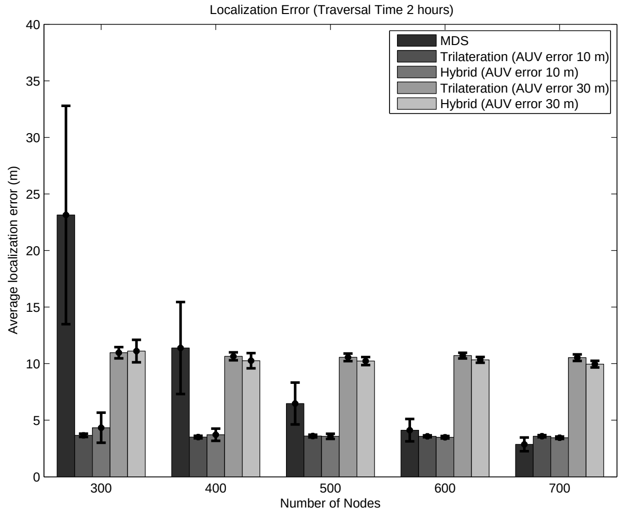


Figure 4. Localization Error (dense networks).

multilateration approach. At lower speeds, the error in the multilateration position estimates affects the performance of the hybrid approach.

Figure 4 compares the performance of the three algorithms for different network sizes and AUV errors, when the AUV traversal time is set to 2 hours. The error bars show the standard deviation of the localization error. The figure shows that MDS performs better in dense networks and that the hybrid approach can improve MDS performance in sparse networks. However, if the error introduced by the AUV is too large, MDS should be used by itself.

V. Conclusion and Future Work

We proposed and analyzed three approaches for localization in EM-based underwater sensor networks. The first approach uses an AUV and a multilateration algorithm. One drawback of this approach is that it may not be able to localize all nodes in the network when the AUV moves at high speeds. The second approach uses MDS to localize nodes based on their neighbourhood inter-node distance measurements. While the MDS approach localizes all nodes in the network, it may suffer from localization errors in sparse networks where not all inter-node distance measurements are available. The performance of MDS degrades in sparse networks because the missing inter-node distance measurements are estimated with a shortest path algorithm. The third approach is a hybrid approach that aims at improving the performance of the MDS by using the position estimates from multilateration to calculate the missing inter-node distances. Our simulations show that the hybrid approach improves the performance of MDS in sparse networks.

References

[1] X. Che, I. Wells, G. Dickers, P. Kear, and X. Gong, “Re-evaluation of RF electromagnetic communication in underwater sensor networks,” *Communications Magazine, IEEE*, vol. 48, no. 12, pp. 143 –151, 2010.

[2] M. Erol-Kantarci, H. Mouftah, and S. Oktug, “Localization techniques for underwater acoustic sensor networks,” *Communications Magazine, IEEE*, vol. 48, no. 12, pp. 152 –158, 2010.

[3] M. T. Isik and O. B. Akan, “A three dimensional localization algorithm for underwater acoustic sensor networks,” *Trans. Wireless. Comm.*, vol. 8, pp. 4457–4463, September 2009. [Online]. Available: http://dx.doi.org/10.1109/TWC.2009.081628

[4] J. Coudeville and H. Thomas, “A primer: using GPS underwater,” *Sea Technology*, vol. 39, no. 4, pp. 31–34, 1998.

[5] L. Collin, S. Azou, K. Yao, and G. Burel, “On spatial uncertainty in a surface long baseline positioning system,” in *Proceedings of the Fifth European Conference on Underwater Acoustics, ECUA 2000*, 2000.

[6] M. Erol, L. F. M. Vieira, and M. Gerla, “Localization with dive’n’rise (DNR) beacons for underwater acoustic sensor networks,” in *Proceedings of the second workshop on Underwater networks*, ser. WuWNet ’07. New York, NY, USA: ACM, 2007, pp. 97–100.

[7] M. Erol, L. Vieira, A. Caruso, F. Paparella, M. Gerla, and S. Oktug, “Multi stage underwater sensor localization using mobile beacons,” in *Sensor Technologies and Applications, 2008. SENSORCOMM ’08. Second International Conference on*, 2008, pp. 710 –714.

[8] M. Erol, L. Vieira, and M. Gerla, “AUV-aided localization for underwater sensor networks,” in *Wireless Algorithms, Systems and Applications, 2007. WASA 2007. International Conference on*, 2007, pp. 44 –54.

[9] J. Gould, D. Roemmich, S. Wijffels, H. Freeland, M. Ignaszewsky, X. Jianping, S. Pouliquen, Y. Desaubies, U. Send, K. Radhakrishnan, K. Takeuchi, K. Kim, M. Danchenkov, P. Sutton, B. King, B. Owens, and S. Riser, “Argo profiling floats bring new era of in situ ocean observations,” *Eos Trans. AGU*, vol. 85, no. 19, pp. 179, 190191, May 2004.

[10] A. McMahon and S. Farrell, “Delay- and disruption-tolerant networking,” *Internet Computing, IEEE*, vol. 13, no. 6, pp. 82 –87, 2009.

[11] D. L. Mills, *Computer Network Time Synchronization: the Network Time Protocol*. CRC Press, 2006.

[12] I. Borg and P. J. F. Groenen, “Classical scaling,” in *Modern Multidimensional Scaling*, ser. Springer Series in Statistics. Springer New York, 2005, pp. 137–168.

[13] Y. Zhou and L. Lamont, “Optimal local map registration for wireless sensor network localization problems,” *Lecture Notes in Electrical Engineering*, vol. 64 LNEE, pp. 177–198, 2010.

ISBN: 978-1-61208-144-1

Experimental Assessment of Adaptive Spatial Combining for Underwater Acoustic Communications

Guosong Zhang
Dept. of Elec. and Telecomm.
Norwegian Univ. of Sci. and Tech.
7491 Trondheim, Norway
Guosong.Zhang@iet.ntnu.no

Jens M. Hovem
SINTEF ICT
Norwegian Univ. of Sci. and Tech.
7491 Trondheim, Norway
Jens.Martin.Hovem@sintef.no

Hefeng Dong
Dept. of Elec. and Telecomm.
Norwegian Univ. of Sci. and Tech.
7491 Trondheim, Norway
Hefeng.Dong@iet.ntnu.no

Abstract— **This paper presents a receiver structure of passive-phase conjugation (PPC) based adaptive spatial combining for underwater acoustic communications. Based on temporal diversity exploited by PPC processing, the presented structure exploits spatial diversity by adaptive multichannel combining. The structure was assessed in two field experiments which were conducted in two different seasons in Trondheim harbor. The experiments were carried out with the same configuration, but the channel physics were different due to different environmental conditions. A cross receiving array of 10 hydrophones, deployed in a water depth of 10 m, was used to collect the received waveform in a range of 2.0 km. By off-line signal processing, we have demonstrated the superior performance in exploiting spatial diversity for underwater communications.**

Keywords- adaptive spatial combining; passive time reversal; passive-phase conjugation; underwater communication;spatial diversity.

I. INTRODUCTION

Underwater acoustic communication is frequently limited by intersymbol interference (ISI) due to a time-varying extended multipath structure in an acoustic channel [1]. Time delayed arrivals due to multipath propagation can be spatial and temporal refocused by time reversal method [2, 3], which mitigates ISI caused by multipath. After the equal weight multichannel combining, there is only one channel equalizer required by passive time reversal communications [4, 5], which can be realized by passive-phase conjugation (PPC). PPC achieves pulse compression for the time delayed arrivals [6], and this property is used for underwater acoustic communications with a reduced number of taps for adaptive channel equalization.

With multiple receivers distributed in space, there is spatial diversity which could be used to avoid deep fading in communications [7]. It is necessary to use multiple receivers in a time-varying acoustic channel to achieve stable communications. Since ISI cannot be eliminated by refocusing, an adaptive channel equalizer is required. It is known that a decision feedback equalizer (DFE) removes residual ISI. Spatial diversity is exploited by passive time reversal to mitigate ISI [8], where the side lobe level of $q(t)$ -function is reduced with an increased number of receivers. Multichannel combining is performed by passive time reversal to achieve refocusing at the receiving array. In a real ocean environment, since there are interchannel correlations among the receivers [9], the diversity combining of passive time reversal does not take spatial coherence into account.

In this paper, a receiver structure which is a PPC based multichannel equalization scheme is presented, where adaptive spatial combining is conducted to exploit spatial diversity. The structure takes advantage of pulse compression, where the number of taps for a multichannel equalizer is significantly reduced. As discussed by Yang [9], there is lack of model which can precisely predict the channel characteristics in a real ocean, and therefore the receiver structure is assessed by experiments. Two sea experiments were conducted to assess the receiver structure, where a receiving array of 10 hydrophones was used, and the received waveforms were recorded for off-line processing in the laboratory. The structure of PPC based multichannel DFE (PPC-McDFE) is used in digital signal processing, and it improves performance of the passive time reversal structure which is realized by PPC plus a signal channel DFE (PPC-DFE). The improvement is shown in terms of output signal-to-noise ratio (SNR).

This paper is organized as follows. The proposed receiver structure of PPC-McDFE is introduced in Section II, and the diagram shows its differences from PPC-DFE. Section III describes the experimental setup. The information about experimental area and sound speed profiles are shown, and the signal transmission is introduced. In Section IV, the channel measurements and communication results are presented and discussed. Conclusions are given in Section V.

II. THE RECEIVER STRUCTURE

The communication information consists of a sequence of symbols denoted as $I[n]$, and each symbol occupies a duration of T . The baseband data signal $s(t)$ can be expressed as

$$s(t) = \sum_{n} I[n] g(t - nT) \tag{1}$$

where $g(t)$ is the pulse shape function for each symbol such that

Copyright (c) IARIA, 2011.   ISBN: 978-1-61208-144-1

$$g\left(\tau\right)=\begin{cases}1, \text{for } 0\le\tau<T\\ 0, \text{otherwise}\end{cases}. \qquad (2)$$

At the *i*th receiver, the received signal $r_i\left(t\right)$ can be written as

$$r_i\left(t\right)=h_i\left(t\right)\otimes s\left(t\right)+w_i\left(t\right) \qquad (3)$$

where $h_i\left(t\right)$ represents the channel impulse response, $w_i\left(t\right)$ is a band-limited noise, and ⊗ denotes convolution. PPC processing can be seen as match-filtering of the received data signal using a channel response. The output of PPC is expresssed as

$$\begin{aligned}z_i\left(t\right)&=h_i\left(-t\right)\otimes r_i\left(t\right)\\&=h_i\left(-t\right)\otimes\left(h_i\left(t\right)\otimes s\left(t\right)+w_i\left(t\right)\right)\\&=q_i\left(t\right)\otimes s\left(t\right)+\varsigma_i\left(t\right),\end{aligned} \qquad (4)$$

where $q_i\left(t\right)$ is the autocorrelation function of $h_i\left(t\right)$, and $\varsigma_i\left(t\right)$ is a filtered noise. The main lobe width and side lobes of $q_i\left(t\right)$ are determined by the channel physics, and ISI caused by $q_i\left(t\right)$ exists after pulse compression.

ISI can be mitigated by refocusing time delayed arrivals at a receiving array, where one channel output is obtained by passive time reversal. The output of diversity combining over *K* receivers is

$$\begin{aligned}z\left(t\right)&=\sum_{i=1}^{K}h_i\left(-t\right)\otimes r_i\left(t\right)\\&=\sum_{i=1}^{K}q_i\left(t\right)\otimes s\left(t\right)+\sum_{i=1}^{K}\varsigma_i\left(t\right)\\&=q\left(t\right)\otimes s\left(t\right)+\varsigma\left(t\right)\end{aligned} \qquad (5)$$

where $q(t)$ is a function of autocorrelations summed over the *K* receivers, and $\zeta(t)$ is a filtered noise. Residual ISI caused by $q(t)$ can be removed by an adaptive channel equalizer [5]. Figure 1 shows the receiver structure for communications using passive time reversal, and it is realized by PPC plus a single channel DFE (PPC-DFE). Carrier-phase tracking is realized using the technique of a second order digital phase-locked loop (DPLL) proposed by Stojanovic [10], and the DPLL tracks the mean frequency shift of *K* receivers. Based on the minimum mean square error (MSE) criterion, the coefficients for the feed-forward and feedback filters are obtained by the recursive least squares (RLS) algorithm [11]. The output error is defined as

$$e\left[n\right]=I\left[n\right]-\tilde{I}\left[n\right] \qquad (6)$$

where $\tilde{I}\left[n\right]$ is the estimate. Out the training mode, $I\left[n\right]$ is replaced by the decided output $\hat{I}\left[n\right]$.

The receiver structure PPC-McDFE is shown in Figure 2, and it takes advantage of pulse compression in order to reduce the number of taps for the multichannel DFE. For the K receivers, independent DPLLs compensate the frequency shifts, and the RLS algorithm updates coefficients of the feed-forward filters. This structure is designed to improve the performance of PPC-DFE by adaptive multichannel combining. As shown in Figure 1, one channel output exploits spatial diversity by low-complex combining without considering both the interchannel correlations and the output errors. For the structure shown in Figure 2, spatial diversity is exploited by performing adaptive spatial combining. Based on a previous output MSE, the RLS algorithm updates the filter tap coefficients for combining in order to minimize current output MSE. PPC-McDFE achieves superior performance in the trials.

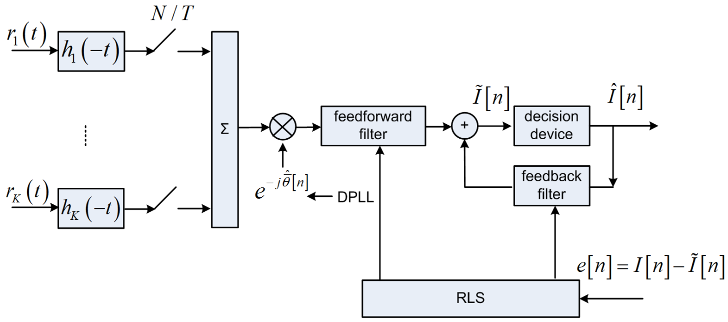


Figure 1. The diagram of PPC-DFE. There are *N* samples per symbol in baseband digital signal processing.

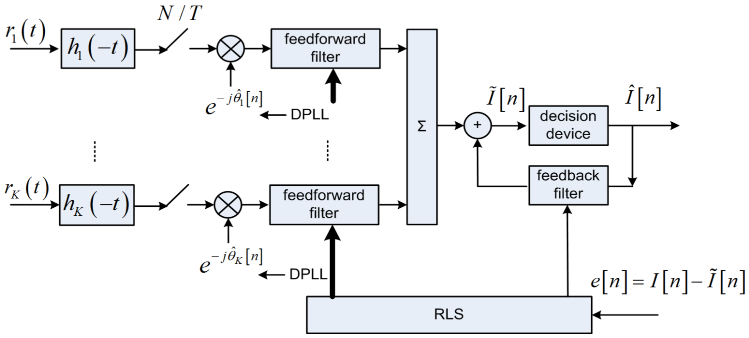
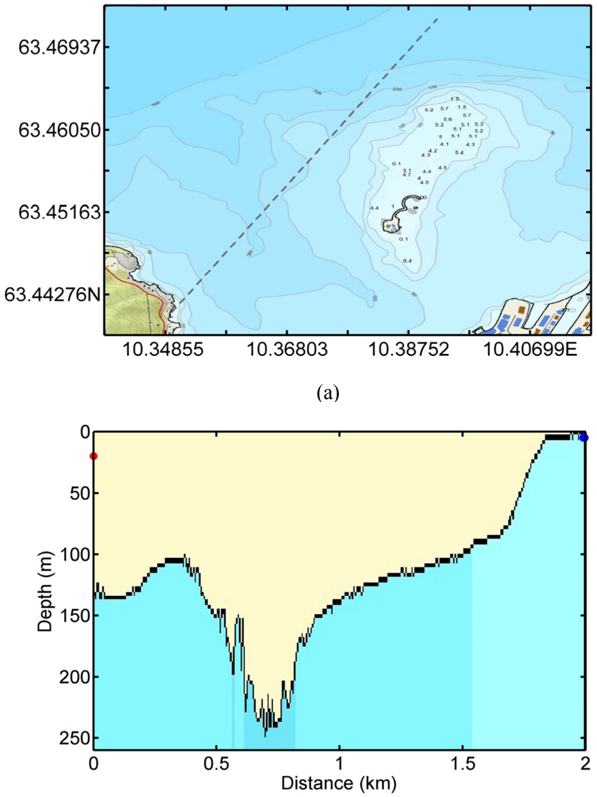


Figure 2. The diagram of PPC-McDFE.

III. The Experiments

The two experiments were carried out on June 30 (Trial A) and September 9 (Trial B), 2010, in Trondheim harbor in Norway shown in Figure 3(a). In the experimental area, the shallow littoral region less than 20 m extends about 100 m off shore. The instrumentation and the source sound level were the same for both Trial A and Trial B. Figure 3(b) shows that the depth profile from the source to the CRA changes from 240 m to 10 m. The red dot denotes the position of the transmitter in a distance of 2 km to the CRA, and the transmitter used a hemispherical acoustic transducer suspended at a depth of 20 m from the NTNU research vessel R/V Gunnerus. The CRA is near-shore deployed in a water depth of 10 m, and it consists of a vertical receiving array of 6 hydrophones with 1 m element spacing and a horizontal receiving array of 4 hydrophones with 1.5 m element spacing. The dynamic positioning system of R/V Gunnerus was activated to reduce drifting.



(a)

(b)

Figure 3. (a) Experimental area in Trondheim harbor. The dash line is the communication direction. (b) The depth profile in the communication direction. The red dot on the left denotes the transmitter position, and the blue dot on the right denotes the receiving array.

The sound speed profiles measured during the trials are shown in Figure 4. There was a sound channel at the depth of 25 m in Trial A. In Trial B, there was a negative gradient sound speed profile causing the acoustic energy emitted by the source bent towards the sea bottom. The receiving array was deployed at a depth which was different from the transmitter depth, and the received signal exhibited time-varying fading due to reflections from the sea surface and bottom.

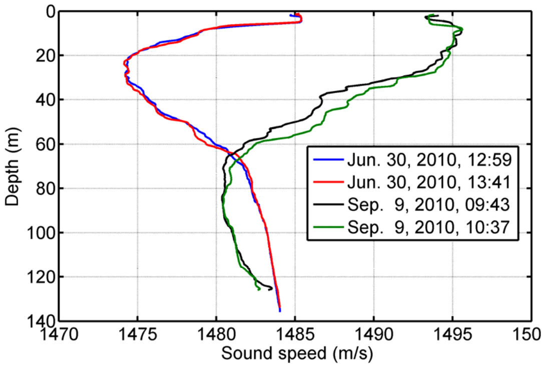
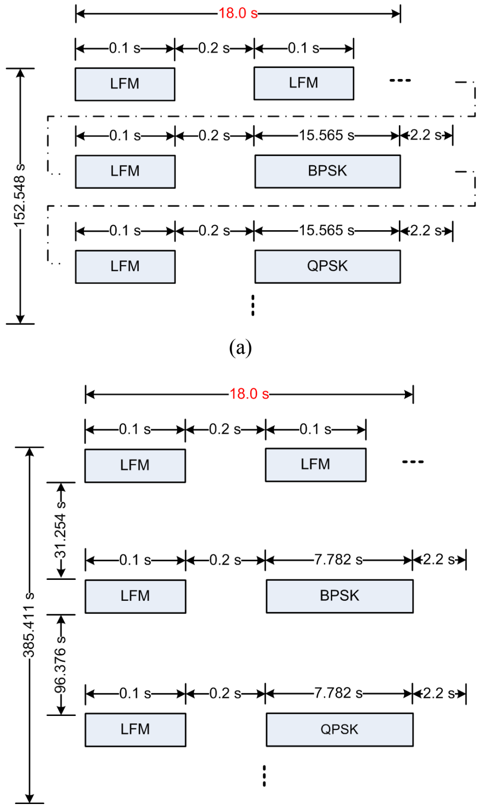


Figure 4. Sound speed profiles measured by Gunnerus.

The carrier frequency of the transmitted signals was 12 kHz. The channel probe signal was a linear frequency modulation (LFM) chirp of 0.1 s with a Hanning window, and its effective bandwidth was 2.2 kHz. The chirp was also used as a pulse-shaping signal for the data signals, which were generated by binary phase shift keying (BPSK), quadrature phase shift keying (QPSK) modulations. The symbol rate in communications was R=1/T=1 kilosymbols/s.



(a)

(b)

Figure 5. One period of the transmitted signal. (a) Trial A. (b) Trial B.

Figure 5(a) shows one period of transmitted signal for Trial A, where the signals were repeatedly transmitted 18 times. Figure 5(b) shows one period of transmitted signal for Trial B, where the signals were repeatedly transmitted 6 times. The 60 chirps spanning 18 s in each period were used to measure the channel response. The received waveforms were recorded with a sampling frequency of 96 kHz for off-line processing in the laboratory.

IV. THE RESULTS AND ANALYSIS

The parameters for PPC-DFE and PPC-McDFE are given in TABLE I. In baseband digital signal processing, the feed-forward filter taps span 4 symbol intervals with an over sampling rate of N=2. As suggested in [12], the second order DPLL is configured with K_2=K_1/10 to get good performance.

TABLE I. RECEIVER PARAMETERS

Parameters	Description	Value
F_s	The sampling frequency at the CRA	96 kHz
f_c	Carrier frequency	12 kHz
R	The symbol rate	1000 symbols/s
N	Over sampling factor	2
N_{ff}	The number of feed forward filter taps	8
N_{fb}	The number of feedback filter taps	2
N_t	The number of training symbols	72
λ	RLS forgetting factor	0.999
K	The number of receiving channels	10
K_1	Proportional tracking constant in PLL	0.01
K_2	Integral tracking constant in PLL	0.001

A. *Trial A*

In each period, the channel response within 18 s can be measured by the 60-chirp using the method of replica correlation. Figure 6 shows the overview of channel response measurements in hydrophone No. 4, which was located at a depth of 4.5 m above the sea bottom. The temporal compression is observed in the 45-minute trial, the time delayed arrivals in 18 periods are not aligned in delay time, and the first arrivals in the following periods approach the receiving array earlier than those in the previous periods. The compression rate changes with period, and the variation is caused by time-varying Doppler shift due to relative movement between the transmitter and the receiving array [13]. The channel response changes with time, it changes with receivers, and spatial diversity in the channel can be exploited to achieve stable communications.

Figure 7 shows the receiver performance in the first frame versus the number of hydrophones, where the output SNR is calculated by (1-MSE)/MSE [7]. The performance increases with the number of receiving channels, where spatial diversity is exploited, and PPC-McDFE achieves superior performance over PPC-DFE using a small number of receivers. In maximum, an improvement gain of 4.5 dB is obtained where the signals of 7 receiving channels are used for processing. In the following analysis, 10 receiving channels are used in signal processing to assess the improvement in terms of output SNR.

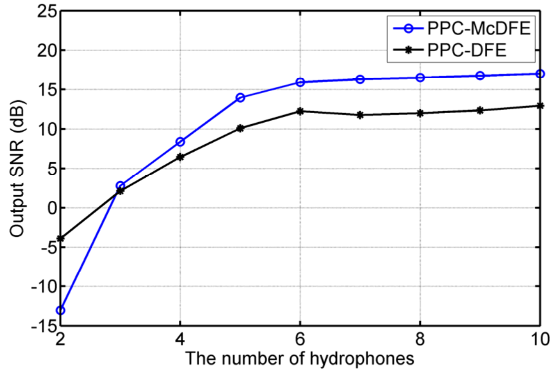


Figure 7. The performance for BPSK in terms of output SNR as a function of the number of receiving hydrophones.

In the 45-minute trial, Figure 8 shows that the receiver performance changes with time, and PPC-McDFE achieves better performance than PPC-DFE. Since the receivers were deployed in a region of reverberation, input SNR varied with time. It is shown in Figure 6 that magnitude of the arrivals changes with time in the trial. It is calculated that the improvement gain for BPSK communication changes from 2.2 dB to 6.7 dB and the improvement gain for QPSK communication changes from 2.5 dB to 6.4 dB. The reason for the improvement variation is that interchannel correlations exist and change with time.

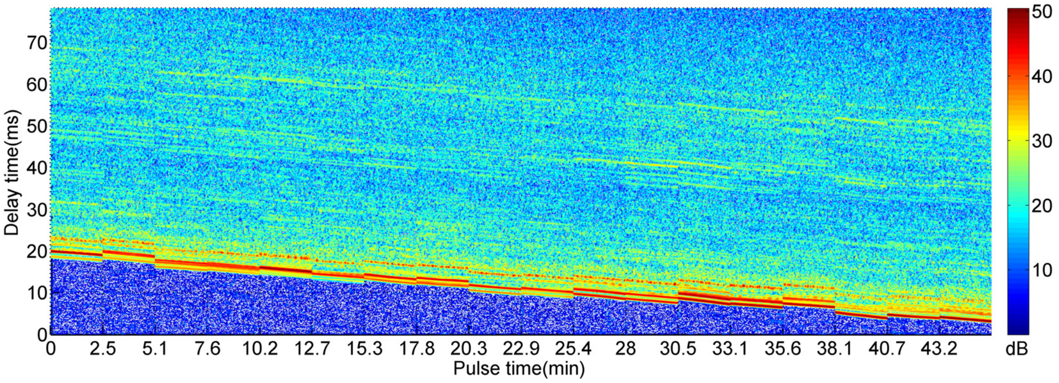
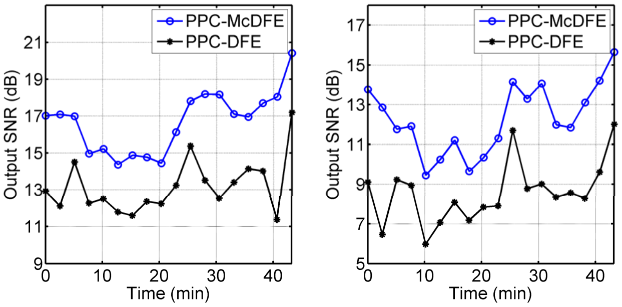


Figure 6. The channel impulse response measurements during the 45-minute experiment.



(a) (b)

Figure 8. The performance in terms of output SNR. (a) BPSK. (b) QPSK.

Spatial coherence measures the strength of interchannel correlation between two receivers, and it can be calculated by

$$\psi(m,n)=\frac{\left|r_m(-t)\otimes r_n(t)\right|_{\max}}{\sqrt{\left|r_m(-t)\otimes r_m(t)\right|_{\max}\left|r_n(-t)\otimes r_n(t)\right|_{\max}}}\ ,\ (7)$$

where $\left|r_m(-t)\otimes r_n(t)\right|_{\max}$ denotes the maximum absolute value of cross-correlation between the two signals $r_m(t)$ and $r_n(t)$, and $r_m(t)$ denotes the signal received by the mth hydrophone. Figure 9 shows the spatial coherence measurement in the 45-minute trial, and interchannel correlation exists and changes with time during the trial. Since it is infeasible to predict the spatial coherence variations in a real ocean environment [9], it becomes intractable to preset weights for the multichannel combining. PPC-DFE assumes that there are independent receivers of the receiving array, where the spatial diversity is exploited by combining prior to the adaptive channel equalization, and this assumption is impacted by the interchannel correlations. PPC-McDFE achieves superior performance which is attributed to the adaptive spatial combining, as the multichannel combining weights are updated based on output MSEs.

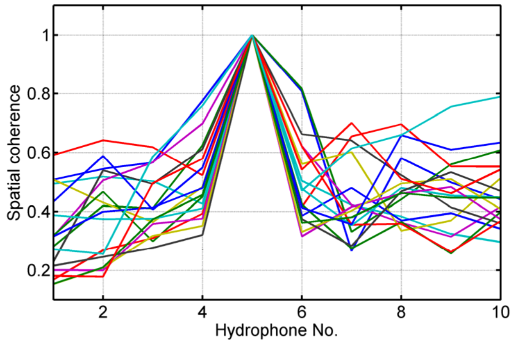
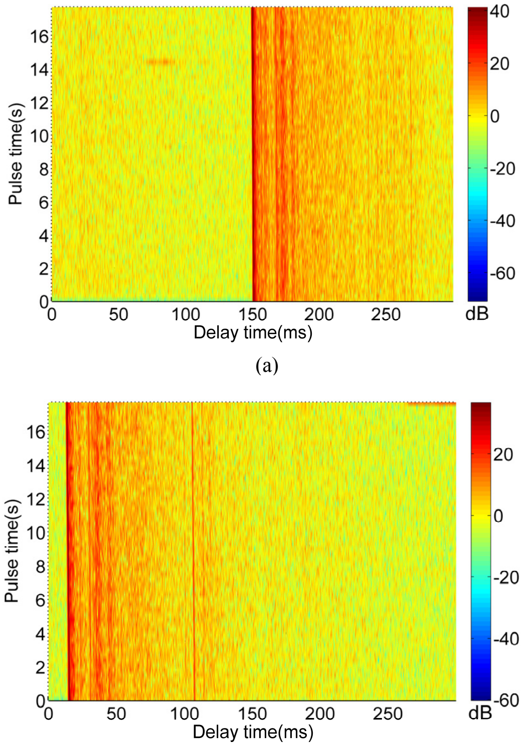


Figure 9. The spatial coherence as a function of the hydrophone index.

B. *Trial B*

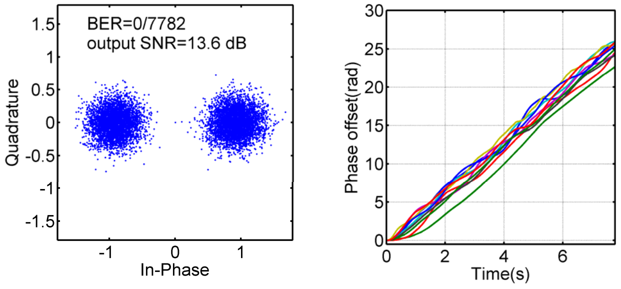
Due to a negative gradient sound speed profile shown in Figure 4, extended time delayed arrivals caused by reflections are observed in Figure 10. The maximum magnitude decreases by 10 dB in the second period, and the compression due to Doppler is observed, where the compression rate is about 3.2e-4 between the two periods. Figure 11 shows the BPSK performance in the first period. There is no bit error of 7782 bits with an output SNR of 13.6 dB, and there is little difference in terms of the slope of phase offset from the DPLLs. This improvement gain is obtained from the adaptive spatial combining. The independent DPLLs are replaced by a common DPLL in the following processing



(a)

(b)

Figure 10. The impulse response measurements in two continuous periods within 385.411 s. (a) The first period. (b) The second period.



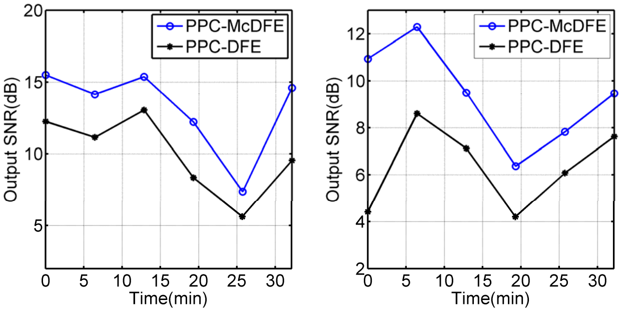
(a) (b)

Figure 11. The receiver performance of PPC-McDFE. (a) The scatter plot. (b) The output of DPLL.

A single DPLL can update an averaged value of the phase-offset corrections among the 10 receivers, and it is given by

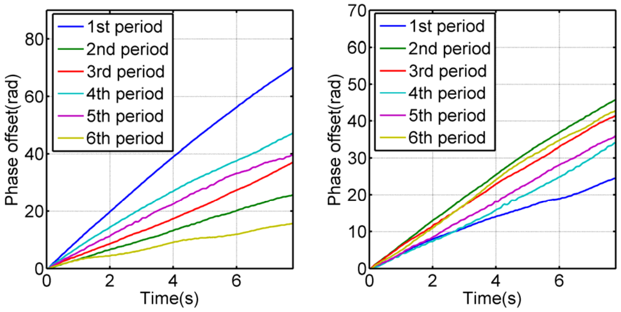
$$\bar{\hat{\theta}}[n] = \frac{1}{10}\sum_{i=1}^{10}\hat{\theta}_i[n] \qquad (8)$$

where $\hat{\theta}_i[n]$ denotes the phase offset estimate from ith DPLL. Figure 12 shows the performance in 6 periods, and improvement of passive time reversal communications is obtained. In calculation, the improvement gain changes from 1.7 dB to 5.1 dB for BPSK, and it changes from 1.8 dB to 6.5 dB for QPSK. The improvement gain is attributed to spatial diversity exploited by the adaptive combining. Figure 13 shows the single DPLL output in 6 periods. For the BPSK signals, the slopes of phase offset change from 2.1 rad/s to 9.0 rad/s, where the equivalent carrier frequency shifts vary from 0.3 Hz to 1.4 Hz. The equivalent carrier frequency shifts for the QPSK signals change from 0.5 Hz to 0.9 Hz. In a time-varying acoustic channel, it is necessary that carrier-phase tracking by DPLL is implemented to compensate Doppler shift for underwater communications.



(a) (b)

Figure 12. The performance in terms of output SNR. (a) BPSK. (b) QPSK.



(a) (b)

Figure 13. The single DPLL output. (a) BPSK. (b) QPSK.

V. CONCLUSIONS

The presented receiver structure which is based on pulse compression is assessed in two sea trials over a range dependent channel of 2 km. The presented results have shown that improvement of passive time reversal communications can be obtained by using adaptive spatial combining to exploit spatial diversity. In a real ocean, since the receivers of a receiving array are unable be distributed independently in space, there is no model which can predict time-varying interchannel correlations, and it is preferred that spatial diversity is exploited by the adaptive combining where the combining coefficients are updated to minimize output MSE. With a receiving array of a small number of hydrophones, we have demonstrated the performance of PC-McDFE.

ACKNOWLEDGMENT

The author would like to thank PhD students Zhongxi Chao, Yan Jiang, technical engineer Tim Cato Netland and the crew of the R/V Gunnerus for the help in conducting the experiments.

REFERENCES

[1] D. B. Kilfoyle and A. B. Baggeroer, "The state of the art in underwater acoustic telemetry," *IEEE J. Ocean. Eng.*, vol. 25, pp. 4-27, 2000.

[2] D. Rouseff, D. R. Jackson, W. L. J. Fox, C. D. Jones, J. A. Ritcey, and D. R. Dowling, "Underwater acoustic communication by passive-phase conjugation: theory and experimental results," *IEEE J. Ocean. Eng.*, vol. 26, pp. 821-831, 2001.

[3] G. F. Edelmann, T. Akal, W. S. Hodgkiss, K. Seongil, W. A. Kuperman, and S. Hee Chun, "An initial demonstration of underwater acoustic communication using time reversal," *IEEE J. Ocean. Eng.*, vol. 27, pp. 602-609, 2002.

[4] T. C. Yang, "Correlation-based decision-feedback equalizer for underwater acoustic communications," *IEEE J. Ocean. Eng.*, vol. 30, pp. 865-880, 2005.

[5] H. C. Song, W. S. Hodgkiss, W. A. Kuperman, M. Stevenson, and T. Akal, "Improvement of time-reversal communications using adaptive channel equalizers," *IEEE J. Ocean. Eng.*, vol. 31, pp. 487-496, 2006.

[6] D. R. Dowling, "Acoustic pulse compression using passive phase-conjugate processing," *J. Acoust. Soc. Am.*, vol. 95, pp. 1450-1458, 1994.

[7] J. G. Proakis, *Digital communications*. New York: McGraw-Hill, 2001.

[8] H. C. Song, W. S. Hodgkiss, W. A. Kuperman, W. J. Higley, K. Raghukumar, T. Akal, and M. Stevenson, "Spatial diversity in passive time reversal communications," *J. Acoust. Soc. Am.*, vol. 120, pp. 2067-2076, 2006.

[9] T. C. Yang, "A study of spatial processing gain in underwater acoustic communications," *IEEE J. Ocean. Eng.*, vol. 32, pp. 689-709, 2007.

[10] M. Stojanovic, J. Catipovic, and J. G. Proakis, "Adaptive multichannel combining and equalization for underwater acoustic communications," *J. Acoust. Soc. Am.*, vol. 94, pp. 1621-1631, 1993.

[11] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, New Jersey: Prentice Hall, 2001.

[12] M. Stojanovic, "Efficient processing of acoustic signals for high-rate information transmission over sparse underwater channels," *Physical Communication*, vol. 1, pp. 146-161, 2008.

[13] B. S. Sharif, J. Neasham, O. R. Hinton, and A. E. Adams, "A computationally efficient Doppler compensation system for underwater acoustic communications," *IEEE J. Ocean. Eng.*, vol. 25, pp. 52-61, 2000.

Turning submarine telecommunications cables into a real-time multi-purpose global climate change monitoring network

Yuzhu You
Institute of Marine Science
University of Sydney
NSW 2006, Australia
yuzhu.you@gmail.com

Bruce Howe
Department of Ocean and Resources Engineering
University of Hawaii at Manoa
Honolulu, HI 96822 , USA
bhowe@hawaii.edu

***Abstract*— Climate change impacts polar water mass formation and its subsequent sinking and spreading along the ocean floor, thus affecting oceanic storage of greenhouse gases. Currently, there is no low cost way to monitor the polar and other abyssal water masses. Here we propose to use submarine cables by integrating sensors and node functionality into repeaters to form a real-time global climate change monitoring network that also includes tsunami warning capability and provides for multipurpose connections.**

Keywords- Submarine cables; Cable repeaters; Climate change monitoring; Global climate monitoring network, Sensors; Node functionality.

I. INTRODUCTION

Oceans govern climate and climate variability since they store more than 90 per cent of the heat and 50 times as much carbon as the atmosphere in the Earth's climate system. Global warming causes polar ice to melt and sea-level to rise, consequently reducing the ocean's capacity for greenhouse gas storage in the abyssal ocean and further reinforces atmospheric greenhouse warming. The deepest water mass covering the world's ocean floor originates from polar regions such as the Greenland Sea, Labrador Sea, Weddell Sea, Ross Sea and around Antarctica. These polar water masses have the highest densities in the world ocean. The densest water plunges along the polar continental slopes and spreads along the ocean floor to fill up the global ocean bottom waters (Figure 1). While doing so, it carries a large amount of atmospheric greenhouse gasses into the deep ocean with residence times on the order of a millennium.

Temperature and salinity are basic state variables of ocean climate. They govern the density and thus, along with wind and solar forcing, the overall circulation of the world ocean. The polar bottom water is formed at the surface of polar seas through air-sea interaction as warm, salty water is cooled and sinks. This process is being affected by climate change, impacting water temperature and salinity as a function of time and space. Consequently, this would affect the formation of the oceans' bottom water and its volume transport, and thus overall ocean circulation. Under a warming environment, polar water masses can absorb less gas and are less capable of sinking. Consequently, the transport decreases and less atmospheric greenhouse gases are brought into the deep ocean. If the ocean-bottom water temperature, salinity and pressure could be measured on the ocean floor at many locations including "choke points", changes of the planet's climate could be monitored in this crucial part of the Earth climate system. One of the major results from the recent OceanObs09 conference 21-25 September 2009 in Venice, Italy (http://www.oceanobs09.net), was the need for abyssal measurements below 2000 m since the current Argo program can measure temperature and salinity only above 2000 m.

It is generally understood that climate change monitoring requires very long-term observations from decades to centuries. These observations need sustainable technical and financial support. Currently, there is no low cost way to effectively monitor the ocean climate in the long run. Particularly, the high pressure at abyssal depth (~6000 m) and complicated bottom topography cause instrumentation to be extremely difficult and costly. In traditional oceanographic measurements from ships, the sea bottom is intentionally avoided for possible damage of the instruments in case they hit the bottom. Due to its vast extent and volume, the measurement of the layers of polar water mass on and just above the ocean floor is virtually nonexistent. However, submarine telecommunications cables typically with repeaters (optical amplifier) 40-120 kilometers apart, lie on the ocean floor and can fill this gap (Figure 1). They can be used to measure the bottom water flowing by on a continuous, sustained basis for many decades – something that cannot be done by other means. At the same time, electric signals from the cables can yield information about the ocean currents crossing the cables and as well as average temperature. Importantly, the cables and the repeaters can provide power to transmit data from general purpose observatory nodes on the sea floor.

The rest of the paper is organized as follows. From a historical appoint of view, Section 2 emphasizes the importance of turning submarine telecommunications cables into a real-time multipurpose global climate change monitoring network. Such an opportunity was missed and should not be missed again. Section 3 calls for facilitating the usage of retired and in-service submarine cables for ocean climate monitoring. This section also describes the technical feasibility of designing future generation of cables and repeaters, which enable climate measuring sensors to be integrated into repeater housing and to allow for node functionality. The technologies have been developed in oil/gas industry and ocean observatory in the last decade and are ready to be implemented into the global climate monitoring network. Perspective of realizing the network is given in Section 4 to meet the demand for real-time global tsunami warning, sea-level rise observation and climate change monitoring.

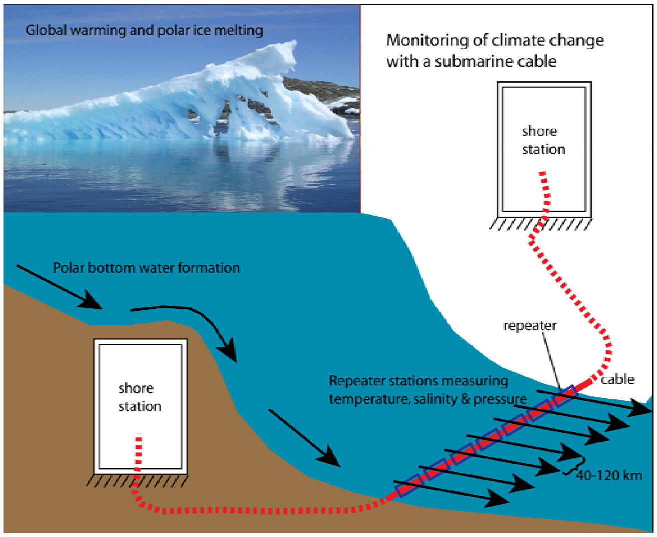

Global warming and polar ice melting
Monitoring of climate change with a submarine cable
shore station
Polar bottom water formation
repeater
cable
shore station
Repeater stations measuring temperature, salinity & pressure
40-120 km

Figure 1. Schematic of polar water formation. It sinks and spreads across the ocean floor, which is affected by global warming and ice melting at its polar source formation region. The change of water properties such as temperature and salinity can be measured with sensors installed in the repeaters (optical amplifiers) of a submarine cable [1] [2].

II. A MAJOR OPPORTUNITY

Since 1850, over a million kilometres of submarine cables have been laid on the ocean floor, covering a significant part of the global oceans [2]. Thousands of repeaters in these cable systems are currently serving only the single purpose of amplifying communication signals (Figure 2). Slight modification of these repeaters – plugging in only one pressure sensor into their housing, for example – could turn the single purpose telecommunication network into multipurpose, real-time global tsunami warning and sea-level rise monitoring network [4] [1]. Such a global network could quickly locate the source of the tsunamis triggered by an earthquake or other factors from most of the ocean floor. The very dense repeater stations – typically about 100 repeaters for a transatlantic cable and 200 repeaters for a transpacific cable – can provide tsunami-wave travel times, amplitudes and speeds for early warning in real time. Such a network can be sustained for decades at low cost. The long-term bottom pressure time series data has also important scientific value for oceanographers to study tides far from land [5].

If more sensors – such as sensors for measuring temperature and salinity – are installed with the repeaters, the telecom cables can be turned into a much-needed real-time global climate change monitoring network. Those repeaters can also be used as nodes for multipurpose connections, using, for instance, inductive power and communications transfer through the repeater pressure case. In the long run, a robust design enables even more channels to measure additional climate data such as ocean currents (using an up-looking acoustic Doppler ocean current profiler), oxygen, greenhouse gasses, seismicity, large-scale temperature using acoustic tomography, geophysical and biochemical properties, and underwater video and acoustics. This very useful resource has been overlooked. This is a major opportunity that should be taken advantage of.

III. PLUG SENSORS INTO NEW GENERATION OF CABLE REPEATERS

At present, only a tiny fraction of the existing cables (about a thousand kilometres or 0.1 per cent) is used for measuring climate change data such as the Florida Current cable that has been used to take daily measures of the volume of water transported by the Florida Current over the past 25 years (derived from the voltage fluctuations measured at the shore station; [4] [2]; [6]). In particular, repeaters have not been modified for this potential of climate monitoring capacity. This is a significant opportunity for telecommunication companies to design the new generation of cable repeaters and to provide additional climate data to stakeholders other than their usual telecoms services. The new repeaters that are integrated with integrated sensors and nodes will enable us to measure the major climate variables of temperature, salinity and pressure and in a cost-effective long-term climate change monitoring network. The designing cost of the new generation of repeaters is estimated at several to ten million US dollars [7]. This non-recurring engineering cost is regarded as modest when amortized over thousands of units for many years to come. Telecom companies can make additional profit by serving the additional stakeholders (e.g., government science agencies) and the broad community with very much needed climate data and relevant products.

Modern technologies enable the new type of repeaters to be assembled into one cable body but two operation systems, one for telecommunication and one for other purposes such as oil/gas, local ocean observatory and science in general, without interfering with each other. The dual-conductor cable and four-cable branch unit recently developed by Tyco Electronics Subsea Communications [8] and the Modular Connecterized Distribution Unit by Ocean Design Incorporation (Flynn, personal communication, 2010) enable independent power and fibre connectivity in a layered network allowing three power path and nodes for multiple connections. The currently used repeaters have space to install the temperature, salinity and pressure sensors (Figure 3). The measured signals can be transmitted to shore via the dual-conductor cable with two coloured fibres, one for telecommunication and one for scientific data. As a result, the chain of repeaters in a cable can be turned into a densely sampled mooring-like time-series with instant data availability spanning seconds to several decades. Using just a fraction of the new cable systems being constantly installed would provide thousands of time series stations to form a truly real-time global network. Such a network will effectively monitor global climate change including long-term sea-level rise as well as short term tsunami detection and warning at low cost.

To make the network more useful, future technical development should consider inductive (transformer) transferring of power and communication through a pressure case without connectors. Even power of a watt and communications rate of bits per second is useful. With the inductive capability of power and communication, horizontal electric field (HEF) pressure –inverted echo sound (HPIES) can be placed next to the repeater using a remotely operated vehicle (ROV). A repeater can support an acoustic modem for communications (acomms) [4]. This allows instruments to be placed a farther distance away from the repeater with a large battery.

At present, many of the available technologies are first developed in the oil/gas industry a beneficial situation towards achieving the global climate change monitoring network. As more and more of these technologies are moving to fully instrumented and widely distributed fields for production such as command, control, power and communication as well as 4-D tomographic monitoring of the fields, they, if extensively qualified, can then be migrated to telecommunications.

IV PERSPECTIVE

The recent great earthquakes and resulting tsunamis (Sumatra 2004, Chile 2010, Japan 2011) show yet again how vulnerable our civilization can be to natural disasters. As part of the global effort to build a tsunami warning system, USA has deployed a total 39 Deep-ocean Assessment and Reporting of Tsunamis (DART) tsunami stations (or buoys) in the Pacific and Atlantic Oceans and Caribbean Sea by 2008 (Figure 2). More buoys have been planned in other parts of the world oceans. At the same time, many other coastal countries have deployed or planned to deploy their tsunami buoys as well. The cost for purchasing a DART buoy is typically about US$250,000, and maintaining cost for a buoy is about US$125,000/year excluding ship-time; the latter typically costs much more than the buoy itself [9]. Tsunami buoys are usually deployed as close to potential source locations (e.g., along the entire “rim of fire” in the Pacific) as possible for achieving longer lead and evacuation time. But there are still many gaps, for instance, the current system has not covered the mid-ocean ridges and vast open ocean area which are far from land and would be more costly to maintain with ships. On 11 March 2011 Japanese tsunami crossed the open Pacific and caused damage and casualty in west coast of America. Due to lack of buoys in vast open Pacific, insufficient information is provided to public on tsunami travel time, speed and amplitude.

The principal of the DART (or other types) buoys, for example, developed by US National Oceanic and Atmospheric Administration (NOAA), is the ocean bottom pressure sensor which can record tsunami wave amplitude of less than 1 cm in open ocean. As depicted above, if the pressure sensors are integrated into submarine cable repeaters, harnessing telecoms cable repeaters can form a global real-time tsunami warning network with less cost than the present system. That is because the cabled measurements can save the maintenance/ship costs. Further, tsunami buoy life times are limited to a maximum of 4 years because they are powered by batteries. In contrast, cable systems can power the sensors for decades. The harsh sea-surface environment, possible device failure and potential vandalism reduce the tsunami buoys’ effective availability. Cables lie on the ocean floor avoiding the harsh sea surface condition. The designed operation time for DART buoys is typically 80 per cent but actual available up-time is less than 70% [10]. A cabled tsunami warning network is more reliable in terms of its better global ocean coverage and highly dense sensors.

Because warming water expands, sea-level rise is an integrating measure of global ocean heat content, and global warming. Current sea-level observations rely on tide gauges and satellites. Tide gauge records are affected by land rising and sinking, showing considerable variations in the long term. The estimated sea level rise measured with satellite altimetry is greater than with tide gauges. It is still unclear how much the precision of the satellite measurement is affected by orbital decay and the difference between assumed orbit and the Earth geoid. Cabled measurements of sea-bottom pressures are expected to provide more reliable and accurate sea-level data as they measure the whole water column pressure change and better cover the ocean-basins with less effects of vertical ground movement for decades.

With regard to polar climate change condition, the polar region has been warming stronger than global average. In 2007, the summer minimum sea ice in the Arctic was an unprecedented 40 per cent below the minimum sea ice content of the 1980s. Autumn temperatures were 6°C higher than the 1958-98 mean. The Greenland ice sheet has been melting with a 16 per cent increase from 1979-2002. In Antarctica, a significant temperature anomaly of more than +2°C has been found in the Antarctic Peninsula. In responding to the warming, a collapse of the Antarctic ice shelf and drifting of icebergs away from Antarctica are frequently reported in recent years. The polar region’s augmented warming may have already impacted the polar water-mass formation, transport and greenhouse gas carriage. Details are unknown as no effective monitoring is being performed. With the aforementioned temperature, salinity and pressure sensors, submarine cables and repeaters can effectively contribute to the monitoring of polar and global climate change at a low cost. Since the demand for internet usage continues to grow exponentially, telecom cables will only expand. With little additional cost, a new generation of cable repeaters can meet the requirement of very much needed climate change monitoring system. For example, in January 2010, Kodiak Kenai Cable Co. announced plans for a fibre optic cable on the Arctic Ocean seabed to connect Europe and Asia [7]. Australian Telstra also announced recently it would lay a new cable from Sydney to Oakland and then to Los Angeles using more than 500 repeaters with 40 km spacing. These cables and repeaters are still going to serve only a single purpose, i.e., to amplify the communication signal. Telecommunication companies and scientists are urged to work together to realize the much-needed global climate change monitoring network in future cable systems.

These business opportunities for telecommunication companies are obvious and should not be missed. The current services in tsunami warning and sea-level monitoring are mainly financed by governments. In the future model of invest-outcome/invest-profit for efficiently using tax-payer money, governments should work with and persuade and encourage telecom companies to incorporate the new technologies as part of in the overall emission and mitigation strategy. By encouraging technical standardization, the United Nation (UN) agent, International Telecommunication Union (ITU), as well as non-governmental organizations (NGOs) can facilitate the implementation of the capability. To determine the next step, a workshop has been organized to be held in Rome on 5-9 September 2011 including scientists, engineers, governments and legal experts, in cooperation with relevant organizations and UN agencies. The workshop aims to facilitate the use of retired and in-service submarine cables and encourage the development of new technologies and standards for layered systems and of ocean observing subsystems at each repeater.

At the UN Copenhagen Climate Conference in December 2009, all nations unanimously agreed to curb global warming not to exceed 2 °C. Since human-induced global greenhouse warming will soon cross the 1 °C mark and approach 2 °C [11], the next decades will be crucial for monitoring climate change. As the ocean is one of the most important factors in governing the worldwide warming process and climate variability, they must be closely observed. Without other effective means for long-term climatic measurements, harnessing telecom cables for ocean observation is expected to play a major role in monitoring global climate change for the next decades.

REFERENCES

[1] Y. You, “Multipurpose submarine cable repeaters required to monitor climate change,” Submar. Tele. Foru. Mag. London, vol. 54, pp. 7–11, 2010.

[2] Y. You, Using Submarine Communications Networks to Monitor The Climate, vol. 15. Intern. Tele. Union: Geneva, 2010, pp.11.

[3] L. Carter, D. Burnett, S. Drew, G. Marle, L. Hagadorn, D. Bartlett-McNeil, and N. Irvine, Submarine Cables and The Oceans: Connecting The World, UNEP-WCMC Biod. Series, vol. 31, ICPC/UNEP-WCMC, 2009, pp. 64.

[4] Y. You, “Harnessing telecoms cables for science,”Nature, vol. 466, pp 690-691, 2010.

[5] W. H. Munk, Afairs of the Sea, Ann. Rev. Ear. Plan. Sci., 1980, pp 1-17.

[6] . C. Larsen and T. B. Sanford, “Florida Current volume transport from voltage measurements,” Science, vol. 227, pp. 302-304, 1985.

[7] Y. You, “Telecom ompanies could help create global monitoring network,” Sea Tech. Mag., vol. 51 (11), pp. 73, 2010.

[8] M. E. Kordahi, “Dual-conductor cable and a four-cable branching unit meet evolving needs for transoceanic underseas cable networks,” Sea Tech. Mag., vol. 51 (7), 2010.

[9] E. N. Bernard, F.I. Gonzalez, C. Meinig, H.B. Milburn, M.C. Eble, S.E. Stalin, and E.F. Burger, “DART buoys provide real-time reporting of tsunamis,” Tsuna. Newslett., vol. XXXIV, No. 2, pp 3-8, 2002.

[10] National Research Council, Tsunami Warning And Preparedness: an Assessment of the U.S. tsunami program and the nation’s preparedness effort, The National Academies Press, 2010, pp.350.

[11] IPCC AR4, Climate Change 2007: The Physical Science Basis (Summary for Policy Makers), IPCC, 2007.

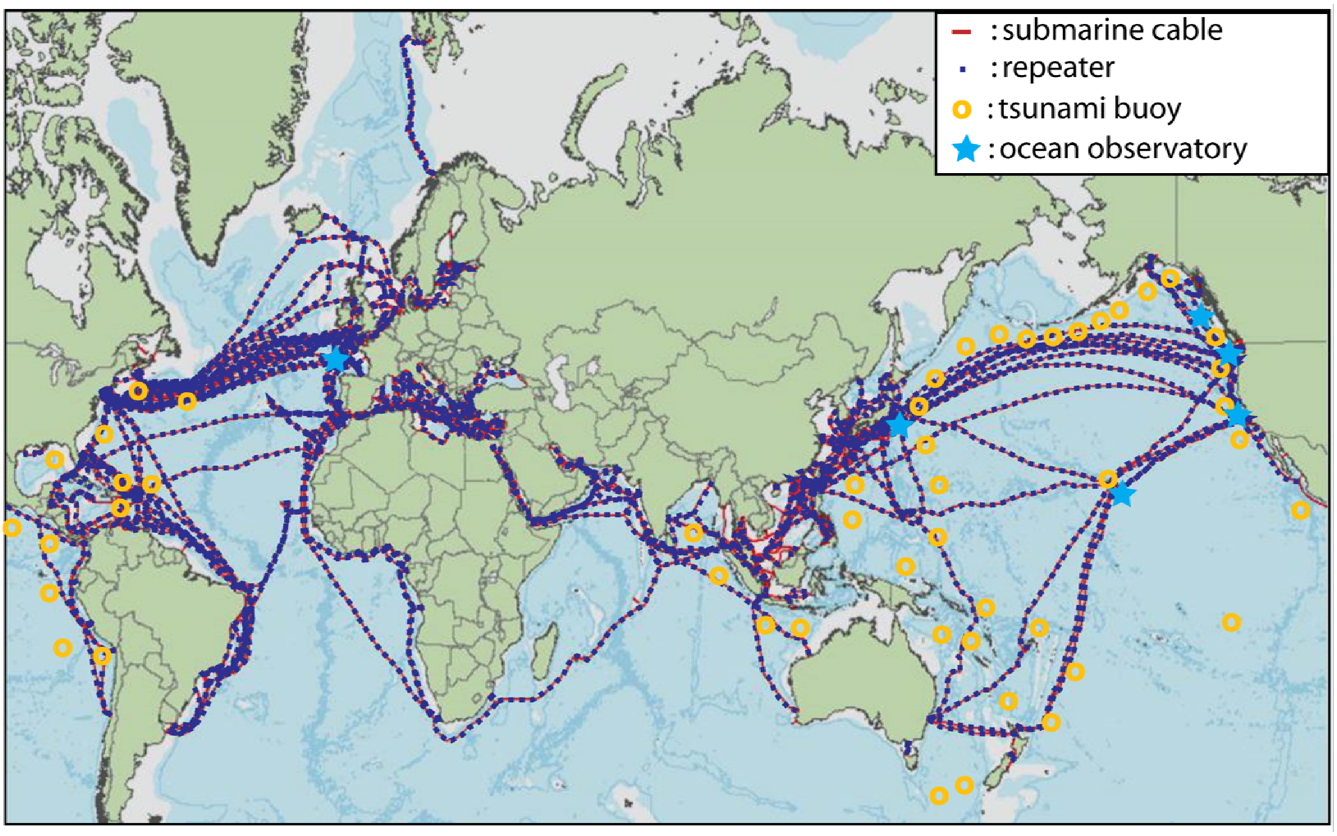


Figure 2. Submarine cables repeaters (blue dots) are symbolically plotted overlapping the cables (in red). The actual number of repeaters is about 4 times more than that plotted with a distance of about 40-150 km apart. For example, a typical transpacific cable would contain about 200 repeaters. Other plotted symbols are tsunami buoys (open circles in orange) and ocean observatories (stars in light blue). The source of background cable distribution is from the cable map of Global Marine Systems Ltd.

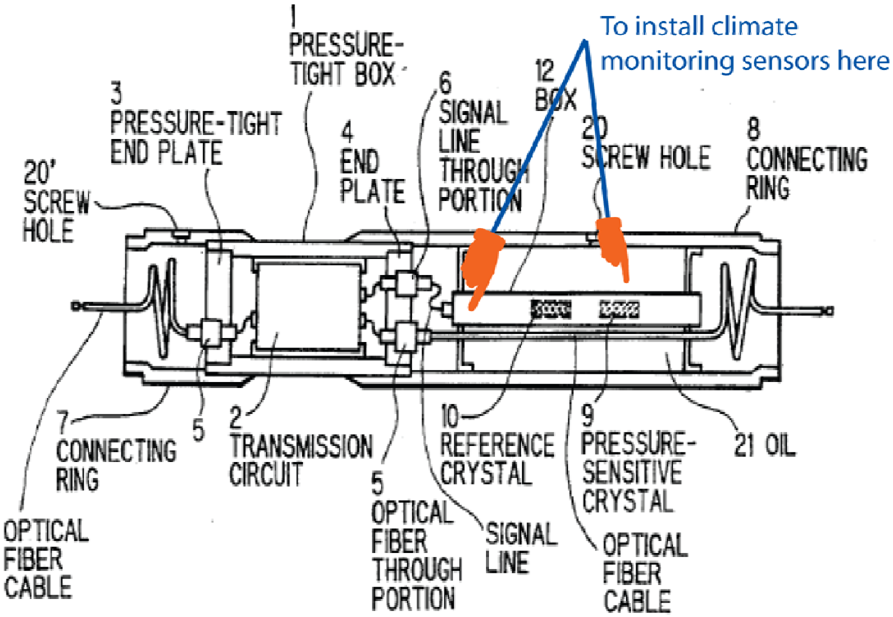

1 PRESSURE-TIGHT BOX
To install climate monitoring sensors here
12 BOX
3 PRESSURE-TIGHT END PLATE
6 SIGNAL LINE THROUGH PORTION
4 END PLATE
20' SCREW HOLE
20 SCREW HOLE
8 CONNECTING RING
7 CONNECTING RING
5
2 TRANSMISSION CIRCUIT
5 OPTICAL FIBER THROUGH PORTION
10 REFERENCE CRYSTAL
SIGNAL LINE
9 PRESSURE-SENSITIVE CRYSTAL
21 OIL
OPTICAL FIBER CABLE
OPTICAL FIBER CABLE

Figure 3. Cable and repeaters are being laid down in the ocean (upper panel), a view of open Alcatel-Lucent repeater (middle panel) and the diagram of a submarine cable repeater (lower panel). Climate monitoring sensors could be integrated into the equipment and the measured signals transmitted back to a shore station via the repeater's transmission device.

Data Prediction in WSN using Variable Step Size LMS Algorithm

Biljana Stojkoska, Dimitar Solev, Danco Davcev
Faculty of Electrical Engineering and Information Technologies, University “Ss. Cyril and Methodius”,
Skopje, Macedonia
biles@feit.ukim.edu.mk, dimitar.solev@gmail.com, etfdav@feit.ukim.edu.mk

***Abstract*— Wireless communication itself consumes the most amount of energy in a given WSN, so the most logical way to reduce the energy consumption is to reduce the number of radio transmissions. To address this issue, there have been developed *data reduction strategies* which reduce the amount of sent data by predicting the measured values both at the source and the sink, requiring transmission only if a certain reading differ by a given margin from the predicted values. While these strategies often provide great reduction in power consumption, they need a-priori knowledge of the explored domain in order to correctly model the expected values. Using a widely known mathematical apparatus called the Least Mean Square Algorithm (LMS), it is possible to get great energy savings while eliminating the need of former knowledge or any kind of modeling. In this paper with we use the Least Mean Square Algorithm with variable step size (LMS-VSS) parameter. By applying this algorithm on real-world data set with different WSN topologies, we achieved maximum data reduction of over 95%, while retaining a reasonably high precision.**

Keywords-Wireless Sensor Network; Data Prediction; Least Mean Square Algorithm; Time Series Forecasting.

I. INTRODUCTION

By being inherently distributed systems, WSN allow not only measuring the temporal progression of the ascertained quantity but also provide the ability to take the spatial progression of this quantity as well. By reporting data measurement at each interval, the node itself consumes a great deal of energy, thus, it vastly reduces its lifetime and creates sufficient communication overhead.

There are several techniques that have been developed to overcome these problems i.e., to lower the communication overhead, to increase energy saving and maximize CPU utilization.

Since wireless communication itself consumes the most amount of energy in a given WSN, the most logical way to reduce the energy consumption is to reduce the number of radio transmissions.

Data-reduction techniques aim to reduce the data to be delivered to the sink. These techniques can be divided into three main groups (Fig. 1): data compression, data prediction and in-network processing [1].

Data compression is applied to reduce the amount of information sent by source nodes. This scheme involves coding strategy used to represent data regardless of their semantics.

In-network processing performs data aggregation while data is routed towards the sink node. Data aggregation aims to transform the raw data into less voluminous refined data. It can be achieved with summarization functions (*minimum*, *maximum* and *average*). For applications that require original and accurate measurements, such a summarization may be inappropriate since it represents an accuracy loss [2].

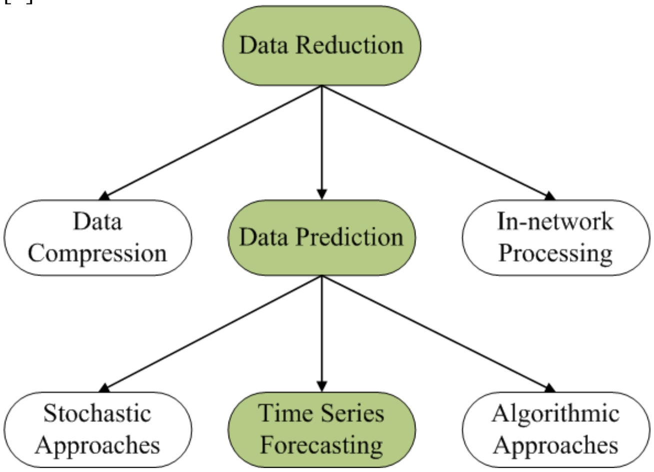


Fig. 1. Data-driven approach for energy saving in WSN [1].

Data prediction techniques usually maintain two instances of a model in the network, one residing at the sink and the other at the sensor. The model at the sink can be used to answer queries without requiring any communication, so the original data can be easily reconstructed within a certain degree of precision. To avoid a rapid deterioration in the predicted values, such approaches thus need their models to be periodically validated and correspondingly updated, implying again increased communication costs.

Data prediction techniques can be divided into three subclasses: stochastic approaches, time series forecasting and algorithmic approaches, which are application specific (Fig. 1).

Stochastic approach is used when sensed phenomena can be considered as a random process by means of probability density function. Although this approach is general, its

ISBN: 978-1-61208-144-1

computational overhead makes it inappropriate for tiny sensors with limited computational capacities.

The most appropriate models for data prediction in WSN are those based on time series forecasting. Moving Average (MA), Auto-Regressive (AR) or Auto-Regressive Moving Average (ARMA) models are simple, easy for implementation and provide acceptable accuracy [3][4]

In this paper, we investigated time-series forecasting technique for WSN based on LMS algorithm with variable step size (LMS-VSS). Least-Mean-Square (LMS) adaptive algorithm as a data-reduction strategy in WSN has been firstly proposed by Santini and Römer [5]. The advantages of LMS algorithm is that it does not require any a priori knowledge or modelling of the statistical properties of the observed signals, thus providing great flexibility and domain independence. Santini and Römer in [5] reported maximum data reduction of 92% for the temperature measurements (on Intel Berkeley Lab dataset [6]) while retaining an accuracy of 0.5°C.

The fact that a great percentage of the real-world wireless sensor networks are multi hop and hierarchy based, motivated us to exploit LMS-VSS on two different network topologies: star topology and cluster-based topology. The proposed algorithm is tested on real data obtained from the Intel Berkeley Research Laboratory sensor deployment [6]. LMS-VSS produce maximum data reduction of 95% for error margin of 0.5°C (for star topology) and around 97% when data aggregation was taken into account (cluster-based topology).

The rest of the paper is organized as follows: the next section explains least mean square algorithm. Section three of this paper describes the LMS-VSS. The fourth and the fifth section explore LMS-VSS on star network and cluster-based topologies respectively. Finally, we conclude this paper in section six.

II. Least Mean Square algorithm

In this section, we present a brief explanation of the least mean square algorithm. A thorough explanation can be found in [6].

A linear adaptive filter samples a data stream/input signal at an instant n, which we will denote as $u[n]$ and calculates a prediction i.e., the output of the filter as $y[n] = \underline{w}^T[n]\cdot\underline{u}[n]$, which effectively is a linear combination of the previous N samples of the data stream (denoted as the vector $\underline{u}$ which is of length M), weighed by the corresponding weight vector $\underline{w}[n]$ (also of length M). M is an integer parameter that the filter uses and it determines the "memory" of the filter i.e., how many previous input sample it will use.

The output y[n] is then compared to the input signal or the sample of the data stream the filter tries to adapt to, denoted as d[n]. The prediction error e[n] is then computed as: e[n] = y[n]−d[n] and fed into the adaptation algorithm, so the filter weights can be updated. The vector w[n] i.e., the weights are modified at each time step n in order to minimize the mean square error.

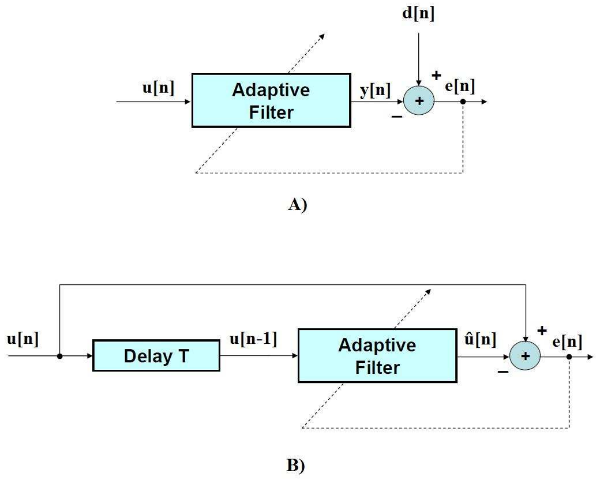


Fig. 2. A) Basic structure of an adaptive filter; B) Adaptive filter used for prediction.

One of the most extensively used adaptive algorithms is the Least-Mean-Square algorithm (LMS). Although it is extremely simple, it has quite good performances and has found implementation in a variety of applications. [7][8]. The LMS algorithm is defined through the three equations:

1. The filter output:
   $y[n] = \underline{w}^T[n]\cdot\underline{u}[n]$,
2. The estimation error:
   $e[n] = d[n] - y[n]$
3. The weight adaptation:
   $\underline{w}[n+1] = \underline{w}[n] + \mu\cdot\underline{u}[n]\cdot e[n]$

where $w[k]$ and $x[k]$ denote the M × 1 column vectors:

$$\underline{w}[k] = [w_1[k], w_2[k], \ldots, w_N[k]]^T ;$$
$$\underline{u}[k] = [u[k-1], \mathrm{u}[k-2], \ldots, \mathrm{u}[k-M]]^T .$$

With simple modification the filter structure from Fig. 2(A) to the so-called predictive structure of Fig. 2(B), the LMS algorithm can be used for prediction. Central to the successful prediction is delaying the current input value $u[n]$ by one time instance and use it as the reference signal $d[n]$. The filter then computes an estimation $\hat{u}[n]$ of the input signal at time instance n, as a linear combination of the N previous readings. Subtracting the prediction signal from the desired signal gives the value of the error which is fed back to adapt the filter weights.

Two parameters need to be defined for the adaptation process: the filter length M and the step-size parameter μ, which is important for updating the filter weights.

Given that $\underline{w}$ and $\underline{u}$ are M × 1 vectors, it can easily be concluded that the LMS algorithm requires $2M + 1$ multiplications and $2M$ additions per iteration. [7].

The practical implementation of the LMS algorithm in the prediction scheme in WSN was introduced in [5] where identical predictive filters were introduced both at the source and at the sink. LMS dual prediction scheme (henceforth referred to as LMS-DPS) consists of simultaneously running an instance of the filter on both the node and the sink. There are three distinct modes of operation: initialization, normal and stand-alone. A node goes through the first only at the beginning and then switches between the normal and stand-alone modes. When a node is in the stand-alone mode it does not report its readings to the sink and this is where the energy savings are made.

The detailed description of each of the modes is as follows:

A. *Initialization mode:*

First, the step-size parameter μ must be determined. A certain amount of data must be collected in the beginning, so a proper estimation of the step-size can be made. At this time, the node keeps sending the data to the sink without making predictions. Both the node and the sink compute the value of μ. To ensure convergence [7][8] it must satisfy:

$$0 \leq \mu \leq \frac{2}{E_x} \tag{2}$$

where:

$$E_x = \frac{1}{N}\sum_{n=1}^{N}|x[n]|^2 \tag{3}$$

E_x is the mean power input and N is the number of iterations used to train the filter. Since the input mean power E_x is continually dependent on time, we can approximate $\hat{E}_x$ by computing over the first M samples i.e., N data readings and easily obtain the upper bound of (2).

After the initialization phase, both the node and the sink will continue to execute the predictive algorithm and node will be switching between the following two modes of operation.

B. *Normal mode:*

Both the node and the sink simultaneously execute the predictive algorithm and make a prediction for the following reading by using the last M readings and accordingly update the filter weights on the prediction error. As stated in [7], if no a priori knowledge is present, the initial weights should be zero. This is rather important, because:

a) we do not possess any a priori knowledge of the data;

b) setting the initial weights to zero (and using the same values for the step-size parameter and for the filter length), ensures that any instance of the LMS will behave exactly the same at any arbitrary point of time t_n.

The node will stay in normal mode (collecting data and reporting it to the sink) as long as the prediction error is greater than the maximum error budget e_{max}. When the error drops below e_{max} for M consecutive iterations, then the node switches to stand-alone mode i.e., stops reporting the readings and consequently stops updating the weights.

C. *Stand-alone mode:*

In this mode, the node still collects data and makes predictions, but if the prediction error is below e_{max}, instead of the reading $u[n]$, it feeds the filter with the prediction $y[n]$, discards the real reading $u[n]$ and does not send it to the sink. This enables both instances of the filter to be consistent and no update of the weights is needed (the error is zero) thus reducing the computational overhead.

If the prediction error exceeds e_{max} the node switches to normal mode and reports the reading. When the node is in this mode, the filter instance at the sink side uses only the predicted readings as an approximation of the real value.

III. Least Mean Square algorithm with Variable Step Size

The step-size parameter μ is critical for the convergence of the algorithm i.e., it determines the convergence speed, so choosing the right value for μ is of critical importance [9]. As we explain later on, there are practical boundaries for the values that μ can receive and with a simple estimation technique, we can determine them. And as we show in this section, we introduce a specific improvement to the LMS-DPS algorithm regarding the step-size parameter μ.

Since LMS-DPS uses single value for the step-size μ, a further improvement can be made, with the introduction of a least mean square algorithm with variable step-size (which we will refer to as LMS-VSS henceforth), where the step-size parameter μ has two distinct values:

1. Until a certain number of good predictions are made, μ has the maximum value according to (3) and two orders of magnitude smaller to ensure robustness [8].

2. After μ has sufficiently learned what kind of data the filter receives, it switches to a stable value, that is:

$$\mu_{new} = \frac{\mu_{old}}{M}$$

where M is the filter length, and $\mu_{old} = 2 \cdot E_x^{-1} \cdot 10^{-2}$.

This improvement accelerates the initial adaptation to the data, so an additional 3-5% reduction can be gained where the maximum error is sufficiently small, as can be seen in Fig. 4.

A crucial aspect is when the switch should be made. The best value for the number of consecutive iterations in stand-alone mode can be found as:

$$M \leq n \leq M^2$$

where M is the filter length and n is the number of consecutive readings in stand-alone mode. The value n=$M^{3/2}$ yielded best results since it suited best both lower and higher filter lengths. For instance, if M = 4, using $n = M^2$ = 16 consecutive good predictions as a switch point is a good choice, but for M = 10, $n = M^2$ = 100 consecutive good predictions may never be reached, thus compromising performance. In this case, the choice $n = M$ = 10 would be much more optimal. Also, note that the point of switching is determined such that both the node and sink can execute it at the same time and thus retain consistency of both algorithm instances.

One of the advantages of having only two distinct values for the step-size parameter, one to accelerate the initial weight adaptation and another for fine-tuning the weights after the adaptation is that it contributes to the overall data reduction without creating additional computational overhead to the algorithm. Additionally, without using a specific heuristic for a particular type of data, it can be used in other schemes.

In order to compare LMS-VSS and LMS-DPS, both algorithms were implemented in MatLab. For algorithms evaluation, a set of experimental data from Intel Berkeley Research Lab network [6] was used. The 54 Mica2Dot sensors deployed in the laboratory were equipped with weather boards and measured humidity, temperature, light and voltage values once every 31 seconds. The measurements were collected between February 28th and April 5th, 2004. The dataset includes 2.3 million readings collected from these sensors. For our evaluation we use only temperature and humidity readings at different locations in a lab space. We run the simulations for 50 different error margins eMax (raging from 0.1°C to 5°C).

Metrics used for measuring algorithms performance varies from author to author. Some authors tend to reduce the number of transmissions, thus they count the number of sent massages from the sensor node to the sink node [5]. Here, the metric is the reduction of transmissions in percentage. Another way used for evaluating algorithm performance is by measuring the difference between the predicted and the true value, i.e., mean square error (MSE) or root mean square error (RMSE)[4]. In our case, the first metric is used, assuming that every transmission requires an equal amount of energy.

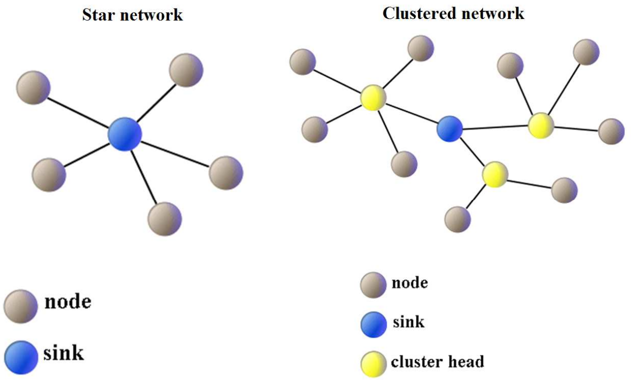


Fig. 3. Star network topology (left), Clustered-based network topology (right).

We consider two basic network topologies on which we evaluated the LMS-VSS algorithm (Fig. 3). The first is the star topology, where every node sends its readings directly to the sink. The advantage of this topology is its simplicity and low latency communications between the sensors and the sink, but the sink must be within radio transmission range of all the individual sensors. If this is not possible, cluster-based network topology is a suitable alternative.

IV. The LMS-VSS Algorithm in Star Topology

If a star network topology is considered, LMS-VSS algorithm achieves considerable reduction in number of transmissions. We use data readings from [6] and assume that each sensor sends its reading to the sink node. Fig. 4 shows the reduction gain when using LMS-DPS and LMS-VSS algorithms. The results are average for all 54 nodes from the Intel Berkeley Research Lab network [6]. LMS-DPS uses only one fixed step size μ=1.2·10^{-5} and filter length M=4 and obtain the average savings in the entire network of around 88% for error margin of 0.5°C. LMS-VSS uses the same filter length, but uses the first M=4 data readings to calculate the initial value of μ and then after $M^{3/2}$ readings switches to $\mu_{new} = \mu_{old} \cdot M^{-1}$. LMS-VSS obtain average savings in the entire network of around 92%.

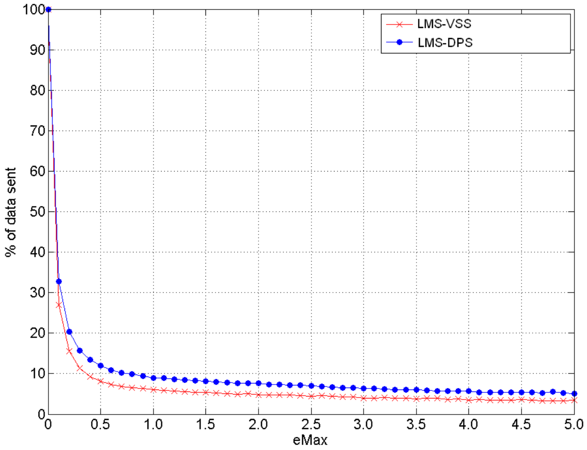


Fig. 4. The improvement of LMS-VSS over LMS-DPS algorithm for the entire Intel network (temperature).

In Fig. 5, we can see the improvement of LMS-VSS over the LMS-DPS algorithm for particular nodes. We chose these nodes because node 11 performs the best results (95% reduction for 0.5°C) and node 49 the worst (90% reduction for 0.5°C).

In addition, we investigated humidity readings from the same Intel network [6], where humidity is temperature corrected relative humidity, ranging from 0-100%. The results for node 20 and node 10 are shown on Fig. 6 and Fig. 7 respectively. As it can be seen from the figures, LMS-VSS performs more than 5% better results compared with LMS-DPS for error margin of 0.5°C, and more than 10% better results for error margin of 1°C and greater. For node 10, root mean square error is given for both algorithms (Fig.7). As can be seen from the figure, our algorithm gives smaller RMSE.

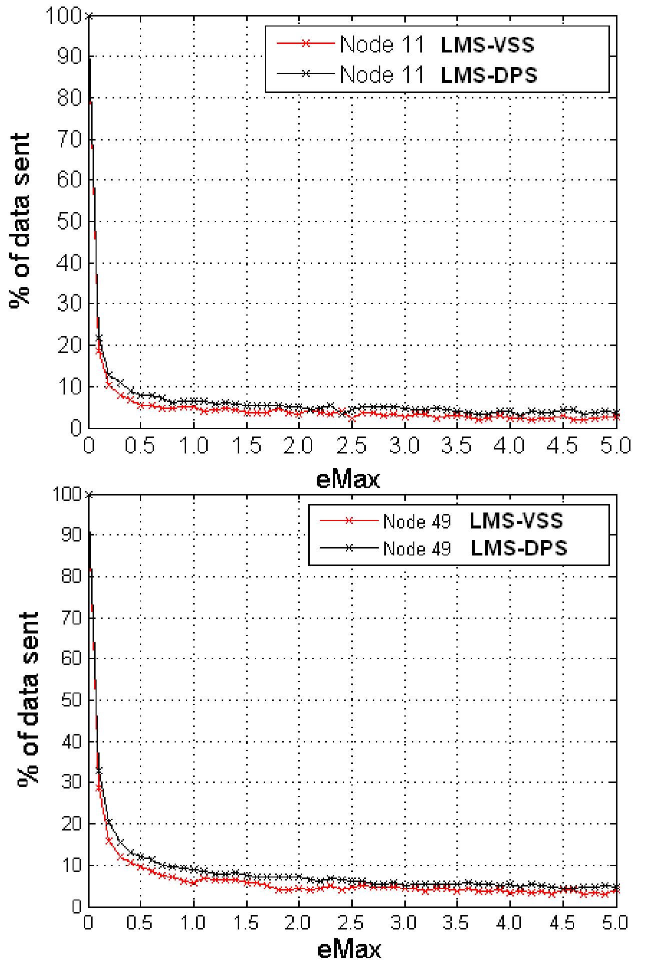


Fig. 5. Improvement of LMS-VSS over the LMS-DPS algorithm, filter length $M = 4$ (temperature).

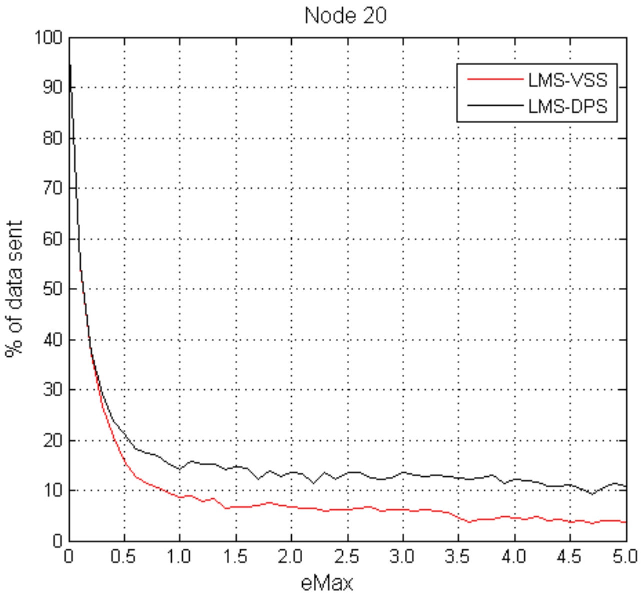


Fig. 6. Improvement of LMS-VSS over the LMS-DPS algorithm for humidity (node 20), filter length $M = 4$.

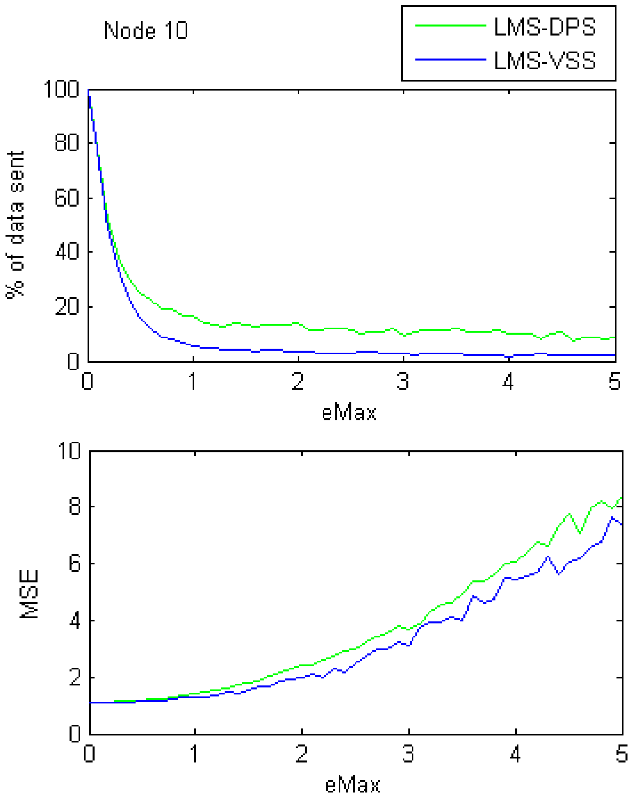


Fig. 7. Improvement of LMS-VSS over the LMS-DPS algorithm for humidity (node 10), filter length $M = 4$.

V. THE LMS-VSS ALGORITHM IN CLUSTER-BASED TOPOLOGY

Nodes clustering represents another way of prolonging WSN lifetime. Here, the sensor nodes are geographically grouped into clusters. In each cluster one representative node is chosen to be a cluster-head. Other nodes in the cluster are called members of the cluster and they report their readings to their cluster head, and cluster head forwards the messages directly to the sink. Note that except for the sink, we consider that all the nodes in the network have exactly the same characteristics, so even when a node is a cluster head, it still acts as a sensing node (and its readings are included in the computations for its respective cluster), with the additional duty to forward readings from cluster members. An example of such clustered-based network is given in [6], as presented on Fig. 8.

Our LMS-VSS algorithm was evaluated in this case by using MatLab simulation tool on the same Intel network [6]. We assume that each sensor sends its reading to the cluster head, and then cluster head resends the reading to the sink. As a result, each reading is sent twice, except the readings taken at the cluster heads. The clustering method we used is the simple k-means clustering, with k = 10. The value k = 10 was selected on a purely intuitive basis and it produced satisfactory results. Other clustering methods may be used as well. As it can be seen from Fig. 8, the clustering parameter was geographic position, i.e., Euclidian distance.

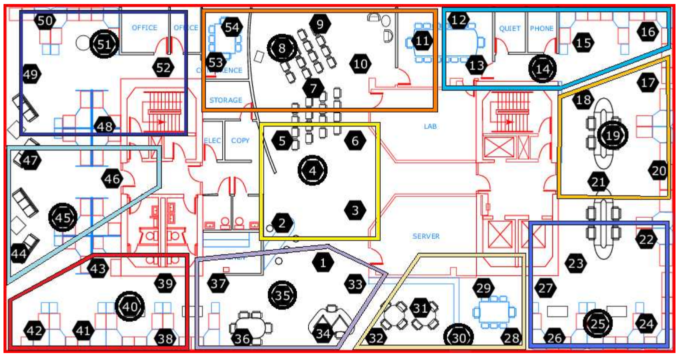


Fig. 8. A clustered view of the Intel Berkeley Research Lab wireless sensor network [6] (cluster heads are circled).

Additionally, we simulated data aggregation technique at cluster heads. In this scenario, every message carries exactly the same amount of data, since the cluster head aggregates the messages from its cluster members using a certain aggregation function (Average, Minimum, Maximum, etc) and forwards the aggregate to the sink.

The step size was $\mu = 1.2 \cdot 10^{-5}$ and the filter length was $M = 4$ for all the nodes, although changing the step-size or the length of the filter for a specific cluster can further improve the performance and the results.

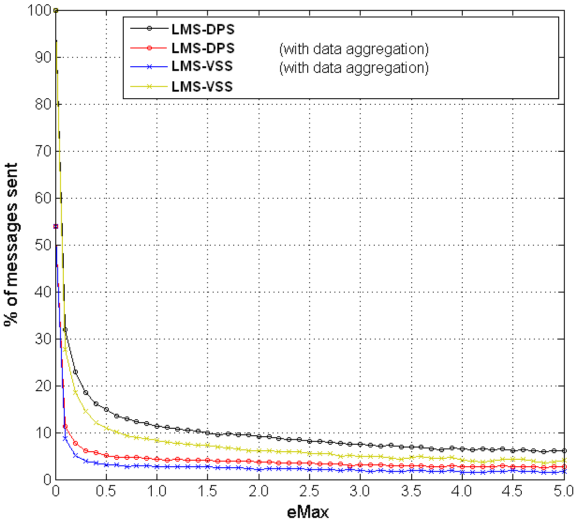


Fig. 9. Various savings in the cluster containing the nodes: 7, 8, 9, 10, 11, 53 and 54 (temperature).

We used the cluster containing nodes: 7, 8, 9, 10, 11, 53 and 54. As displayed in Fig. 9, the averages for this cluster show that there is a substantial gain of 5% when using the LMS-VSS algorithm over the LMS-DPM algorithm. When data aggregation is assumed to be implemented in the cluster, the reduction is far greater and when it is used alongside the LMS-VSS algorithm it results in 97% reduction of the total messages sent for the given error margin of 0.5C.

VI. Conclusion

In this paper, we investigated time-series forecasting technique for WSN based on LMS algorithm with variable step size (LMS-VSS). Santini and Römer in [5] reported maximum data reduction of 92% for the temperature measurements (on Intel dataset [6]) while retaining an accuracy of 0.5°C.

We exploited our LMS-VSS on two different network topologies: star topology and cluster-based topology. We evaluated the LMS-VSS on Intel dataset [6]. Our algorithm outperforms LMS-DPS in terms of data reduction and root mean square error for all evaluated nodes. LMS-VSS performs data reduction of around 92% for error margin of 0.5°C (for the entire Intel network using star topology), and maximum data reduction of around 95% for particular nodes. In cluster-based topology, when data aggregation was taken into account LMS-VSS achieves data reduction of around 97%.

From the simulated results, we can conclude that star network is the most suitable network topology by means of energy saving, since each reading is sent only once. If sensors are not within each other radio range, cluster-based topology could be used, where each reading is being resend by the cluster head. By applying data aggregation at cluster head, this topology can achieve even greater data reduction in scenarios where loosing data precision is affordable. It is obvious that the trade-off is application specific.

For future work, we intend to investigate our LMS-VSS on tree-based network topology, and to evaluate the results using different datasets (from Intel and other networks). We also plan to explore machine learning techniques for choosing the best values for the parameter μ or implement a scheme that will dynamically readjust the filter length.

References

[1] G. Anastasi , M. Conti , M. Di Francesco , and A. Passarella, Energy conservation in wireless sensor networks: A survey, Ad Hoc Networks, vol.7 n.3, pp. 537-568, May, 2009.

[2] E. F. Nakamura , A. A. F. Loureiro , and A. C. Frery, Information fusion for wireless sensor networks: Methods, models, and classifications, ACM Computing Surveys (CSUR), vol.39 n.3, pp. 9-55, 2007.

[3]Y. L. Borgne , S. Santini , and G. Bontempi, Adaptive model selection for time series prediction in wireless sensor networks, Signal Processing, vol.87 n.12, pp. 3010-3020, December, 2007.

[4] C. Liu, K. Wu, and M. Tsao, “Energy Efficient Information Collection with the ARIMA Model in Wireless Sensor Networks”, In Proceedings from IEEE Globecom, vol. 5, pp. 2470–2474, 2005.

[5] S. Santini and K. Römer: An Adaptive Strategy for Quality-Based Data Reduction in Wireless Sensor Networks, Proceedings of the 3rd International Conference on Networked Sensing Systems (INSS 2006), pp. 29-36, Chicago, IL, USA. June 2006.

[6] Intel lab data. Web Page, (Accessed on 06/06/2011) http://db.lcs.mit.edu/labdata/labdata.html

[7] S. Haykin: Least-Mean-Square Adaptive Filters. Edited by S. Haykin, New York Wiley-Interscience, 2003.

[8] G. Moschytz and M. Hofbauer: Adaptive Filter. Springer Verlag, Berlin, 2000, ISBN 3-540-67651-1.

[9] R. Kwong and E.W. Johnston, A Variable Step Size LMS Algorithm, IEEE Trans. On Signal Process., vol. 40, pp. 1633 - 1642, 1992.

Energy Efficient 2-Tiers Weighted in-Sensor Data Cleaning

Jacques M. Bahi, Abdallah Makhoul, Maguy Medlej
Computer Science Laboratory (LIFC)
University of Franche-Comté
Rue Engel-Gros, 90016, Belfort, France
{jacques.bahi, abdallah.makhoul, maguy.medlej}@univ-fcomte.fr

***Abstract*— Managing efficiently the battery and power consumption became a major challenge in sensor networks. Data transmission is very costly in terms of energy which leads to think of a data cleaning technique in order to reduce the size of packets sent to the sink. However, the quality of the information should be preserved during the in-network transmission. This paper introduces a tree-based bi-level periodic data cleaning approach implemented on the source node and the aggregator levels. Our contribution in this paper is two folds. First we look on a periodic basis at each data measured and periodically clean it while taking into consideration the number of occurrences of the measures captured which we shall call weight. A data cleaning is performed between groups of nodes on the level of the aggregator which, contains lists of measures along with their weights. This algorithm will not tolerate the effect of the information that each data measurement provides by preserving the weight of each measure. The experimental results show the effectiveness of this technique in terms of energy efficiency and quality of the information by focusing on a periodical data cleaning while taking into consideration the weight of the data captured.**

Keywords- Sensor Networks, periodic data aggregation, tree based algorithms, quality of information.

I. INTRODUCTION

Controlling and predicting natural disasters, preventing failures, improving food production, overall productivity and improving human well being became eminent demand thus pushing subjects' related to think of instantly recurrent information collection and prediction mechanisms all serving the above purposes among others. Monitoring large range of application areas, mostly in which power or infrastructure limitations make a wired solution costly, challenging or even impossible, constituted an essential goal for scientific researchers' worldwide. Latest researches show that this task is dedicated to spatially distributed autonomous devices. Such devices, or nodes, combined with routers and a gateway constitute a wireless sensor network (WSN). Wireless sensor networks are composed of large distributed number of sensor nodes; each sensor node has a separate sensing, processing, storage, and communication unit. The sensing unit is responsible for gathering data from its environment whereas the processing unit in the form of a microprocessor manages the tasks. Memory is used to store temporary data or data generated during processing. The communication unit communicates with the environment. The distributed measurement nodes communicate wirelessly to a central gateway, which, provides a connection to the wired world where collecting, processing, analyzing, and presenting of measurement data is needed. Each collects a considerable amount of raw data which is sent periodically to a central sink (gateway). Each sensor node is powered by a battery which, supplies energy; however sensor nodes are tightly limited in battery, power and memory storage. As a result, a sensor node is not expected to carry huge amount of data or complex computations. It is important to highlight that a sensor network usually consists of thousands or ten thousands of nodes deployed redundantly in order to ensure reliability and where each single sensor is expected to cooperate with other sensors to provide service. Thus the collected data is partially redundant and is subject to aggregation offering. The major challenge in a wireless sensor network is improving the lifetime of the network in other word managing efficiently the battery and power consumption. Recent researches focused on such task as it is difficult and cost ineffective to recharge the battery. Energy is mainly consumed during data transmission from the source node to the sink (gateway) making network data transmission one of the core issues to address by reducing energy consumption within the wireless sensor network. Furthermore, data accuracy is another main design concern in wireless sensor networks. In order to avoid any faulty alarms, the distributed measurements nodes communicate wirelessly to a central gateway, providing "interaction between people or computers and surrounding environment" [3]. To achieve data accuracy we strongly believe that only the right information should be communicated through the wireless sensor networks. The authors intrigued by such interesting challenge suggest through this article a multilevel data cleaning algorithm aiming to optimize the volume of data transmitted by saving energy consumption and reducing bandwidth on the network level.

This article introduces a periodic multilevel data cleaning algorithm aiming to optimize the volume of data transmitted thus saving energy consumption and reducing bandwidth on the network level. Instead of sending each sensor node's raw data to a base station, the data is cleaned periodically at the first level of the sensor node then another "data aggregator " sensor node collects the information from its associated nodes. We shall call this first level "in-sensor process periodic cleaning" approach. A second cleaning is applied on the level of the aggregator node itself. The

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

cleaned data is finally sent from the aggregator to the base station. It is important to note that the weight of each measure (number of occurrences of each measure in the set) is preserved through both described above techniques thus preserving the quality of information provided by each measure. The described approach pioneers in the field of focusing on a periodical data cleaning while taking into consideration the weight of the data captured.

The rest of the paper is organized as follows: The 1st section presents and accredits data cleaning and aggregation related work and research. While the second introduces the first level periodic data cleaning algorithm applied on the sensor node level. The third presents a heuristic method aiming to clean data on the aggregator level and index the data cleaned by a weight significant of its redundancy and quality. The fourth section shows the experimental results of our suggested multi cleaning algorithm and its contribution to the network life through optimizing energy consumption. We conclude by emphasizing the added value of our approach and its contribution to the world of wireless sensor network research.

II. RELATED WORK

Limited battery power and high transmission cost in wireless sensor networks make in-network cleaning and aggregation a challenging area for research. Data transmission is the most costly operation in sensors [1], compared with it, the energy cost of in-network computation is trivial and negligible. Reducing the number of packets being transmitted in the network will eventually lead to energy consumption reduction. In order to reduce the number of packets, data cleaning and data aggregation related approaches have been conducted. Based on this, we have presented some network data reduction and aggregation related works that fall into different levels of data cleaning approach: In-network approach between hops or on the level of the sensor itself where each sensor takes up some computation according to the applications (e.g., query processing, data collection, event detection, and so on). Several performance measures like network lifetime, data accuracy, false alarm, high data redundancy, latency and scalability need to be considered concurrently [4][5]. Zhuang and Chen Hong Kong [2] focuse on the outliers cleaning within multi-hops by including wavelet based outlier correction and neighboring DTW (Dynamic Time Warping) distance-based outlier removal. The cleaning process is accomplished during multi-hop data forwarding process, and made use of the neighboring relation in the hop-count based routing algorithm. On the other hand, data aggregation methods in sensor networks have been reported [11]. Zheng, Chen, and Qiu [12] propose a method to build an aggregation tree model in WSN such that the captured data are aggregated along the route from the leaf cells to the root of the tree. In this scheme, the tree is not built directly on sensors, but on the non overlapping cells which, are divided with equal sizes in the target terrain. A representative sensor in each cell acts in name of the whole cell, including forwarding and aggregation of the sensing data in its cell and the receiving data from the neighbor cells. In light of large-scale and high-density sensor nodes, the scheme cuts down the data transmission overhead from three aspects. Firstly, primary aggregation should be conducted in the cell, based on the observation that the measurement data in one small cell are almost identical. Secondly, aggregation operation in one large-scale network should be directed to avoid the dynamic change of aggregation topology. Finally, using cell-by-cell communication instead of hop-by-hop communication reduces the density of communication and the complexity of the aggregation topology in the network. Greedy aggregation is proposed in [9][11], where a tree is constructed to indicate the path from each sensor node to the sink. The shortest path linking a node to the sink is used as the initialization of the tree. Then, the shortest paths linking the remaining nodes to the current tree will be incrementally added to enlarge the tree. With this technique, the packets will be aggregated as early as possible and the aggregated packet will be directly routed back to the sink. However, the efficiency of the greedy incremental method is entirely determined by the shortest path. The data transmission is not reliable since once the path is broken, a large region will be disconnected and will not be able to send information to the sink.

All the presented work didn’t take into consideration the accuracy of the information affected by the number of similarity between measures. In this paper we shall focus on periodic data collection at the first level of sensor nodes. We consider that at each determined time interval the sensing unit is configured to capture measurements. At this level, our approach consists of comparing measurement captured at an interval of time t with measurements already captured at a previous interval in order to perform some in-sensor processing and evaluate data. We shall call this in-sensor process periodic data cleaning approach. Our aim is to periodically clean the data captured from noisy and redundant measures while maintaining an acceptable level of quality and accuracy of the information that is deduced from the captured measures. The measures' occurrences are called weighted measures in this article and will serve as a parameter passed to all data cleaning levels and subsequently saving accuracy of the purged data. Such scheme will form in future works training set for the classifier, predicting with reasonable accuracy the class of each instance fed. When applying the suggested algorithm, it cleans periodically the data while assigning to each measure its proper weight. The cleaning will be processed in two steps: the source node will constitute the first step whereas a special sensor node called aggregator receiving the data from different source nodes will conduct data cleaning at the second step.

III. DATA AGGREGATION SCHEMA

This section gives the main definitions and notations, together with our approach that will be used for an efficient and accurate in sensor nodes data reduction. The main focus is the periodic data collection where each sensor takes measurement at regular time interval. We classify our approach as 2 tiers data cleaning approach: the source node will constitute the first tier whereas a special sensor node called aggregator receiving the data from different source nodes will be the subject of data cleaning at the second tier. Fig. 1 illustrates our tree based data aggregation scheme. At the first tier exists the source nodes. The second tier contains the aggregator.

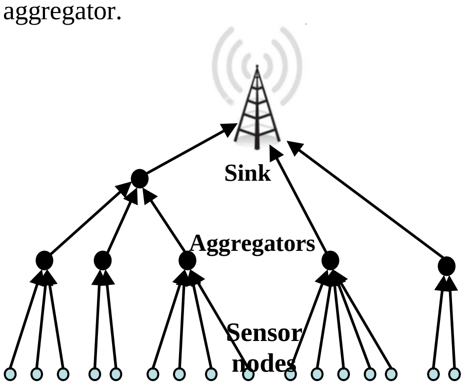


Figure 1. Tree based data aggregation scheme

A. Definitions and Notations

The set of sensor nodes is denoted by N = {1, 2, …, n}, where n is the number of nodes. Each node is composed of many sensors S that produce a measurable response to a change in a physical condition like temperature or pressure or humidity, etc. Each sensor node takes a vector of measurements $M[t] = (m[t_1],\ldots m[t_{\prod\text{-}1}])\in\Re^{\Pi}$ at regular time interval t during a period $\prod$. The unit time is called slot, whose length is the time interval between two measurements. After $\prod$-1 slots, each sensor node N_i will have a vector of measurements M_i as follows:

$$\begin{array}{c} M_1 \\ M_2 \\ M_3 \\ \\ \\ M_S \end{array}\begin{bmatrix} m_1[t_1] & m_1[t_2] & \ldots\ldots. & m_1[t_{\pi-1}] \\ m_2[t_1] & m_2[t_2] & \ldots\ldots. & m_2[t_{\pi-1}] \\ m_3[t_1] & m_3[t_2] & \ldots\ldots.. & m_3[t_{\pi-1}] \\ & & & \\ & & & \\ m_S[t_1] & m_S[t_2] & \ldots\ldots.. & m_S[t_{\pi-1}] \end{bmatrix}$$

Definition1: Substitution between two measures

At each interval I and for each sensor s, we associate to each measure $m_s[t_i]$ a function noted Substitution($m[t_i]$, $m[t_j]$) which, define a kind of similarity to unify or not with a measure $m_s[t_j]$ taken a time t_j / j< i.

$$\mathrm{Substitution}(m[t_i], m[t_j]) = \begin{cases} 1 & \text{if } ||m[t_i] - m[t_j]|| \leq \delta \\ 0 & \text{otherwise.} \end{cases}$$

where δ is a threshold fixed by the application.

Definition2: Weight of a measure

Intuitively, redundancy gives more importance to some information which, are represented by many features and may occult less than others that are less present. We define weight of the measure m at a time t the total number of measures captured after the time t and can be unified with m.

$$\mathrm{Weight}(m[t_i]) = \Sigma^{\Pi}{}_{j=ti+1}\mathrm{Substitution}(m[t_i], m[t_j])$$

Definition3: Cell's measure

On the level of an aggregator A, we define a cell's measure cell $C_a[i] = A[i](\lambda_i, m_i)$ such that the cell contains the received measure m_i from the node n_i with its corresponding weight λ_i. A cell $C_a[i]$ is built based on distinct measures and weights existing in the aggregator A. we refer to $C_a[i](m)$ as the measure in a cell while the respective weight is $C_a[i](\lambda)$.

B. First Tier: Periodic data Cleaning

The proposed method calculates on periodic basis the substitution function between the measures already captured and the current measure captured at the current period. If the substitution is equal to 1, which, means that the new measure can be unified with the existing measure on which we are performing the substitution function, the weight of the existing measure is incremented by 1 and the new measure is disregarded. Algorithm1 illustrates the first tier. At the end of this algorithm, no redundant measure will exist. Each sensor will send to the aggregator a set of reduced measures associated to their corresponding weight and ready for the 2nd tier data cleaning algorithm.

C. Second Tier: Weighted data Cleaning

We define a special node for each set of nodes which, we shall call "aggregator", such aggregator will receive data from its set of nodes. We assume that the aggregator is more powerful than its set of nodes N. At this stage each aggregator will hold n lists for each type of measurement where n is the number of nodes associated to this aggregator and each list contains measures with their related weights.

Our approach aims at reducing data transmitted from the aggregators to the sink subsequently reducing energy consumption. The obvious idea will suggest looping each list comparing its measures with the remaining lists looking for redundant data. Such approach proved to be costly in terms of data processing since it will scan the whole existing set many times and is attributed a complexity of O(n!). Our approach, illustrated in Algorithm. 2, suggests building progressively a dynamic arraylist as follows:
We define A = Union of all existing lists in the aggregator: $A = (\cup(\lambda_j, m[t_{,i}])|i\in N)$. Than we select a random measure with its related weight from A in order to create the first cell of our dynamic array list by placing the above random value in it. We continue by selecting value (λ_i, m_i) from A and calculating the Substitution function for each selected value m_i with the array list values { (λ_i, m_i), j∈array list values}. The first measure m_j answering Substitution $(m_i, m_j) = 1$ is observed and the weight of matched values are added. If no match occurs the value is added to the dynamic array list by creating a new cell. Finally, the selected value m_i is deleted

ISBN: 978-1-61208-144-1

from A. As we proceed in the algorithm an array list is built up.

D. Illustrative Example:

Let AM be the set of values related to one type of measures received from different nodes connected to an aggregator A.

AM = $\{(\lambda_{11}, m_{11}), (\lambda_{12}, m_{12}), .., (\lambda_{21}, m_{21}), …., (\lambda_{nk}, m_{nk})\}$.

We create the first cell in the array list where we place the first value (λ_{11}, m_{11}).For each (λ_{ij}, m_{ij}) we compute Substitution $(C_a[1](m), m_{ij})$ where m is a measure from AM. If the function returns 1 it means that these two measures are similar. Then the weights are added to each other and we remove (λ_{12}, m_{12}) from the set A, else we create a cell $C_a[2]$ for m_{12} affected of the weight λ_{12} as shown in Table I. At the end we remove (λ_{12}, m_{12}) from the set A.

```
Algorithm1: First Tier Data Cleaning.
Input:
New measure $m[t_i]$.
Output:
Reduced set of measurements M.
For each slot $t_i$ during a period $\prod$ do
        Get a measure $m[t_i]$
        For each measure $m[t_j]$ do
            If Substitution ($m[t_i], m[t_j]$) = 1 then
                $\lambda_j \leftarrow \lambda_j+1$ // $\lambda_j$ is the weight($m[t_j]$)
                Disregard $m[t_i]$
            Else $\lambda_i \leftarrow \lambda_i+1$ //$\lambda_i$ is the weight($m[t_i]$)
                Add $m[t_i]$ to M: $M \leftarrow \{(M \cup(\lambda_j, m[t_{,i}])\}$
            End if
    End for
End For
```

TABLE I. ARRAYLIST UNDER CREATION

Cells	$C_a[1]$	$C_a[2]$
	$\lambda_{11,,} m_{11}$	$\lambda_{12,} m_{12}$

Supposing we are in the case where the measures are not similar we continue as follows:

We move to (λ_{13}, m_{13}), then we check if the similarity is reached with the measure m. If so, the weights are added as follows: $C_a[1](\lambda) = C_a[1](\lambda) + \lambda_{13}$ and $m=m_{13}$ is removed from the set A. Otherwise we continue checking the similarity with the measure existing in the second cell. If Substitution $(C_a[2](m), m_{13})$ =1 then $C_a[2](\lambda) = C_a[2](\lambda) + \lambda_{13}$ and m_{13} is removed from the set A. If the measure is not similar with any of the existing measure in the array we create a cell $C_a[3]$ for m_{13} affected by its weight λ_{13} before we remove $(\lambda_{13}, m_{13)}$ from the set A. Instead of looping through the entire set of values in A, we are only scanning the cells progressively created in the dynamic array list while computing the Substitution function. If the latter is not verified then we create a new cell containing the measure with its related weight otherwise we are only adding the weight to an existing slot as in Table II.

```
Algorithm2: Second  Tier Data Cleaning.
Input:
N: number of nodes associated to one aggregator A.
K: number of measurements received by the aggregator A.
A= $(\cup_{nk}\lambda,m| n\in N, k\in K )= \{(\lambda_{nk}, m_{nk})| n\in N, k\in K\} = \{(\lambda_{11}, m_{11}), (\lambda_{12}, m_{12}) ,.., (\lambda_{21}, m_{21}), …., (\lambda_{nk}, m_{nk})\}$.
 Output:
Final dataset sent to the sink.
Initialization:
We create a cell $C_a[1]$ which contain a random value from the set A.
L←K
T←1  //T is the number of cells created
For i ←2 to L do
   Remove←False
   For j=1 to T do
   Compute Substitution($C_a[j](m)$,A[i](m))
   If Substitution($C_a[j](m)$,A[i](m)) =1 Then
            $C_a[j](\lambda) \leftarrow C_a[j](\lambda) + A[i](\lambda)$
            Remove A[i]($\lambda$, m) from the set A
            Remove←True
  End if
  End For
 If remove ←False Then
            Build a cell $C_a[j+1]$ to contain A[i]($\lambda$, m)
            Remove A[i]($\lambda$, m) from A
            Remove←True
End if
L← length (A)
T← number of cells created for an aggregator.
End For
Send to the sink the built Array list of measures and weights.
```

TABLE II. SAMPLE OF THE RESULT SET SENT TO THE AGGREGATOR

Cells	$C_a[1]$	$C_a[2]$
Weight	$\lambda_{11,} m_{11}$	$(\lambda_{12}+\lambda_{13}),m_{12}$

IV. EXPERIMENTAL RESULTS

To validate the approach presented in this paper, we developed a C# based simulator that we ran on the readings collected from 46 sensors deployed in the Intel Berkeley Research Lab [13]. Every 31 seconds, sensors with weather boards were collecting humidity, temperature, light and voltage values. In our experiments, we are interested in two sensors measurements: the temperature and the humidity. Each node reads an average of 83000 values of each measurement per day and per field. Our approach consists of a two tiers aggregation: (1) first tier where the aggregation is done on a periodic basis every 31 seconds (2) second tier where the aggregation is done on the level of the aggregator that receives the input from a group of nodes.

A. *First tier: periodic data aggregation*

At the first tier, data is filtered on a periodic basis where each period is constituted of 31 seconds. At each period, each measure is affected by its weight. The result depends from the threshold delta that we choose to vary between 0.01 and 0.07 based on the variation of measurements. Fig. 4 shows the percentage of data sent to the aggregator. Obviously the data size is disproportional to the threshold data. The goal of this tier is to reduce the size of the data collected by each node while preserving the frequency of each value as to not affect the analysis on the sink level. The experimental results show that a minimum of 5% of the total set for each measure remains. The size of the affected probability for each value is equal to the number of items existing in the message to be sent to the aggregator. The total size of the messages sent to the aggregator is then equal to the total number of measures to be sent in addition to the total number of affected probability. As per the experimental results displayed in Fig. 2, minimum of 10% for each measure is sent to the aggregator.

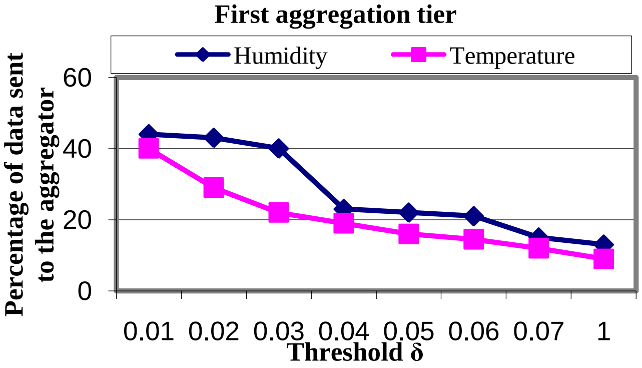


Figure 2. First Tier Data Aggregation

B. *Second Tier: Group weighted data aggregation*

At this level, sets of weighted data measures are received by the aggregator. The cleaning on the level of the aggregator can't ignore the weight of each measure. Weighted data aggregation between sets is performed at the level of the aggregator taking as input the sets received and giving as output one reduced set containing the cleaned measure associated with their weight. The weight of each measure can define the probability of the s data measure existence in this aggregator. Result in Fig. 3 shows that maximum 13% of the data is sent to the sink adding to it 13% related to their respective probability of occurrence. We conclude that only 26% of the size of messages received by each aggregator A will be sent to the sink.

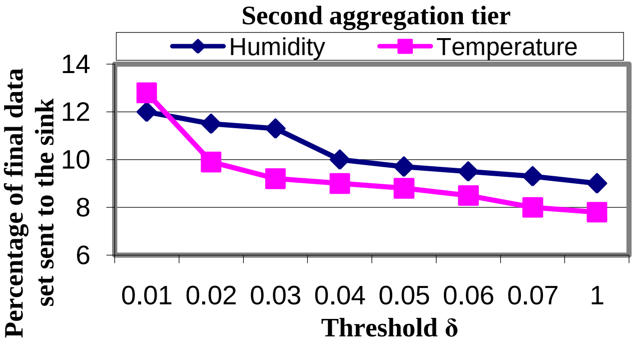


Figure 3. Second Tier Data Aggregation

C. *Energy study*

Sensor nodes that are used to form a sensor network are normally operated by a small battery which has small amount of energy. Therefore, in wireless sensor networks reducing energy consumption of each sensor node is one of the prominent issues to address in the network lifetime, since wireless communications consume significant amount of battery power, sensor nodes should be energy efficient in transmitting data. Protocols can reduce transmitted power in two ways. First where nodes can emit to short distances such as data sinks or cluster nodes. The cluster node can then send the data over a larger distance preserving the power of the smaller nodes. The second is by reducing the number of bits (amount of data) sent across the wireless network. Our approach reduces the overhead by detecting and cleaning redundant measures while preserving the information integrity. To evaluate the energy consumption of our approach we used the same radio model as discussed in [20]. In this model, a radio dissipates Eelec = 50 nJ/bit to run the transmitter or receiver circuitry and βamp = 100 pj/bit/m^2 for the transmitter amplifier. The radios have power control and can expend the minimum required energy to reach the intended recipients as well as they can be turned off to avoid receiving unintended transmissions. The equations used to calculate transmission costs and receiving costs for a k-bit messages and a distance d are respectively shown below in (1) and (2) :

$$E_{TX}(\kappa, \delta) = E_{elec} * \kappa + \beta_{amp} * \kappa * d^2. \qquad (1)$$

$$E_{RX}(\kappa, \delta) = E_{elec} * \kappa. \qquad (2)$$

Receiving is also a high cost operation, therefore, the number of receptions and transmissions should be minimal. In our simulations, we used a measure length k of 64 bits which, corresponds to a packet length. With these radio parameters, when d2 is 500m2, the energy spent in the amplifier part is equal to the energy spent in the electronics part, and therefore, the cost to transmit a packet will be twice the cost to receive.

At the first level, and at the end of the period, each node will contain m messages affected each by a weight λ. The size of the message sent by each node is equal to the number of weight sent in addition to the number of values sent. We consider that each value is equal to 64 bits. The total energy consumed is equal to the sum of the energy consumed by each node when the packet is sent to the aggregator from the source nodes and can be calculated as follows:

$$Eagg\ (\kappa, d) = \Sigma E_{Tx}\ (\kappa, d) + EP_{x}\ (\kappa) = \Sigma(E_{elec} * \kappa + \beta amp * \cdot\ \kappa * d^{2}) + E_{elec} * \kappa. \qquad (3)$$

At the second level, the energy consumption will be equal to the energy consumed when the aggregator send the data to the sink in addition to the energy consumed by the sink when receiving the data as shown in (4).

$$E_{sink}\ (\kappa, d) = \Sigma\ E_{Tx}\ (\kappa, d) + E_{Rx}\ (\kappa) = (\ E_{elec}\ * \kappa + \beta_{amp} * \kappa * d^{2}) + Eelec * \kappa. \qquad (4)$$

The total energy consumed on the level of the network is calculated as follows:

$$E = E_{agg}\ (\kappa, d) + \ Esink\ (\kappa, d). \qquad (5)$$

To evaluate the energy consumption of our approach we compared it to a classical clustering approach, where every node sends all its measures to a cluster head which, in his turn relays all the received data to the sink. Fig. 4 shows that our approach outperforms clustering approaches and minimizes the energy consumption by at least 50%.

Our approach is efficient since the information integrity is fully preserved. All taken measurements appearing in the final set arrived to the sink along with their weight. Therefore, we can consider that our approach decreases the amount of redundant data forwarded to the sink and performs an overall lossless process in terms of information and integrity by conserving the weight of each measure.

V. Conclusion

Data aggregation is a well known technique to achieve energy efficiency, in wireless sensor networks, when propagating data from sensor nodes to the sink. The main idea behind is that rather than sending all captured data from sensors to the sink, multiple redundant data are aggregated as they are forwarded by the sensor network. In our approach, we proposed two-tiers weighted periodic data aggregation method. We provided two non complex algorithms that allow at the first level sensor nodes, and at the second level aggregators to identify and reduce duplicate sensor measurements. The experimental results show the effectiveness of our approach in reducing the amount of redundant data; furthermore, we confirm that the proposed method outperforms existing clustering method in terms of energy consumption.

As part of our future work, we plan to show the effectiveness of this approach in data mining and how the weighted measures in the training set will serve the classifier predicting with reasonable accuracy the class of each instance fed.

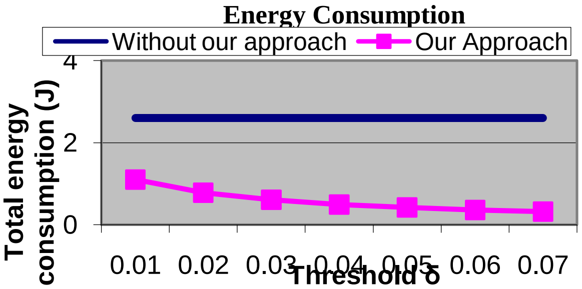


Figure 4. Energy Consumption.

VI. References

[1] G. Pottie and W. Kaiser, “Wireless integrated network sensors," Communications of the ACM, vol. 43, no. 5, p.51C58, 2000.

[2] Y. Zhuang and L. Chen Hong Kong, “In-network Outlier Cleaning for Data Collection in Sensor Networks,” Proceedings of the First International VLDB Workshop on Clean Databases,(CleanDB06), 2006.

[3] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, “Wireless Sensor and Actuator Networks,” Academic Press/Elsevier, London, 2008.

[4] R. Rajagopalan and P. K. Varshney, "Data-Aggregation Techniques in Sensor Networks: A Survey," IEEE Communication Surveys and Tutorials, Vol. 8, No. 4, pp. 48-63, December 2006.

[5] X. Li, "A Survey on Data Aggregation in Wireless Sensor Networks," Project Report for CMPT 765, Spring 2006.

[6] A. Bakhtiar Qutub, N. Pissinou, and K. Makki, “Belief based data cleaning for wireless sensor networks,” Wireless Communications and Mobile Computing, 11: n/a. doi: 10.1002/wcm.970, 2011.

[7] A. Mehdi Esnaashari and M. R. Meybodi, “ Data aggregation in sensor networks using learning automata,” Wireless Networks, 16(3):687–699, 2010.

[8] R E. Elnahrawy and B. Nath, “Cleaning and querying noisy sensors,” In Proceeding of International Workshop of Wireless Sensor Networks and Applications (WSNA), 2003.

[9] B.J. Chen, K. Jameison, H. Balakrishnan, and R. Morns, Span: “An Energy Efficient coordination Algorithm for Topology Maitnenance in Ad-Hoc Wireless Networks,” Wireless Networks 8(5):481-494, 2002.

[10] H. Albert, R. Kravets and I. Gupta, "Building Trees Based On Aggregation Efficiency in Sensor Networks," Ad Hoc Networks, Vol. 5, No. 8, November 2007, pp. 1317-1328.

[11] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network Aggregation Techniques for Wireless Sensor Networks: A Survey, " IEEE Wireless Communications, Vol. 14, No. 2, pp. 70-87, April 2007.

[12] Y. Zheng, K. Chen, and W. Qiu, “Building Representative-Based Data Aggregation Tree in Wireless Sensor Networks, Mathematical Problems in Engineering, vol. 2010, Article ID 732892, 11 pages, 2010.

[13] http://db.csail.mit.edu/labdata/labdata.html, Apr. 2004, Last access March 2011.

An Energy Efficient and Trusted Data Fusion by using Cellular Automata in Wireless Sensor Networks

Shaghayegh Jaberi
Department of Computer Engineering
Science and Research Branch, Islamic Azad University
Tehran, Iran
sh.jaberi@gmail.com

Amir Masoud Rahmani
Department of Computer Engineering
Science and Research Branch, Islamic Azad University
Tehran, Iran
rahmani@sriau.ac.ir

***Abstract*—Wireless sensor networks are becoming more and more common. One of the limitations of wireless sensor nodes is their inherent limited energy resource. Besides maximizing the lifetime of the sensor node, it is preferable to increase the trust value of data fusion results. In this paper, a new protocol is introduced, named EETDFCA (an Energy Efficient and Trusted Data Fusion by using Cellular Automata) in Wireless sensor Networks. EETDFCA uses cellular automata rule to find the most suitable cluster head, perform data fusion, find the most trusted neighbors for sending the fusion result to base station, and transforms from current state to a new state. The network is intended for the long-term monitoring of packets produced by jammer nodes. The data flow of the network is mainly toward a cluster head node, which is responsible for collecting data generated by sensor nodes. When the network is first deployed, an initialization algorithm is performed and preliminary clusters, cluster heads and sensors alive are determined. Simulations and results show that the algorithm can extend the lifetime of the wireless sensor network and boost trusted data fusion frequency.**

Keywords: Wireless Sensor Network; Energy Efficient; Cellular Automata; Trust value; clustering.

I. INTRODUCTION

Wireless sensor networks (WSNs) have been used increasingly in every type of environment due to their ease of deployment. WSNs provide their users which fast and easy access to their data and services anytime and anywhere, especially in remote area such as battlefield, forest and volcano. WSNs have limitations such as: limited energy resources, battery life, computation, and communication capacities, and high cost of transmission. All of these characteristics of wireless sensor networks are complete opposites of their wired network counterparts, in which energy consumption is not an issue, transmission cost is relatively cheap, and the network nodes have plenty of processing capabilities.

In addition, as in many other kinds of network or communications system, the data and services provided require protection. However, WSNs are more vulnerable to security attacks than other traditional networks and Ad Hoc networks due to their unattended nature. For example, an adversary can physically capture some nodes and use them to inject faulty or false data into the network system disturbing the normal cooperation among nodes. Cryptographic and authentication mechanisms, such as TinySec [1] and TinyPK [2], alone cannot be used to solve this problem as internal adversarial nodes will have access to valid cryptographic keys. Moreover, in WSNs, the function of data fusion is mostly shown on saving energy, improving data collection efficiency, enhancing data accuracy, and getting synthesis information [3]. The typical data fusion algorithms, such as Data Funneling, AIDA [4], TAG [5], and TINA [6], do not pay attention to correctness of received data. In addition, clustering is another way to reduce the energy consumption in WSNs. By associating each sensor to one cluster, sensor sends its sensed data to the cluster head of that cluster. Therefore, instead of each sensor sending its message to base station, it sends it to the cluster head; so clustering leads to reducing the message conveyed on the network.

This paper uses cluster heads as fusion nodes and combines data fusion with trust calculation by using a secure data fusion algorithm based on its neighbors' behavioral trust. In addition, selecting cluster head for each round and seeking next-hop after calculating the fusion results are based on cellular automata's rule. Moreover, each cluster head punishes or motivates its neighbors based on their activity and sending accurate data. Moreover ,it lead to sensors transform from its current state to a new state based upon its current state and the states of its neighbors, according to cellular automata's rule to achieve the energy efficiency which is the main purpose of this paper . In Section II, briefly describes the related work about energy efficient data fusion protocols for WSNs and existing network simulators. Section III presents brief description of the network model. Section IV presents the network algorithms. Suggested protocol is discussed in this section. Sections V and VI present results of simulation and conclusion, respectively.

ISBN: 978-1-61208-144-1

II. RELATED WORKS

Wireless sensor networks have attracted a plethora of research efforts due to their vast potential applications [7]. In particular, extensive research work has been devoted to providing energy efficient routing algorithms for data gathering [8-12]. While some of these approaches assume statistically independent information and have developed shortest path tree based routing strategies, others have considered the more realistic case of correlated data gathering [9-11]. By exploring data correlation and employing in-network processing, redundancy among sensed data can be curtailed and hence the network load can be reduced [8]. The objective of sensor routing algorithms is then to jointly explore the data structure and network topology to provide the optimal strategy for data gathering with as minimum energy as possible

Then it is a critical consideration to collect and fuse sensed information in an energy efficient manner for obtaining a long lifetime of the sensor network. Based on researchers' findings that the conventional methods of direct transmission, shortest path routing, and Dempster-Shafer tool may not be optimal for data fusion of sensor networks, in [13] Low-Energy Event Centric Fusion (LEECF) is proposed, a event-centric-based protocol that utilizes the centric sensor node to aggregate the event data among the triggered sensors in a short delay. LEECF incorporates a fast information fusion into the routing protocol to reduce the amount of information that must be transmitted to the sink and the time complexity of fusion computation of fusion center. Simulations show that LEECF can decrease the energy and fusion time significantly compared with conventional routing protocols.

Many routing, power management, and data dissemination protocols have been specially designed for WSNs where energy awareness is an essential design issue. The focus, however, has been given to the routing protocols which might differ depending on the application and network architecture In general; routing in WSNs can be divided into flat-based routing, hierarchical-based routing, and location-based routing depending on the network structure. In flat-based routing, all nodes are typically assigned equal roles or functionality. In hierarchical-based routing, however, nodes will play different roles in the network. In location-based routing, sensor nodes' positions are exploited to route data in the network. A routing protocol is considered adaptive if certain system parameters can be controlled in order to adapt to the current network conditions and available energy levels. Furthermore, these protocols can be classified into multipath-based, query-based, negotiation-based, QoS-based, or coherent-based routing techniques depending on the protocol operation. In addition to the above, routing protocols can be classified into three categories, namely, proactive, reactive, and hybrid protocols depending on how the source finds a route to the destination. In proactive protocols, all routes are computed before they are really needed, while in reactive protocols, routes are computed on demand. Hybrid protocols use a combination of these two ideas. When sensor nodes are static, it is preferable to have table driven routing protocols rather than using reactive protocols. A significant amount of energy is used in route discovery and setup of reactive protocols. Another class of routing protocols is called the cooperative routing protocols. In cooperative routing, nodes send data to a central node where data can be aggregated and may be subject to further processing, hence reducing route cost in terms of energy use. Many other protocols rely on timing and position information. We also shed some light on these types of protocols in this paper. In order to streamline this survey, we use a classification according to the network structure and protocol operation (routing criteria).

Currently, Agent-based Modeling and Simulation is the only paradigm which allows the simulation of even complex behavior in the environments of Wireless sensors [14]. Simulators like QualNet [15], OPNET Modeler [16], NetSim [17] and NS2 [18] can be used to simulate Wireless Sensor Networks. Other simulators, like IDEA1 – based on SystemC – have hardware-level libraries that permits system-level simulations by taking low-level constraints into account. NS (from network simulator) is a name for series of discrete event network simulators, specifically NS2 and NS3. Both simulators are used in the simulation of routing protocols, among others, and are heavily used in ad-hoc networking research, and support popular network protocols, offering simulation results for wired and wireless networks alike then it is suitable to write CA rules, define different types of nodes that are used in this simulation, and simulate new routing protocol.

III. NETWORK MODEL

The network consists of four types of nodes:

- Sensor nodes: These fixed nodes collect data using their sensors. The collected data are then passed to the cluster head nodes through the network. There may be as many sensor nodes as needed depending on the area to be covered.
- Cluster Head nodes: Fixed cluster head, which has the same design as sensor node, just do some extra activities. These nodes are responsible for collecting data from sensor nodes, performing data fusion algorithm on them, and producing fusion result on their own cluster; run the Find Routing algorithm to find its most trusted neighbor for sending fusion result; and in the end perform Find Cluster Head algorithm to determine new cluster head for later round.
- Jammer nodes: These mobile nodes produce disturbed packet throughout the network and are known as intruders.
- Base station: Data from each cluster are gathered and transferred to a central base station.

The sensor nodes, which receive disturbed packet that is produced by jammer nodes, send a data packet to their cluster head. By using the received data packets, cluster heads make decision about the existence of jammer node and send the final result to base station. The cluster heads are connected to the base station in a hierarchical manner.

Each cluster head send the final result about existence of jammer to its trusted neighbor to send it to the nearest cluster head to base station.

IV. ENERGY EFFICIENT AND TRUSTED DATA FUSION BY USING CELLULAR AUTOMATA (EETDFCA) PROTOCOL

In this section, Trusted Data Fusion by using Cellular Automata (TDFCA) [19] is introduced in terms to increase the trusted data fusion frequency; as can be seen in [19] by using TDFCA WSN has more coverage, longer life time, and greater trusted data fusion in comparison to BCDCP [20]; by using TDFCA,WSN's trusted data fusion frequency will be greater than using Secure Data Fusion Algorithm Based on Behavior Trust [3] scenario; and finally network's lifetime, End-to-end delay, and Packet delivery ratio noticeably decreases, using AODV instead of the TDECA. Therefore, although TDFCA increases the trusted data fusion frequency, it has the higher priority on decrease network's lifetime. In this paper, a new protocol is introduced, named EETDFCA (Energy Efficient and Trusted Data Fusion by using Cellular Automata) in Wireless sensor Network which its main purpose is to extend the network's lifetime and address the main problem of TDFCA protocol (pseudo-codes of the first three algorithms are available in [19]).

A. *Initialization*

Since the network in consideration is an ad hoc network, an initialization algorithm is needed to establish preliminary connections autonomously. The algorithm is based on polling and as such it guarantees connectivity to all the nodes that are acoustically reachable by at least one of their nearest neighbors. During initialization, the nodes create *neighbor tables*. These tables contain a list of each node's neighbors and the premiere clusters and cluster head. The initialization steps can be listed as follows:

All sensor nodes are alive and broadcast a find_premiere_cluster head packet to all the sensors in its transmitted radius.

Each sensor has a variable named *#neighbors*. By receiving find_premiere_cluster head packet adds one to this variable and adds a new row to its *neighbor* table. This table has three columns, the first one is the id of its neighbors filling by the ID field of receiving packet, which contain sender node's x-axis and y-axis, second column is a place to save the data sent by neighbors, and the last one is for saving $normalized_{trustvalue}$. In the beginning of the simulation, the data are set to -1 and the $normalized_{trustvalue}$ set to 1. If the sensor does not receive any packet for ΔT second, then the next step is performed.

Each sensor must calculate the density and distance to the base station by using the following equations:

$$density= (\#neighbors) /2r^2 \qquad (1)$$

$$distance=\sqrt{} (pos_x- b.s_x)^2+(pos_y- b.s_y)^2 \qquad (2)$$

Where r is transmitted radius, pos_x is the x_axis and pos_y is the y_axis of current sensor, $b.s_x$ is the x_axis and $b.s_y$ is the y_axis of base station. All sensors know base station's x_axis and y_axis. Then, calculate apt, which shows its convenience to be cluster head.

$$apt=\alpha_1\ density +1/\beta_1\ \ distance+\mu_1\ trst\text{-}val \qquad (3)$$

Where α_1, β_1 and μ_1 are empirical coefficients that define the simulation. The *trst_val* is trust value of sensor node, which is evaluated by using (4) that is explained in following section.

Then, the result of (3) sends by a packet, whose name is find_max_apt to all neighbors of a sensor. By receiving this packet, each sensor performs the Find Cluster Head algorithm to determine whether or not it is a candidate of being this round cluster head. This algorithm is based on cellular automata (CA) rule and makes a decision locally by determining a flag, namely ch_flag. Therefore, it runs for about ΔT second after receiving the first packet.

By receiving the ch_announce packet, each sensor saves the new cluster head's ID.

B. *Data Fusion*

When a sensor node detects the intruder noise, it will compare its $normalized_{trustvalue}$ with normal_trustvalue of its neighbors; if its $normalized_{trustvalue}$ is greater than majority of $normalized_{trustvalue}$, then it generates data packet and sends it to the cluster head (CA rule).

When the cluster head gets this packet, first it updates the $normalized_{trustvalue}$ of the sender neighbor and then performs Data Fusion algorithm to make its cluster decision, which determines whether its cluster detects the intruder noise or not and sends the final decision to base station.

Data Fusion algorithm has two phases, first get the $normalized_{trustvalue}$ of sender node, which is evaluated by using (4) and (5) in individual sensor nodes as follows:

$$trst\text{-}val=\alpha_2\ r\text{-}power+\beta_2\ ch_t+\delta \qquad (4)$$

$$normalized_{trustvalue}=(trst\text{-}val)/(\delta+\alpha_2 r\text{-}power) \qquad (5)$$

where α_2 and β_2 are empirical coefficients that define the simulation; δ is motivation or punishment coefficient of the cluster head after assessing the final_result and compares it with data sender data, then motivates them if they send the same data as final_result and otherwise punishes them; *r_power* is amount of remaining power to maximum amount of power, (6) divided to maximum amount of power of sensor node; and finally, ch_t is the number of times that this sensor is selected as a cluster head.

$$E_t(b,d_2)=E_{tx}b+b\epsilon_1 d_2^{\,2}+processing\ energy \qquad (6)$$

That b is number of bit sent or received by a sensor, d_2 is distance between the sensor and cluster head, E_{tx} is constant determined by simulation, and ϵ_1 is transmit amplifier.

Cluster head multiplies the received data by *normalized*$_{trustvalue}$. Then, for evaluating that, the fusion result is trusted or not, compares the result of first step to threshold, TL [21].

Therefore, cluster head makes decision and then must create final_result packet and send it to base station using multi hop strategy and make pun_mot packet and send it to sender sensor to punish or motivate them.

At the end of each round of data fusion, it must compare the amount of remaining power with the lower amount of power that requires to do cluster head duties. If the remaining power, assessed by (7), is less than the amount required to cluster head's duties or is equal to zero, the Find Cluster Head algorithm must be performed.

$$E_t(b,d_1)=bE_{tx}x+b\,E_{da}\,x+b\,\epsilon_2\,d_1^4+processing\ energy \qquad (7)$$

That E_{da} is data aggregation energy, x is number of received data, d_1 is distance between cluster head, and next hop will be detected in next part.

C. Candidate trusted neighbor to receive data fusion results

The other crucial issue is finding the most trustable sensor for conveying the final_result from cluster head to base station. Each sensor, which is nearer to base station in comparison with its cluster head finds the maximum *normalized*$_{trustvalue}$ among its neighbors, compares it with its own value if its *normalized*$_{trustvalue}$ is bigger than maximum value, then candidates itself as next hop to receive fusion result (CA rule).

Therefore, the most trustable receiver node is available for each sensor.

D. Update Trust value

By receiving a pun_mot packet, changing the remaining power or the number of times that this sensor is selected as a cluster head, or the parameters that play a pivotal role in trust value, each sensor must update its normalized$_{trustvalue}$ using formulas (4) and (5). Whenever the value of *normalized*$_{trustvalue}$ would be changed, this new value must be broadcasted urgently. Then, its neighbors replace old value of *normalized*$_{trustvalue}$ with received *normalized*$_{trustvalue}$ in their neighbor lists. Therefore, the trust values in their list are always updated.

E. Make decision about transforming state

As mentioned before to extend the network's lifetime and increase the number of active nodes in wireless sensor networks, this new protocol is introduced. The assumption is each sensor has two states one is alive, which sensor sends or receive packets and performs the entire algorithms (clustering, data fusion and routing) and the other is standby , which sensor does not send any packet just received them and discards all the packet except for packets contain updated *normalized*$_{trustvalue}$ -after receiving this packets sensor updates its neighbor tables, therefore this table always is up to date -, ch_announce packets- by receiving these packets sensor changes cluster head axis, then the sensor always knows cluster head and cluster that belongs to -, and standby packets, which indicate that one of its neighbors transforms from alive state to a standby state.

At the beginning of simulation all the sensors are alive, after the first change is happened in its *normalized*$_{trustvalue}$; they execute change state algorithm, which is indicated in Table I. The sensor will compare its *normalized*$_{trustvalue}$ by *normalized*$_{trustvalue}$ of its neighbors; if its *normalized*$_{trustvalue}$ value is less than majority of *normalized*$_{trustvalue}$, then it transforms from alive to a standby state (CA rule). It will remain in this state and updates the *normalized*$_{trustvalue}$ of its neighbors and cluster head, which is belong to until it receives the standby packet, which indicates that one of its neighbors transforms from alive state to a standby state then it must perform change state algorithm again.

Each sensor, the state of which is set to alive must compare its *normalized*$_{trustvalue}$ by its neighbors whenever changes is happened to its *normalized*$_{trustvalue}$ or in neighbor tables ; if its *normalized*$_{trustvalue}$ value is less than majority of *normalized*$_{trustvalue}$, then it transforms from alive to a standby state (CA rule) and send standby packet to its neighbors.

TABLE I. CHANGING STATE

```
Data : $normalized_{trustvalue.i}$ and neighbor $table_i$
Result : changing sensor state
if change is happened in $normalized_{trustvaluei}$ or neighbor $table_i$ then
    if $normalized_{trustvalue.i} < = (\sum normalized_{trustvalue.neighbors}/2)$ then
        $sensor_{state}$ = standby;
    else
        $sensor_{state}$ = alive;
    end
end
if standby packet is received then
    if $sensor_{state}$ == alive then
        do nothing;
    else
        if $normalized_{trustvalue.i} < = (\sum normalized_{trustvalue.neighbors}/2)$ then
            do noting;
        else
            $sensor_{state}$ = alive;
        end
    end
end
```

By using this algorithm, the amounts of consumption energy is decreased and network's lifetime is extended.

V. SIMULATION RESULTS

In what follows, results of the new protocol will be described by using NS2 simulator. First, it is necessary to use some assumptions: the network has N fixed sensors that propagate in the area that is L meter square (L×L m^2). Sensors used here also have the same structures and attributes, and then the network is homogeneous and they are trustable and do not show any intruding behaviors. Two different scenarios are used for evaluating this proposed protocol. Then, the new protocol will be assessed in tow region whose scales are 100×100 m^2 (having 10,000, 5000, and 2500 sensors) and 250×250 m^2 (having 62,500, 31,250, and 15,625 sensors). The sensors have a power, which charges up 0.8j, have a timer that is set to 200 ms, and the maximum transmission radius is about 6m (R_{max}=6m). This radio transmission updates by using (8) whenever the remaining power evaluates; in this formula α is coefficient.

$$r\_power = \alpha R_{max} \quad (8)$$

EETDFCA with TDFCA will be compared in terms of three parameters: total energy, coverage, and trusted data fusion frequency.

A. Total Energy

As mentioned above, one of the crucial problem in TDFCA is network's lifetime, which is noticeably decreased because of massive amount of packets sending and receiving by sensor node; Using EETDFCA instead of TDFCA owing to fewer energy consumption by changing the state of sensor node to standby in which the sensor just receives packets and do not participate in sending them. Figure 1 shows the total energy in network. As can be seen in Fig. 1, in all different scenarios EETDFCA has more energy in comparison with TDFCA this fact end in longer networks' lifetime by using changing state algorithm. Figure 1 shows that the effectiveness in teams of energy consumption in EETDFCA protocol is 1.2 times greater. Therefore, it seems that EETDFCA almost achieves to extend network's lifetime.

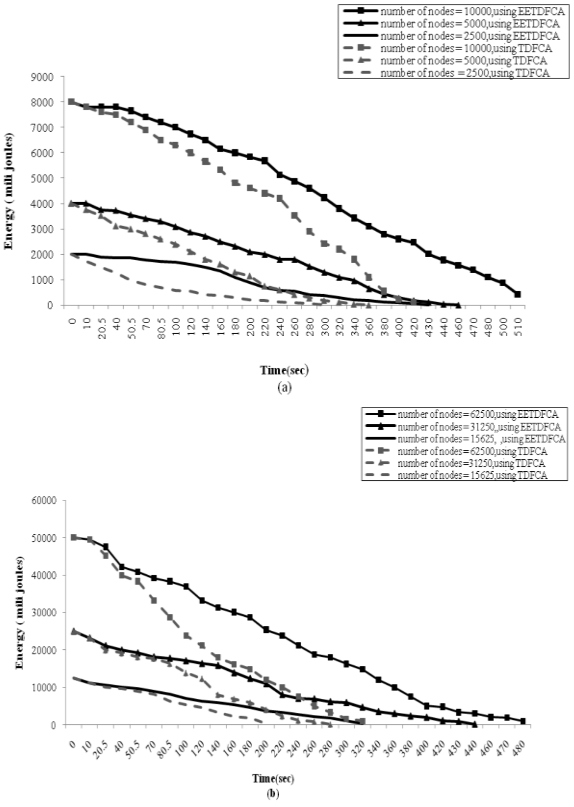


Figure 1. Remaining energy in the network (a) in first scenario (100*100) (b) in second scenario (250*250) EETDFCA in comparison with TDFCA

B. Coverage

As mentioned above in the proposed protocol, changing remaining power of a sensor lead to calculation of transmission radius. Therefore, by using EETDFCA changes are happening in the remaining power of a sensor seem to be happening at less slow a pace; then it has more coverage and can keep it longer in comparison with TDFCA that does not pay attention to sensor's state, then to some extend the coverage decreased. As can be seen, figures Fig. 2(a) and Fig. 2(b) are the same. This similarity leads to the fact that the scale of region does not have direct effect on coverage.

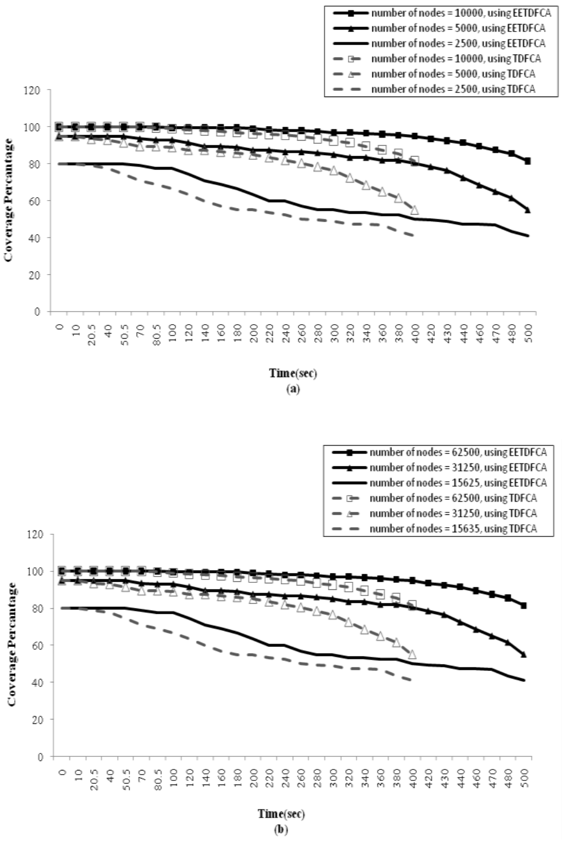


Figure 2. Relationship between coverage and time (a) in first scenario (100*100) (b) in second scenario (250*250) EETDFCA in comparison with TDFCA

C. Trusted data fusion frequency

Whatever data sent to cluster head have higher trust value, the fusion result will be more trusted. By using the both protocols, only when the sensors having higher trust value will send their data to the cluster head; then data fusion is more trusted. Both protocols use this method. Then, the difference between these two protocols is related to standby sate, which has no knock on effect on the trusted fusion results. Similar to coverage, trusted data fusion frequency is same for all regions with different scales, hence just one scenario is shown, Fig. 3. As can be seen in [4] trusted data fusion frequency dramatically increase in comparison with

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

BCDCP and Secure Data Fusion Algorithm Based on Behavior Trust.

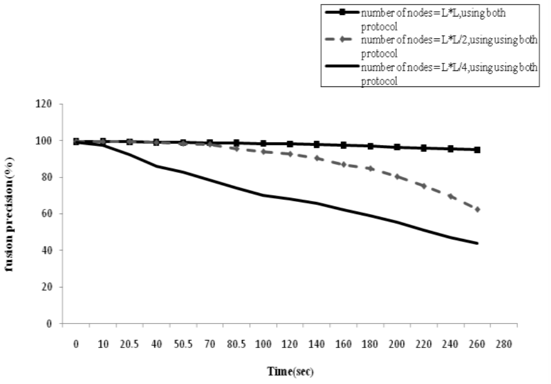


Figure 3. Trusted Data Fusion frequency in all scenarios

VI. CONCLUSION ND FUTURE WORKS

The suggested protocol, named EETDFCA, based on Cellular Automata for a wireless sensor network was tested by NS2. The simulation results show that EETDFCA protocol expands the network's lifetime, and coverage dramatically, and has no effect on trust to fusion result in comparison with TDFCA, in wireless sensor networks simultaneously.

The importance of decreasing energy consumptions and increasing trust value, especially in data fusion, in WSNs leads to various works on it. Then, the following changes are suggested for this protocol to assess its efficiency:

1. Implement this protocol in immobile WSNs.

2. Implement this protocol in WSNs whose their sensors have intruding behavior.

3. Using other types of cellular automata instead of totalistic ones, which are used in this paper, to evaluate this protocol.

REFERENCES

[1] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: a link layer security architecture for wireless sensor networks," Proc. second ACM Conference on Embedded Networked Sensor Systems(SensSys 2004), Nov. 2004, pp. 162-175, doi:10.1145/1031495.1031515.

[2] R. Watro, D. Kong, S.F. Cuti, C. Gardiner, C. Lynn, and P. Kurus. "TinyPK: secure sensor networks with public key technology," , Proc. 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, Oct. 2004, pp. 59-64, doi: 10.1145/1029102.1029113.

[3] Z. Cheng, Z. Ming-zheng, X. Jin-sheng, and Y. Qing, "A secure data fusion algorithm based on behavior trust in wireless sensor networks security," 5th International Conference on Wireless Communications, Networking and Mobile Computing(Wicom 08), Oct. 2008, pp. 1-4, doi: 10.1109/WiCom.2008.1106.

[4] T. He, B. M.Blum, J. A.Stankovic, and T. Abdelzaher, "AIDA: Adaptive Application Independent Data Aggregation in Wireless Sensor Networks," ACM Transaction on Embedded Computing System (TECS), vol. 3, May 2004, pp. 426-457, doi: 10.1145/993396.993406.

[5] S. Madden, M. J . Franklin, J. M. Hellerstein, and W. Hong, "TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks," ACM SIGOPS Operating Systems Review - OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation, vol. 36 , Dec. 2002, doi : 10.1145/844128.844142.

[6] M.A. Sharaf, J. Beaver, A. Labrinidis, and P. K.Chrysanthis, "TINA: A Scheme for Temporal Coherency Aware in Network Aggregation," Proceedings of the 3rd ACM international workshop on Data engineering for wireless and mobile access(MobiDE03),Sep 2003, pp. 69-76, doi : 10.1145/940923.940937.

[7] C. Chong and S. Kumar, "Sensor networks: Evolution, opportunities, and challenges," Proceedings of the IEEE, Aug. 2003,vol. 91, pp. 1247-1256, doi: 10.1109/JPROC.2003.814918.

[8] B. Krishnamachari, D. Estrin, and S. Wicker, "Impact of data aggregation in wireless sensor networks," the 22nd International Conference on Distributed Computing System (ICDCSW), July 2002, pp. 575-578.

[9] S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," Information Processing in Sensor Networks 2004 (IPSN'04), Apr. 2004, pp. 28-35, doi: 10.1109/IPSN.2004.1307320

[10] W. Zhang and G. Cao, "Dctc: Dynamic convoy tree-based collaboration for target tracking in sensor networks," IEEE Transactions on Wireless Communication, Sept. 2004, vol. 3, pp. 1685–1701, doi: 10.1109/TWC.2004.833443.

[11] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," IEEE/ACM Transactions on Networking, Feb. 2003, vol. 11, pp. 2-16, doi: 10.1109/TNET.2002.808417.

[12] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," INFOCOM 2004, Mar. 2004, pp. 2571-2582, doi: 10.1109/INFCOM.2004.1354677.

[13] B. Zeng, J. Wei, and T. Hu, "An energy-efficient fusion protocol for wireless sensor network ," 10th International Conference on Information Fusion , july 2007, pp. 1-7, doi: 10.1109/ICIF4408009.

[14] M.A. Niazi and A. Hussain, "A Novel Agent-Based Simulation Framework for Sensing in Complex Adaptive Environments," Sensor Journal IEEE, vol. 11, Nov. 2010, pp. 404-412, doi: 10.1109/JSEN.2010.2068044.

[15] QualNet manual web page available at http://www.scalable-netwoks.com/products/qualnet/, Last access: june 2011 .

[16] Opnet manual webpage available at http://www.opnet.com/, Last acces: june 2011.

[17] NetSim manual webpage available at http://haemgen.haem.cam.ac.uk/netsim/ , Last access: june 2011.

[18] NS manual webpage available at http://www.isi.edu/nsnam/ns/, Last access: june 2011.

[19] Sh. Jaberi, A.M. Rahmani, and A. Khadem Zadeh, "Trusted data fusion by using cellular automata in wireless sensor networks," unpublished.

[20] L. LI, S. Dong, and X. Wen," An Energy Efficient Clustering Routing Algorithm for Wireless Sensor Networks," The Journal of China Universities of Posts and Telecommunications, vol. 13, Sep. 2006. pp. 71-75 , doi : 10.1016/S1005-8885(07)60015-6.

[21] S. Sahnis and X. Chun Xu, "Algorithm for wireless sensor networks,"International Journal of Distributed Sensor Netwoks,vol. 1, Sep. 2004, pp. 35-56,doi= 10.1080/15501320490886323.

[22] A. Stauffer and M. Sipper ,"Biomorphs implemented as a data and signals cellular automaton," European Conference on Artificial Life, vol. 2801/2003, 2003, pp. 724-732, doi = 10.1007/978-3-540-39432-7_78 .

A Column Generation based Heuristic for Maximum Lifetime Coverage in Wireless Sensor Networks

Karine Deschinkel
LIFC Laboratory, Université de Franche-Comté
Belfort, France
Email: karine.deschinkel@univ-fcomte.fr

***Abstract*—Several studies in recent years have considered many strategies for increasing sensor network lifetime. We focus on a centralised management scheme where a large number of sensors are randomly deployed in a region of interest to monitor a set of targets and we propose an adaptive scheduling by dividing sensors into non-disjoint cover sets, each cover set being active in different period of time. In this paper, we design a column generation (CG) method based heuristic for efficiently solving the maximum lifetime coverage problem. We first model the problem with a linear programming (LP) formulation for non-disjoint cover sets where the objective is to maximise the sum of activation times of cover sets, with respect the sensor's battery lifetime. As the number of cover sets may be exponential to the number of sensors and targets, an initial set of cover sets is constructed and other cover sets are generated through the resolution of an auxiliary problem formulated as a integer programming (IP) problem. This problem is either solved at optimality by standard branch-and-bound or solved sub-optimally by a heuristic. Simulation results show the efficiency of the proposed heuristic which provides near optimal solutions while saving computational time.**

***Keywords*-target coverage; wireless sensor networks; centralised method; column generation;**

I. Introduction

Recent years have witnessed significant advances in wireless sensor networks which emerge as one of the most promising technologies for the 21st century [1]. In fact, they present huge potential in several domains ranging from health care applications to military applications. A sensor network is composed of a large number of tiny sensing devices deployed in a region of interest. Each device has processing and wireless communication capabilities, which enable to sense its environment, to compute, to store information and to deliver report messages to a base station. These sensor nodes run on batteries with limited capacities. To achieve a long life of the network, it is important to conserve battery power. Therefore, lifetime optimisation is one of the most critical issues in wireless sensor networks. In this paper we concentrate on the target coverage problem, with the objective of maximizing the network lifetime by using an adaptative scheduling. We assume that sensors are randomly sprayed for monitoring a set of targets with known locations and we also assume the sensors have location determination capabilities. We propose energy-aware centralized method by organizing the nodes in non-disjoint cover sets where each cover set is capable of monitoring all the monitored targets and by activate these cover sets successively. Scheduling and grouping sensors into non-disjoint sets is done by the base station, which informs every sensor of the time intervals to be activated. In this paper, the scheduling problem is formulated as a linear program where the variables are the active times of the different cover sets. The objective is to maximize the sum of their active times which corresponds to the network lifetime such that for any sensor its total active time does not exceed its initial lifetime. Unfortunately the number of cover sets may be huge (exponential in number of sensors and targets). Therefore we develop a resolution method based on a column generation (CG) process which is a well-known and widely practiced technique for solving linear programs with too many variables to include in the initial formulation explicitly. Our main contribution is to design a column generation method based heuristic for efficiently solving the maximum lifetime coverage problem.

The remainder of the paper is organized as follows. Section II reviews the related work in the field. Section III is devoted to the description of the target coverage problem and to its formulation by a linear program and explains the basics of the column generation technique. Next, in Section IV, we present the different algorithms of the proposed scheme. Section V discusses implementation details of our algorithms and shows the simulation results. Section VI concludes the paper.

II. Related work

Many works, including centralised, distributed and localized algorithms, have been proposed to extend the network lifetime. In this paper we focus on centralised algorithms because distributed algorithms are outside the scope of our work. Note that centralised coverage algorithms have the advantage of requiring very low processing power from the sensor nodes which have usually limited processing capabilities. Moreover, a recent study conducted in [2] concludes that there is a threshold in terms of network size to switch from a localized to a centralized algorithm. Indeed

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

the exchange of messages in large networks may consume a considerable amount of energy in a localized approach compared to a centralised one.

The authors in [3] and [4] independently highlight that the use of non-disjoint cover sets may increase the network lifetime by giving appropriate algorithms. For instance, Cardei et al. [3] formulate the maximum set covers problem as a mixed integer programming (MIP) and then apply a relaxation technique to design a LP-based heuristic of time complexity $O(n^3m^3)$ (where m is the number of targets). They also propose a greedy heuristic with a lower time complexity $O(dm^2n)$ (where d is the number of sensors that covers the most sparsely covered target). This heuristic forms individually set covers by covering first the most critical targets as in [5].

In [4], Berman et al. clearly provide a LP formulation for non-disjoint cover sets. We reuse this formulation in our work but instead of developping an approximation algorithm as in [4], we exploit the CG technique to deal with the huge number of variables in the LP formulation. In [4], the authors propose an algorithm with an approximation factor of $(1+\epsilon)(1+2\log n)$ for any $\epsilon > 0$ based on the $(1+\epsilon)$-approximation of the Garg and Könemann algorithm.

More recently Zorbas et al. [6] present a novel algorithm that can produce both disjoint cover sets as well as non-disjoint cover sets by using a cost function taking into account various parameters as the monitoring capabilities of a sensor, its association with poorly monitored targets and also its remaining battery life. Through simulations, they compare their proposed algorithm with other approaches found in [3] and [5] and show that it exhibits comparable results in terms of generated cover sets but in faster execution times.

The closest work to ours are [7] and [8]. In [7] and [8], the authors adress the problem of maximizing lifetime in wireless network subject to QoS, energy or coverage requirements. They propose a CG approach to decompose the original formulation into a master problem and an auxiliary problem. The auxiliary (AUX) problem is an IP problem in [8] or a MIP problem in [7] which is solved at optimality by a branch-and-bound algorithm. In both papers analyses show that the resolution at optimality of the AUX-problem is time-consuming. Based on this observation, we propose in our work a heuristic for adressing the AUX-problem which produces good solutions in lower computational times.

III. Problem description

We try to produce an adaptive scheduling which allows sensors to operate alternatively so as to prolong the network lifetime. For convenience, the notations and assumptions are described first. Then the lifetime problem of sensor network covering a set of targets is formulated.

A. Notations and assumptions

- m : the number of targets
- n : the number of sensors
- I : the set of targets
- K : the set of sensors
- S_i : set of sensors which cover the target i
- Z_k : set of targets covered by the sensor k
- T_k : the lifetime of a sensor k, which is time duration when the sensor k is in the active state all the time

B. Formulation

The problem of monitoring all targets by activate non-disjoint cover sets successively in order to extend the network lifetime can be formulated as a LP. The variables in the LP are as follows. t_u is the lifetime of the cover set u, that means that all sensors in the cover set u are active during the time period t_u. We denote by U the set of all elementary cover sets. The problem is as follows :

$$\begin{cases} \max \sum_{u \in U} t_u \\ \text{subject to :} \\ \sum_{u \in U} a_{ku} t_u \leq T_k, \quad \forall k \in K \\ t_u \in R^+ \end{cases} \tag{1}$$

The objective function maximizes the total work time of all the cover sets. The constraint shows the lifetime constraint for each sensor k. a_{ku} is a binary index which is set to 1 if sensor k is active in the cover set u and 0 otherwise. An elementary cover set corresponds to a configuration where all targets are covered as well as no superfluous sensor is activated. However the number of elementary cover sets is very high.

C. Example

To illustrate our problem we provide a simple example with only 10 sensors and 4 targets. Table I presents the sensors which are able to cover each target. We consider only two cover sets. Each cover set is given as a tuple and covers all targets.

- Cover set 0 = (0,3,9).
- Cover set 1 = (0,4,8).

Target	Sensors
0	3,4,9
1	4,6,9
2	0,2
3	1,3,5,8

Table I
Collection of sensors to monitor a target

The linear programming corresponding to this simple example is the following :

$$\begin{cases} \max \quad t_0 + t_1 \\ \text{subject to :} \\ t_0 + t_1 \leq 1.00 \quad \text{(sensor 0)} \\ t_0 \leq 1.00 \quad \text{(sensor 3)} \\ t_1 \leq 1.00 \quad \text{(sensor 4)} \\ t_1 \leq 1.00 \quad \text{(sensor 8)} \\ t_0 \leq 1.00 \quad \text{(sensor 9)} \\ t_0, t_1 \in (R)^+ \end{cases} \qquad (2)$$

t_0 and t_1 are respectively the lifetimes of the cover sets 0 and 1. The right part of each inequality corresponds to the maximal lifetime of each sensor. Here sensor lifetime is set to 1. In this example we do not enumerate all elementary cover sets, we solve the LP with only two cover sets and the network lifetime obtained is equal to 1 where $t_0^* = 0.5$ and $t_1^* = 0.5$. That means that sensors of cover set 0 are active during 0.5 time unit and sensors of the cover set 1 are active during 0.5 time unit. Note that some sensors (1,2,5,6,7) do not appear in the LP because it is not part of any generated cover sets. Only sensor 0 has consumed its entire energy. If we had generated more cover sets, we would have reached the maximal lifetime of the network which is equal to 2. For instance, if we add cover set 2 = (2,3,4) and cover set 3 =(1,2,9), the optimal scheduling is obtained with $t_0^* = t_1^* = t_2^* = t_3^* = 0.5$.

D. *Column generation method*

As the set U of elementary cover sets may be huge we use a CG technique [9] to solve (1). That means that we solve a Restricted Master Problem (RMP) with only a subset $U' \subseteq U$ of elementary cover sets and we introduce an attractive cover set if necessary. Given a subset $U' \subseteq U$ and the dual multipliers $\pi_k \equiv \pi_k(U')$ for sensors k, the AUX-problem consists in finding the most attractive cover set $u \in U \setminus U'$, that means the cover set u with the maximal reduced cost $r_u = (1 - \sum_{k \in K} a_{ku} \pi_k)$. If $r_u > 0$ then the cover set u is said to be attractive and it is added in the formulation of the RMP, otherwise the problem (1) is optimal.

The AUX-problem is to find a new feasible cover set u which maximizes $r = (1 - \sum_{k \in A} \pi_k)$, or which minimizes $\sum_{k \in A} \pi_k$ where A denotes the set of active sensors in the cover set u. If this sum is less than 1, then the new valid cover set u is added in the Restricted Master Problem. We formulate the AUX-problem as an IP problem with binary variables y_k for each sensor k which is set to 1 if sensor k is active in the cover set u, and 0 otherwise. The following constraints represent the coverage guarantee for each target i ($1 \leq i \leq m$).

$$\sum_{k \in S_i} y_k \geq 1 \quad \forall i \in I$$

The AUX-problem is formulated as follows :

$$\begin{cases} \min \sum_{k \in K} \pi_k y_k \\ \text{subject to :} \\ \sum_{k \in S_i} y_k \geq 1 \qquad \forall i \in I \\ y_k \in \{0, 1\} \qquad \forall k \in K \end{cases} \qquad (3)$$

Note that the formulation of the AUX-problem corresponds to the model of the classical set covering problem [10]. This complete CG approach seems to be efficient. The RMP is formulated as a (LP) (1) where the entire set U of elementary cover sets must be replaced by a subset U' which contains initially a small number of elementary cover sets. Then the RMP (1) and the AUX-problem (3) are solved sequentially and the set U' grows up until no attractive cover set is generated. The optimal solution, that means the adaptive scheduling of cover sets which maximizes the network lifetime, is always found. The RMP is a classical LP problem, thus can be solved in polynomial time $O(n^3 m^3)$ with the algorithm proposed by Ye [11]. However the AUX-problem, which is a IP problem, may require a large unacceptable running time (This problem is also classified as NP-hard [12]). The intuitive idea is to speedup the generation of attractive cover sets by the use of a heuristic. To measure the efficiency of our approach, we design three methods, called respectively the Exact Method, the Heuristic Method and the Mixed Method. The Exact Method consists of solving to optimality the AUX-problem at each step of the column generation process with an IP solver. For the Heuristic Method, we propose to generate an attractive cover set taking into account the dual multipliers of sensors k and without resolving the auxiliary problem at optimality. In the Mixed Method, in case of impossibility for the heuristic to generate an attractive cover set, we solve the (AUX)-problem at optimality. Note that the Exact and the Mixed methods lead to an optimal scheduling compared to the Heuristic Method which provides near optimal solution. The resolution method based on CG technique and its three versions are explained in more details in the following part.

IV. Resolution method

The resolution method requires to generate some elementary cover sets to form the set U' at the beginning. Note that the initial number of cover sets will not affect the final optimal output. The generation of elementary cover sets involves two steps. First, a cover set is generated and is then analyzed to determine if some sensors are not superfluous.

A. *Production of cover set*

The algorithm 1 ensures the production of a cover set where all targets are covered. This algorithm does not produce an elementary cover set because some active sensors could be superfluous. That is why we have to check if it is possible to desactivate some sensors through algorithm 2. This algorithm is applied for each generated cover set whatever the generation process.

Algorithm 1 Cover_Set_Generation(u)

```
Require: A set of targets $I$, a set of sensors $K$
Ensure: A random cover set $u$
  $V \leftarrow I$
  $u \leftarrow \emptyset$
  while $V$ is not $\emptyset$ do
    Select a target $i \in V$ randomly
    $V \leftarrow V \setminus \{i\}$
    Select randomly a sensor $k \in S_i$ to cover the target $i$
    $u \leftarrow u \cup \{k\}$
    for all targets $h \in Z_k$ do
      $V \leftarrow V \setminus \{h\}$
    end for
  end while
```

Algorithm 2 Check_Elementary_Cover_Set(u)

```
Require: A cover set $u$
Ensure: An elementary cover set $u$
  $G \leftarrow u$
  while $G$ is not $\emptyset$ do
    Select a sensor $k \in G$ randomly
    Check if it is possible to desactivate sensor $k$
    if yes then
      $u \leftarrow u \setminus \{k\}$
    end if
    $G \leftarrow G \setminus \{k\}$
  end while
```

B. Generation of an attractive cover set

Once some elementary cover sets are formed and compose the initial set of variables/columns, the CG process consists of introducing new attractive columns in the RMP. This task may be done by the exact resolution of the IP AUX-problem as written in the algorithm 3 or by using a heuristic as described in the algorithm 4.

Algorithm 3 Generation_Attractive_CoverSet_Exact(π,u,r)

```
Require: The dual multipliers $\pi_k$ for each sensor $k \in K$
Ensure: The generated cover set $u$ and the associated
  reduced cost $r$
  $u \leftarrow \emptyset$
  $r \leftarrow 0$
  Solve the IP (3) with dual multipliers $\pi_k$
  $(y_k^*) \quad \forall k \in K$ are the optimal values
  for all $k \in K$ do
    if $y_k^* = 1$ then
      {The sensor $k$ is active}
      $u \leftarrow u \cup \{k\}$
    end if
  end for
  $r \leftarrow 1 - \sum_{k \in u} \pi_k$
```

Here a heuristic is proposed to provide a new cover set u such that all targets are covered. Considering the dual multiplier π_k for each sensor k as a cost, the objective is to activate less expensive sensors in the cover set such that the resulting reduced cost of this cover set is positive. We first select randomly a target, then we choose a sensor with minimal cost that covers this target. We repeat the process until all targets are covered. If there are multiple sensors of minimum costs, the choice of one of them is made randomly. The algorithm 4 of complexity $O(mn)$ presents the generation of an attractive cover set with the heuristic. As our heuristic integrates a random part, it may be applied several times (no more than Nb_Max_Ite iterations) until a cover set with positive reduced cost is found. Note that the two generation methods do not necessarily generate an elementary cover set. Each time an attractive cover set is generated, we call the algorithm 2 to eliminate superfluous sensors.

Algorithm 4 Generation_Attractive_CoverSet_Heuristic(π,u,r)

```
Require: The dual multipliers $\pi_k$ for each sensor $k \in K$
Ensure: The generated cover set $u$ and the associated
  reduced cost $r$
  $V \leftarrow I$
  $u \leftarrow \emptyset$
  $r \leftarrow 0$
  while $V$ is not $\emptyset$ do
    Select a target $i \in V$ randomly
    Select a sensor $k \in S_i$ with minimal cost ($\pi_k$)
    $V \leftarrow V \setminus \{i\}$
    $u \leftarrow u \cup \{k\}$
    $r \leftarrow r + \pi_k$
    for all targets $h \in Z_k$ do
      $V \leftarrow V \setminus \{h\}$
    end for
  end while
  $r \leftarrow 1 - r$
```

C. Global process

The algorithm 5 presents our resolution method based on CG which provides a cover set's scheduling to prolong the network lifetime.

V. Experimental results

Section V is dedicated to experimental results.

A. Experimental setup and environment

We have implemented the three methods presented in section IV. Our experiments have been conducted on a regular Linux workstation with a AMD Athlon(tm) 64 X2 Dual Core Processor 4000+ of 2,1 GHz. Resolution of the LP or IP problems are respectively carried out the simplex method and the branch-and-bound method implemented in

ISBN: 978-1-61208-144-1

Algorithm 5 Resolution Method

```
  $U \leftarrow \emptyset$
  {Generation of E elementary cover sets}
  for $e = 0$ to $E$ do
    Cover_Set_Generation($u$)
    Check_Elementary_Cover_Set($u$)
    $U \leftarrow U \cup u$
  end for
  Restricted_Master_Problem_Resolution($U$)
  $Stop \leftarrow 0$
  while ($Stop = 0$) do
    $r \leftarrow 0$
    {Search of an attractive cover set (3 versions)}
    ------------------------------------------------
    Version 1 : Exact Method
    Generation_Attractive_CoverSet_Exact($\pi$,$u$,$r$)
    ------------------------------------------------
    Version 2 : Heuristic Method
    $Nb\_Ite \leftarrow 0$
    while (($r <= 0$) and ($Nb\_Ite \leq Nb\_Max\_Ite$)) do
      Generation_Attractive_CoverSet_Heuristic($\pi$,$u$,$r$)
      $Nb\_Ite \leftarrow Nb\_Ite + 1$
    end while
    ------------------------------------------------
    Version 3 : Mixed Method
    $Nb\_Ite \leftarrow 0$
    while (($r <= 0$) and ($Nb\_Ite \leq Nb\_Max\_Ite$)) do
      Generation_Attractive_CoverSet_Heuristic($\pi$,$u$,$r$)
      $Nb\_Ite \leftarrow Nb\_Ite + 1$
    end while
    if ($r <= 0$) then
      Generation_Attractive_CoverSet_Exact($\pi$,$u$,$r$)
    end if
    ------------------------------------------------
    if ($r <= 0$) then
      {the method did not provide an attractive cover set}
      $Stop \leftarrow 1$
    else
      {An attractive cover set is added}
      $U \leftarrow U \cup \{u\}$
      Restricted_Master_Problem_Resolution($U$)
    end if
  end while
```

GLPK (GNU linear Programming Kit) [13] available in the public domain.

In this section we evaluate the performance of our algorithms by way of simulations. We simulate a network with sensor nodes and target points randomly located in a $500m \times 500m$ area. We assume the sensing range is equal for all the sensors in the network and is set to $150m$. In the different scenarios we vary the number of randomly deployed sensor nodes n between 50 and 200 with an increment of 50. The number m of targets to be covered varies between 30 and 120 with an increment of 30. Each sensor has a lifetime of 1. The following requirements are satisfied: each sensor covers at least one target and each target is covered by at least one deployed sensor, the connectivity of the network is ensured and all sensors are capable of communicating with the base station. We measure the network lifetime and the execution times. For each scenario, results are averages over 10 instances (we generate 10 random topologies per scenario). In the algorithms we set Nb_Max_Ite to 10. The set of elementary cover sets is initialized with ($E = 10$) configurations.

B. Comparison of the execution times

First we compare and comment the CPU execution times of the different resolution methods. Table II gives the optimal network lifetime and the distribution of the execution times (in seconds) for the three methods over the 16 scenarios. Results of table II are consistent with those obtained in the literature : network lifetime and execution times increase with sensor density, network lifetime decreases with targets number for a fixed number of sensors because they are more requested. From the above results, we see that the Mixed Method can be up to 6 times faster than the Exact Method which solves an integer programming at each iteration. And the computing times of the Heuristic Method is really lower than the others each time that the number of targets exceeds 60. We observe that the Mixed Method uses 1.83 times on average the algorithm 3 for the resolution of the AUX-problem at optimality, which is really little but enough to slow its execution time.

N	M	$Lifetime$	$Exact$	$Heuristic$	$Mixed$
50	30	3.8	0.25	0.30	0.12
	60	3.0	1.03	0.53	0.52
	90	2.8	2.95	0.82	1.55
	120	2.7	8.40	1.20	4.03
100	30	8.7	3.29	2.97	1.03
	60	7.2	26.53	4.25	8.41
	90	6.9	243.95	6.82	74.19
	120	6.7	749.46	9.70	220.64
150	30	14.7	17.17	14.51	4.94
	60	12.3	315.66	22.21	48.96
	90	11.8	2365.65	30.61	525.21
	120	11.3	9249.81	48.15	1987.04
200	30	19.6	38.80	34.85	9.50
	60	17.3	750.40	56.34	126.39
	90	16.6	8229.53	132.46	1297.82
	120	15.5	28942.49	105.87	4393.04

Table II
EXECUTION TIMES (IN SECONDS) BETWEEN THE 3 METHODS

C. Comparison of the objective value

We compare the optimal solution value obtained with the Exact Method with approximate solution values obtained with the Heuristic Method. And we conclude that the Heuristic Method is a very efficient method because this method is able of finding the same solution value as the value of the optimal solution in all simulated cases with computing times drastically lower. We have also tested an other heuristic which is not presented here due to space limitations and this second heuristic finds the same solution values as the Exact Method with two exceptions over the 160 tests and the difference is equal to 0.0885 in the first case and to 0.2482 in the second case. These results are very promising and should be confirmed on other problem instances with a larger number of sensors and targets.

D. Comparison of the number of generated cover sets

Table III gives the distribution [1] of the number of attractive cover sets which have been generated to build an adaptive scheduling. We see that the scheduling resulting from the Heuristic Method contains a higher number of cover sets. We may have expected this result because the Exact Method always generates the most attractive cover set at each iteration so that the maximal lifetime of the network is reached with a minimal number of cover sets. The Heuristic Method could be improved to generate less cover sets but more attractive. Nevertheless this method is very efficient as it quickly produces a good solution by generating a slightly higher number of cover sets. And it may be interesting for wireless sensor networks to deal with a large number of cover sets such that sensor nodes frequently oscillate between an active and an inactive state as recommended in [5].

	$Exact$	$Heuristic$	$Mixed$
MIN	1.00	1	1.00
FST	10.00	15	16.00
MED	24.50	38	34.00
THD	58.25	74,5	69.50
MAX	130.00	131	124.00

Table III
NUMBER OF GENERATED ATTRACTIVE COVER SETS

VI. CONCLUSION AND FUTURE WORK

Energy-efficiency is crucial in power-limited wireless sensor network, since nodes have significant power constraints (battery life). In this paper we have investigated the problem of prolonging the network lifetime by organizing sensors into non-disjoint cover sets which operate sucessively in order to monitor all targets. We have formulated this problem as a linear programming where variables are the activation times of the cover sets and we have proposed a column generation approach to solve it. Instead of solving the auxiliary problem to optimality to generate an attractive cover set, we design a efficient heuristic. Simulation results show the performance of the heuristic which obtains very good solutions with very low time complexity. Although the method is a centralised one, it may be used to measure the quality of distributed solutions and it can be easily extended to deal with different QoS requirements.

REFERENCES

[1] I. Akyildiz, W. Su, Y.Sankarasubramniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Comm. Magazine*, pp. 102–114, 2002.

[2] T. Padmavathy and M. Chitra, “Extending the network lifetime of wireless sensor networks using residual energy extractionhybrid scheduling algorithm,” *Int. J. of Communications, Network and System Sciences*, vol. 3, no. 1, pp. 98–106, 2010.

[3] M. Cardei, M. T. Thai, Y. Li, and W. Wu, “Energy-efficient target coverage in wireless sensor networks,” in *in IEEE Infocom*, 2005, pp. 1976–1984.

[4] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, “Power efficient monitoring management in sensor networks,” in *Wireless Communications and Networking Conference, WCNC. 2004*, 2004.

[5] S. Slijepcevic and M. Potkonjak, “Power efficient organization of wireless sensor networks,” in *IEEE International conference on Communications*, 2001, pp. 472–476.

[6] D. Zorbas, D. Glynos, P. Kotzanikolaou, and C. Douligeris, “Solving coverage problems in wireless sensor networks using cover sets,” *Ad Hoc Netw.*, vol. 8, pp. 400–415, June 2010.

[7] A. Alfieri, A. Bianco, P. Brandimarte, and C.-F. Chiasserini, “Maximizing system lifetime in wireless sensor networks,” *European Journal of Operational Research*, vol. 181, no. 1, pp. 390–402, 2007.

[8] Y. Gu, Y. Ji, J. Li, and B. Zhao, “Qos-aware target coverage in wireless sensor networks,” *Wirel. Commun. Mob. Comput.*, vol. 9, pp. 1645–1659, December 2009.

[9] G. Dantzig and P. Wolfe, “Decomposition principle for linear programs,” *Operations Research*, pp. 101–111, 1960.

[10] A. Caprara, M. Fischetti, and P. Toth, “Algorithms for the set covering problem,” *Annals of Operations Research*, vol. 98, p. 2000, 1998.

[11] Y. Ye, “An o(n^3l)) potential reduction algorithm for linear programming,” *Mathematical Programming*, vol. 50, pp. 239–258, 1991.

[12] M. Garey and D. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.

[13] A. Mahkorin, *GNU Linear Programming Kit, Reference Manual*, 2010.

[1] MIN stands for MINimum, FST for FirST quantile (25% of the population), MED for MEDian (50 % of the population), THD for THirD quantile (75% of the population) and MAX for MAXimum

PCB Integration of Dye-sensitised Solar Cells for Low-cost Networked Embedded Systems

Jens Eliasson and Jerker Delsing
Dept. of Computer science, space and electrical engineering
Luleå University of Technology
Luleå, Sweden
Email: jens.eliasson@ltu.se

Simon J. Thompson and Yi-Bing Cheng
Dept. of Materials Engineering
Monash University
Melbourne, Australia
Email: simon.thompson@eng.monash.edu

***Abstract*—Wireless sensor networks are envisioned to make a large impact on how sensor data from physical phenomena can be utilized by millions of users on the Internet. However, one concern in deploying a large number of real-world physical sensors is that replacing spent batteries might not be feasible. One solution to this issue may involve energy harvesting technology, e.g. solar panels. Solar panels are currently relatively expensive because they require a time-consuming and therefore costly assembly process. As an alternative, this paper suggests a new approach to powering networked sensors: the direct integration of a solar cell onto the sensor nodes printed circuit board. This approach eliminates the need for manual assembly and the use of expensive connectors. This article presents test results and a feasibility analysis of the direct integration of a dye-sensitised solar cell onto a circuit board. Preliminary results indicate that this approach is feasible for networked sensors. The aim of this work is to develop a method for the assembly of complete systems, consisting of a printed circuit board, components, and power supply, using a single production process. The first steps towards this aim have been taken, and the authors believe that the proposed approach may be one enabling technology for future large-scale, low-cost wireless sensor networks.**

***Keywords*-Dye sensitised solar cells, energy harvesting, networked sensors, wireless sensor networks**

I. Introduction

A wireless sensor network (WSN) is composed of a large number of heterogeneous sensor nodes, or *sources*, that sense phenomena in the physical world [1]. A sensor network also includes one or several gateways, or *sinks*, which forward sensor data from nodes in the internal network to an external network [2]. Research on WSN technology originally focused on military applications, such as battlefield surveillance, land mine detection, and soldier monitoring [3]. Current wireless sensor network research is additionally motivated by an increasing number of civil usage scenarios, such as environmental and habitat monitoring, seismic and volcanic monitoring, structural monitoring, and industrial applications [4], [5].

Wireless sensors are expected to have a drastic impact on how measurements of the physical world will be presented to users on the Internet. A vision, in which Internet-connected wireless sensors are deployed in the vicinity of users, named *the Internet of Things* [6] is also projected to enhance both safety and quality of life for future generations. For this vision to be realized, a number of issues must be resolved. Three of these issues, addressed by this article, are:

- Reducing power consumption
- Enabling wireless power
- Lowering the cost of the sensor nodes

Reducing power consumption can be achieved using a number of methods, such as using more efficient components, integrating more intelligent routing protocols [7], or developing energy-aware computing. Wireless power requires power harvesting, power storage, and an appropriate power usage architecture at the sensor node; see for example [8], [9], [10]. A node's cost will be reduced with the use of more integrated components, and the price of printed circuit boards (PCB), integrated circuits (ICs), and other components will drastically decrease with increased production volumes. However, the costs of certain node components, such as batteries and power supplies, do not scale as effectively as circuit board production volumes. The cost of packaging a complete node with a circuit board, batteries, solar panels, and enclosure will not be reduced by the same order of magnitude as that of the electronics. This is a major obstacle for realizing the vision of massive wireless sensor networks.

One consideration for energy harvesting relates to the energy density from different sources. It is clear that solar cells are superior to other energy harvesting approaches such as vibrations and thermoelectric power, as reported by Yang et. al [8]. Therefore, the focus of our work was to investigate direct, low-cost solar cell technology integration with a sensor node.

When comparing different solar cell technologies, both power efficiency and cost must be considered. Two main candidate technologies: silicon based solar cells and dye sensitised solar cells (DSC), sometimes called Grätzel cells [11], have been selected for further investigation. A comparison between silicon based solar cells and DSC can be found in [12]. Regarding energy capability a traditional silicon-based solar cell offers about $43mA/cm^2$ at 0.7V, whereas current DSCs offer about $22mA/cm^2$ at about 0.6V [13]. Regarding cost, DSCs are considered superior because material cost and manufacturing cost is clearly lower than for silicon-based cells.

This paper presents a novel approach aimed at further reducing manufacturing and integration cost for DSCs that power wireless sensor nodes. The approach is to manufacture a solar panel directly on a sensor node's circuit board, thus reducing the cost of manufacturing the cell separately and eliminating the assembly cost. This has several benefits, as the resulting device consists of an integrated solution that effectively eliminates costly silicon-based cells, cables and connectors, and an additional integration step. The proposed approach also increases the system's robustness because there are no connectors or cables that can disconnect due to mechanical phenomena, e.g., vibrations or impacts. The goal of this research is to develop a holistic method for producing complete low-power systems, where assembly of the PCB, components, and an energy-harvesting device can be completed with a single process. The first steps have been taken - a solar cell has been integrated with a PCB - and the authors believe that, in the future, a solar cell will be directly printable on a PCB using a sequential build-up (SBU) technique. For example, Blackshear et al. reported in 2005 [14] the advantages of using SBU for chip assembly onto circuit boards.

The paper is outlined as follows: this section has presented related work and a background of wireless sensor networks and solar cell technologies. The next section gives an overview of DSCs. Section III presents the new method of integrating a DSC directly onto a circuit board, and Sections IV and V show the experimental setup, and results from real-world tests, respectively. Finally, suggestions for future work are presented in Section VI, followed by conclusions in Section VII.

II. Dye sensitised solar cells

The dye sensitised solar cell (DSC) is currently being investigated as a low cost method of harvesting the abundant energy of sunlight into electricity [11]. It offers the advantages of low cost and better light harvesting in low and/or diffuse lighting, which are more realistic conditions than would be optimal for other photovoltaic devices, such as silicon-based cells.

The DSC operates by light exciting an electron in a dye molecule adsorbed onto a mesoporous semiconductor to an energy level above the conduction band of the semiconductor. The electron is quickly transferred to the conduction band of the semiconductor and transported through the network of interconnected nanoparticles to the electrode. The electron passes through the external circuit and then reduces an electrolyte at the counter electrode which in turn reduces the dye, returning it to its ground state. This type of solar cell exhibits an efficiency of over 11 %, as shown by Han et. al [15]. The operation of the DSC allows for cheap, abundant materials to be used for device components, combined with less energy-intensive processes used for manufacturing. This offers the potential for significantly lower production costs compared to more traditional silicon solar cells, in turn reducing the energy and cost payback times significantly. These factors make the DSC an attractive renewable energy source for the future.

The drawbacks for DSCs are related to lower performance compared to silicon devices, a corrosive electrolyte that limits material selection options and shorter device lifetimes, primarily due to the volatile electrolyte used in the most efficient designs. It is difficult to construct devices with long lifetimes when encapsulating a volatile, corrosive solvent. To this end alternative electrolytes have been investigated - generally highly viscous, non-volatile ionic liquids. Solid state hole conductors have also been considered and are a more elegant solution, as they also remove corrosive iodine from the system, expanding materials selection options within the cell as well as eliminating any solvent leakage issues. The leading organic hole conductor is 2,2,7,7-tetrakis(N,N-di-p-methoxypheny-amine)-9,9-spirobifluorene (spiro-MeOTAD) [16], with reported device efficiencies up to 4.4% [17]. A solid state device is typically constructed onto fluorine doped tin oxide (FTO) glass with a titania (TiO_2) layer coated on top, which is dyed and then infiltrated with the hole conductor. The counter electrode is a gold layer evaporated onto the coated titania layer and connected to an electrically isolated section of the FTO glass. This architecture is ideal for integration with circuit boards, shown in Fig. 1. The circuit board may be physically contacted to the gold contacts on the back of the DSC module, as shown in Fig. 3. The connections will be made such that each cell is independently measurable and bi-passable if necessary.

III. PCB WITH INTEGRATED DYE SENSITISED SOLAR CELL

DSC modules were created using the screen printing technique, on pre-etched 100 mm × 100 mm 13 Ω/square FTO coated conducting glass (Nippon) masterplates. The etching to separate the contacts for the individual cells was performed using a laser engraving system, a Versa laser VL3.50 unit, which produced fine lines (∼150μm) with high spacial precision. Following this procedure the glass was cleaned and a dense blocking layer of TiO_2 was deposited by spray pyrolysis, with the areas for electrical contacts by solder and the gold layer being masked by flattened aluminium rods.

The screen printing paste for the active layer, provided by JGC Catalysts and Chemicals Ltd, contained 18 nm particles of anatase titania and was diluted by terpineol at a ratio of 2:1 paste (Fluka). The thickness of the titania layer was determined by a Veeco Detak 150 stylus profilometer, to be ∼2 μm. The titania layer was incrementally heated to 450°C for 30 min and then to 500°C for 15 min. The master plates were cut into 50 mm × 50 mm modules and reheated to 500°C for 30 mins before being placed into the dye solution of 30 mM Z907 (Dyesol) in an acetonitrile/tert-butanol 1:1 mixture. for approximately 24 hours. The electrolyte was a solid state hole conductor, namely Spiro-MeOTAD, which was deposited by spin-coating using a solution that consisted of 180mg/mL of Spiro-MeOTAD (Merck) in chlorobenzene (Sigma) with additives of 4-tertbutylpyridine (TBP) (Sigma) (17.6ul/mL) and Li-TFSI (Sigma) (19.5mM). Chlorobenzene was used on a cotton bud to remove excess Spiro-MeOTAD from the glass were series interconnects were to be formed. The gold charge collecting layer was deposited onto the module via thermal evaporation, and the areas not to be coated with gold were masked with Kapton tape^{TM} (3M).

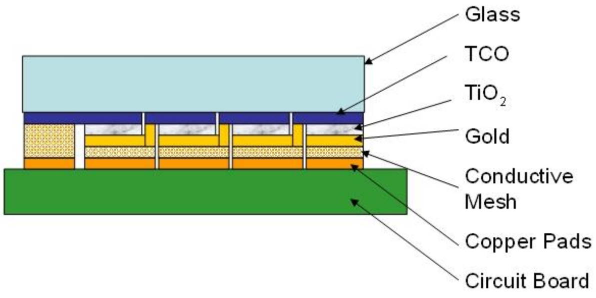


Fig. 1. Layout of the PCB with integrated dye-sensitised solar cell

An attempt was made to integrate these devices onto a PCB using conductive epoxy however, this had a detrimental effect on the DSC leading to dye desorption. Therefore, this approach was abandoned in favour of using a soft compressible conductor. The material used was a polymer mesh substrate with copper deposited onto it. The copper mesh was cut into pieces of the same width as the pads, but slightly longer such that they could be laid over the pads and adhered using Kapton tape. The module was placed on top of the PCB such that the gold contacted the copper mesh and no shorting occurred between cells. The PCB and DSC module where

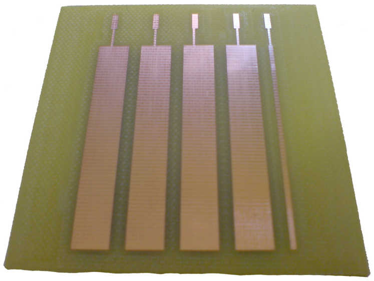


Fig. 2. Prototype board layout

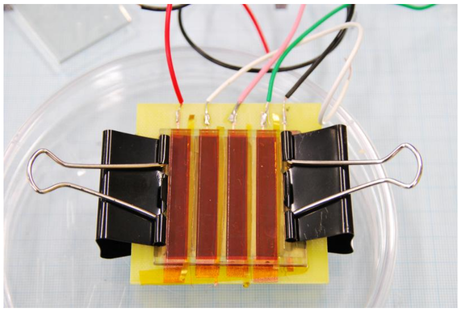


Fig. 3. PCB DSC solar panel prototype board, ready for integration with sensor node

then clipped into place using bulldog clips. During these alignment and clipping processes care was taken not to damage the fragile gold layer. Wires were soldered onto the board such that the entire module could be used or individual cells could be measured and/or bypassed if faulty. Fig. 2 shows the PCB that serves as the base for the new solar cell. The board, which is composed of four copper stripes each 49 mm wide and 6 mm long, was manufactured using a milling machine from an Eagle CAD design.

Fig. 3 shows a board produced with a DSC on a PCB. This board was used for initial tests and for 5-month degradation tests.

IV. Experimental setup

Several experiments were performed to investigate the performance of the PCB-based cell. To evaluate the performance of the module under standard conditions a solar simulator was used. The modules were tested under 1 Sun illumination, 100 mWcm^{-2} AM1.5G, using a 1000 W solar simulator xenon lamp (Oriel) fitted with an appropriate filter to achieve spectral match and a Keithley 2400 source meter. Illumination intensity was varied by the use of fine wire mesh and calibrated using a silicon diode. The active area was 10.5 cm^2, while the size of the glass was 25 cm^2, this shows a poor active area to device area ratio. In future work this will be increased with 80% coverage, which is a challenging, but achievable, target for an interconnected module of this size. No masking was used; efficiencies may therefore be over estimated due to light piping within the glass.

To investigate the real world performance and feasibility for practical use, tests were performed both indoors and outdoors using different light sources.

A. Measurement system

A measurement system was created to capture characterization measurements for the PCB solar cell. The measurement system, shown in Fig. 4, consists of a 24-bit analog-to-digital converter (ADC) that measures the voltage drop over a 5 ohm resistor, which is used to measure current. To obtain an I-V curve, a digitally programmable potentiometer was also used so that different loads could be presented to the cell. A Mulle v3.1 networked sensor node equipped with a Bluetooth 2.0 transceiver was connected to the measurement system. Using this approach, the PCB cell can be tested outdoors by having a wireless connection to a laptop or PC, which can be placed indoors. The measurement system will be used also to measure the temperature dependency of the cell during winter tests. In addition, the measurement system also serves as a building block in the power supply unit (PSU) that may be used together with the PCB-cell. The PSU includes a boost converter that generates a 5.0V output used to charge a super capacitor. A switch is used to select whether the Mulle should be powered by the super capacitor or by a battery. The Mulle v3.1 also features a battery monitor chip, capable of measuring battery voltage, power consumption, available energy, and estimated lifetime. Combined with the Mulle's on-board features, the PSU can enable true energy- and power-aware operation.

Fig. 5 shows the measurement system. The system can measure voltages up to 6.5V, and current with a resolution around 20μA. The load can be programmed to any value between 100Ω and 100kΩ in 256 steps.

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

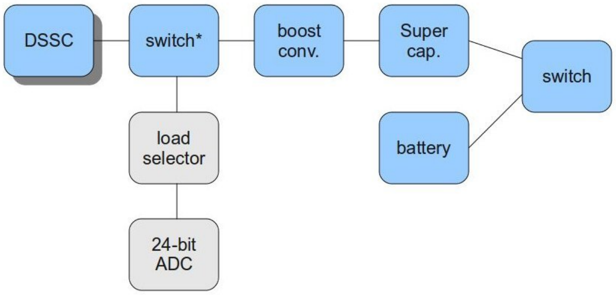


Fig. 4. Measurement system overview

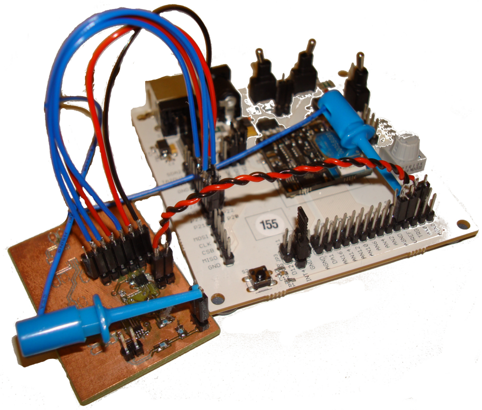


Fig. 5. Measurement system implementation

The measurement system is completely wireless, which allows remote monitoring of the PCB cell. A dedicated software written in C was used to retrieve data from the Mulle and store the results to file. In future implementations, the measurement system should also be integrated with the solar panel to enable true energy- and power-aware sensor node operation. By measuring available stored energy, power harvesting output and power usage, the software can be used to make intelligent decisions regarding how energy-consuming tasks should be managed.

B. Performed measurements

The following experiments were performed in order to test the cell's performance under in a real-world setting. The different tests that the cell was tested in are typical application locations where a networked embedded system can be deployed.

1) Measurement of the PCB DSC module's performance initially and after 5 months
2) Measurement of the PCB DSC's current response at various light incident angles
3) Measurement of the effect of varying light intensity on the current output of the PCB DSC module.
4) Tests of power generation at indoor and outdoor locations and different lighting conditions

The cell was tested for long term degradation effects and different light sources at different angles. However, no temperature tests were performed. It is considered as future work to investigate the cell's performance in low temperature environments.

TABLE I
MULLE V3.1 POWER CONSUMPTION

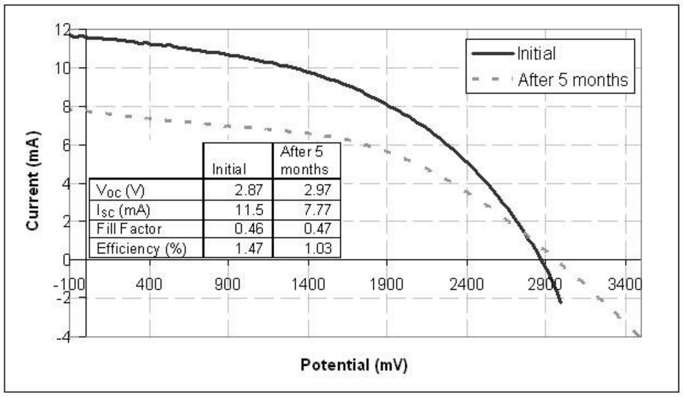
Mode	Delay	Current
All systems sleep	-	0.004 mA
MCU 10.0 MHz, BT off	-	7.6 ma
MCU 5.0 MHz, BT off	-	5.1 mA
MCU 2.5 MHz, BT off	-	3.1 mA
MCU 1.25 MHz, BT off	-	2.2 mA
MCU sleep, BT listen	2-12 s.	1.0 mA
MCU sleep, BT active	-	40.3 mA
MCU sleep, BT sniff (210 slots)	131 ms.	8.4 mA
MCU sleep, BT sniff (2010 slots)	1256 ms.	2.8 mA
MCU sleep, BT parked (18 slots)	13 ms.	7.5 mA
MCU sleep, BT parked (200 slots)	130 ms.	2.7 mA
MCU sleep, BT parked (4094 slots)	2560 ms.	1.8 ma

C. Real-world energy usage

The feasibility of using the prototype solar cell, with the power characteristics presented in the previous section, for a real-world networked sensor is presented here. The Mulle node [18] has been used in a number of WSN and BSN applications [19], which will be used as an example for calculating operational lifetimes when combined with the PCB cell. Table I shows examples of the current consumption of a Mulle v3.1 in different operating modes.

V. RESULTS

The initial performance of the PCB DSC module was 1.47% efficient initially, degrading to approximately 1% after 5 months, as shown in Fig. 6. The aging of the module was performed with no encapsulation and at ambient conditions. The drop in performance is due to reduced current most likely caused by the degradation of the dye molecules by oxygen and water. This suggests good stability and may be significantly improved with encapsulation of the device. Figure 7 shows the performance of the PCB DSC at varying incident light angles. Here 0° corresponds to the light beam being perpendicular to the surface of the module. The module demonstrates good performance with the PCB DSC maintaining 80% of current at a 45° tilt. Figure 8 shows the variation of the output current with varied input light intensity, which remains linear for lower light intensities, but slightly decreases upon approaching full illumination, showing the cell is approaching it's photocurrent limit. This data may help determine the illumination intensity from the photocurrent produced by the module although this will exhibit a significant spectral mismatch for artificial light sources.



	Initial	After 5 months
V_{OC} (V)	2.87	2.97
I_{SC} (mA)	11.5	7.77
Fill Factor	0.46	0.47
Efficiency (%)	1.47	1.03

Fig. 6. PCB DSC current-voltage performance, initially and after 5 months

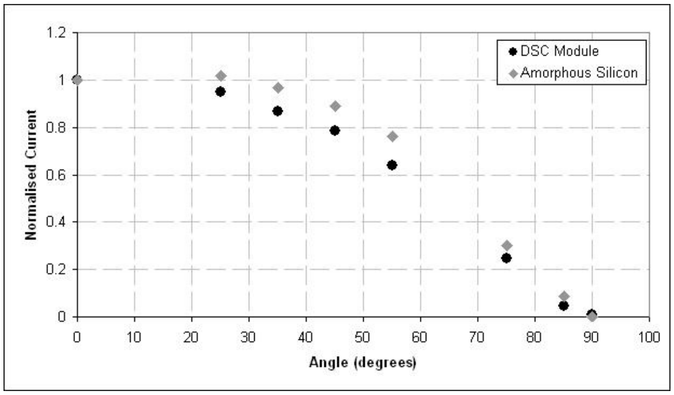


Fig. 7. PCB DSC short circuit current response for different light incident angles

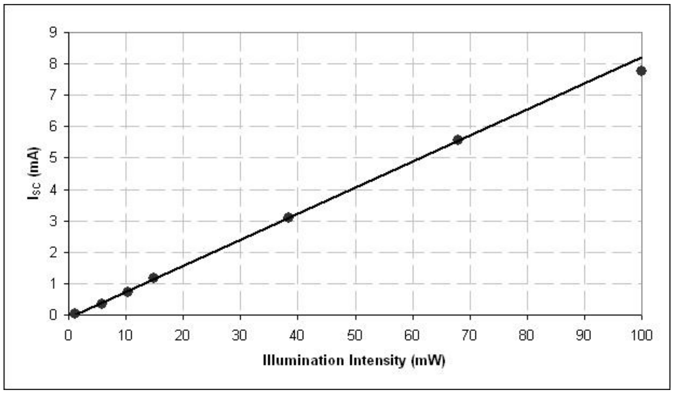


Fig. 8. Short circuit current response for the PCB DSC with light intensity varying between 1 and 100%

To evaluate the module's output in real world scenarios the short circuit current was measured at a number of locations that reflect typical applications for the sensor node which can be placed either outdoors or indoors. The following locations were tested:

i) Office with ceiling fluorescence lightning and ambient light from shaded windows; cell horizontal
ii) Corridor with ceiling fluorescence lightning and no ambient light from windows, cell horizontal
iii) Workshop well lit with with ceiling fluorescence lightning and some ambient light from shaded windows; cell horizontal
iv) Office desk with 23W desk fluorescent lamp; cell horizontal
v) Near a closed window with no direct sunlight; cell horizontal
vi) Near an open window with no direct sunlight; cell horizontal
vii) Near a closed window with some direct sunlight; cell horizontal
viii) Near a closed window with direct sunlight; cell horizontal
ix) Near a closed window with no direct sunlight; cell tilted for maximum illumination
x) Outside in full sun light; cell horizontal
xi) Outside in full sun light; cell tilted for maximum illumination

The resulting data is in Table II. For a number of practical usage scenarios assuming no real-time radio communication, a small dye solar cell should be sufficient to provide the necessary power for making wireless power a reality.

TABLE II
CURRENTS FROM PCB DYE SOLAR CELL IN TYPICAL USAGE SCENARIO LOCATIONS.

Location	distance to source [m]	Current $[\mu A]$
i	2	6
i	0.3	60
i	0.1	220
ii	1	6
ii	0.1	90
iii	2	50
iv	0.2	240
iv	0.01	3000
v	-	220
vi	-	330
vii	-	800
vii	-	2650
ix	-	3700
x	-	6800
xi	-	8000

When comparing the power output from the PCB DSC cell with Table I, it is clear that the generated power is sufficient for powering a Mulle sensor node as long as low-power modes are utilized. Since the peak power of a Mulle is higher than the maximum power output of the PCB DSC cell, some storage will always be required. A super capacitor, a rechargeable battery, or a combination of both can be used for energy storage. Performed tests indicates that the presented approach is feasible for powering low-power electronics such as sensor nodes.

VI. FUTURE WORK

The first steps towards an integrated manufacturing process for solar-powered embedded systems have been successfully completed. In the next step, the authors will continue to investigate printing a dye sensitised solar cell directly onto a printed circuit board using mass production techniques. The ultimate aim is to develop a method for assembling and manufacturing a complete system that includes a PCB, components, and a solar cell, using a single process.

Another performance-enhancing approach worth investigating requires interconnecting a number of cells in a matrix-style framework using MOSFET transistors. This is possible because all connections can be made by vias, instead of wires. This would enable the system to identify cells with bad performance, and to allow them to be bypassed. The system would thereby dynamically reconfigure itself for maximum performance depending on: light irradiation, work load, and properties of the cells. This approach requires software-support for full performance advantages.

Another issue that needs further investigation is how the system should be encapsulated in a transparent package. One method is the embed the entire system in optically transparent glue, as shown in [20]. How low temperatures are affecting the cell's performance must also be investigated. Finally, the use of a more low powered device, such as the Mulle v5.2 with an IEEE 802.15.4 radio, should be used to test the true performance in a wireless sensor network.

VII. CONCLUSION

This paper has presented a novel approach for powering low-power electronic devices, such as networked embedded systems and sensor nodes. The approach integrates a dye sensitised solar cell directly onto a device's circuit board thereby reducing the material and assembly costs. A prototype device has been manufactured to demonstrate the feasibility of this approach and to enable the cells' real-world performance to be evaluated. Test results, both initial and after five months of degradation, have been presented to support the claims.

By integrating the power supply directly onto a circuit board, the authors envision that networked sensors may be manufactured at a greatly reduced cost in the future. When combined with new technologies for energy storage and transparent encapsulation, the presented approach can be an enabling technology for future low-cost, large-scale wireless sensor networks, in support of the vision of *the Internet of Things*.

ACKNOWLEDGMENTS

The authors would like to thank Mikael Larsmark for help with manufacturing the circuit boards and JGC Catalysts and Chemicals Ltd, Kitakyushu-Shi, Japan, for their generous donation of the titania pastes.

REFERENCES

[1] D. Culler, D. Estrin, and M. Srivastava, “Overview of sensor networks,” *Computer*, vol. 37, no. 8, pp. 41–49, Aug 2004.

[2] Özgür B. Akan and I. F. Akyildiz, “Event-to-sink reliable transport in wireless sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 13, no. 5, pp. 1003–1016, 2005.

[3] T. Bokareva, W. Hu, S. Kanhere, B. Ristic, N. Gordon, T. Bessell, M. Rutten, and S. Jha, “Wireless sensor networks for battlefield surveillance,” 2006. [Online]. Available: http://www.cse.unsw.edu.au/~tbokareva/papers/lwc.html

[4] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, “A wireless sensor network for structural monitoring,” in *SenSys ’04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004, pp. 13–24.

[5] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, “Wireless sensor networks for habitat monitoring,” in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM Press, 2002, pp. 88–97.

[6] “IPSO Alliance,” 2010, http://www.ipso-alliance.org/.

[7] K. Zeng, K. Ren, W. Lou, and P. J. Moran, “Energy aware efficient geographic routing in lossy wireless sensor networks with environmental energy supply,” *Wireless Networks*, vol. 15, pp. 39–51, Jan 2009.

[8] R. Moghe, Y. Yang, F. Lambert, and D. Divan, “A scoping study of electric and magnetic field energy harvesting for wireless sensor networks in power system applications,” in *Energy Conversion Congress and Exposition, 2009. ECCE 2009. IEEE*, sept. 2009, pp. 3550 –3557.

[9] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, “Design considerations for solar energy harvesting wireless embedded systems,” *IPSN ’05: Proceedings of the 4th international symposium on Information processing in sensor networks*, vol. 64, p. 64, 2005, cited By (since 1996): 9. [Online]. Available: www.scopus.com

[10] J. Eliasson, P. Lindgren, J. Delsing, S. J. Thompson, and Y.-B. Cheng, “A power management architecture for sensor nodes.” Kowloon: IEEE, feb 2007, pp. 3008 – 3013.

[11] B. O’Regan and M. Grätzel, “A low-cost, high-efficiency solar cell based on dye-sensitized colloidal tio2 films,” *Nature*, vol. 353, no. 6346, pp. 737–740, 1991.

[12] B. A. Gregg and M. C. Hanna, “Comparing organic to inorganic photovoltaic cells: Theory, experiment, and simulation,” *Journal of Applied Physics*, vol. 93, no. 6, pp. 3605–3614, 2003. [Online]. Available: http://link.aip.org/link/?JAP/93/3605/1

[13] M. A. Green, K. Emery, Y. Hishikawa, and W. Warta, “Short Communication Solar cell efficiency tables,” *Progress in Photovoltaics: Research and Applications*, vol. 17, 2009.

[14] E. Blackshear, M. Cases, E. Klink, and S. Engle, “The evolution of build-up package technology and its design challenges,” *IBM journal of research and development*, vol. 49, pp. 641–661, 2005.

[15] L. Han, A. Fukui, N. Fuke, N. Koide, and R. Yamanaka, “High efficiency of dye-sensitized solar cell and module,” in *IEEE 4th World Conference on Photovoltaic Energy Conversion*. IEEE, 2006, pp. 179–182.

[16] U. Bach, D. Lupo, P. Comte, J. E. Moser, F. Weissortel, J. Salbeck, H. Spreitzer, and M. Grätzel, “Solid-state dye-sensitized mesoporous tio2 solar cells with high photon-to-electron conversion efficiencies,” *Nature*, vol. 395, pp. 583–558, 2009.

[17] S.-J. Moon, J.-H. Yum, R. Humphry-Baker, K. M. Karlsson, D. P. Hagberg, T. Marinado, A. Hagfeldt, L. Sun, M. Grätzel, and M. K. Nazeeruddin, “Highly efficient organic sensitizers for solid-state dye-sensitized solar cells,” *Journal of Physical Chemistry C*, pp. 16 816–16 820, 2009.

[18] “Eistec AB,” 2010, http://www.eistec.se/.

[19] J. Eliasson, P. Lindgren, and J. Delsing, “A Bluetooth-based Sensor Node for Low-Power Ad Hoc Networks,” *Journal of Computers (JCP)*, pp. 1–10, May 2008.

[20] J. Eliasson and W. Birk, “Towards road surface monitoring : experiments and technical challenges,” in *IEEE International Conference on Control Applications : CCA ’09*, St. Petersburg, Russia, July 2009, pp. 655 – 659.

Improving the Directed Diffusion in Order to Reduce the Average of Energy Consumption in Wireless Sensor Networks

Miresmaeil Mirnabibaboli
Yerevan Telecommunication
Research Institute
Armenia,Yeravan
m.mirnabi@gmail.com

Mehdi Mirfattahi
Computer Engineering Raja
University of Iran
Iran,Qazvin
m.mirfattahi@gmail.com

Mher Markosyan
Yerevan Telecommunication
Research Institute
Armenia,Yeravan
mark@yetri.am

Abstract—**Network lifetime is an important issue in wireless sensor networks. One of the main problems in directed diffusion is the implementation of flooding diffusion used to forward interest and discover the routing map that reduces network lifetime through high energy consumption. The main purpose of this study is to limit flooding for the purpose of increasing network lifetime. In this paper, interests are classified based on their content; and the network is divided geographically. A counter is assigned to each Geographical location that will determine the use of Flooding for a potential existence of a source and to use Rumor for else where. Using the above mentioned recommendations, energy consumption would decrease due to the reduction in Flooding; and consequently, network lifetime would increase.**

Keywords-Wireless Sensor Network; Node; Sink; Directed Diffusion; Rumor; Flooding.

I. INTRODUCTION

Over the past years, there was a substantial progress in Wireless Sensor Networks. Wireless Sensor Network is a combination of large quantities of small nodes dealing directly with the physical environment. These sensors respond to the environment and gather its information. It also sends the data between the information nodes via wireless transmission. Each node works independently and without the need of human intervention. Its size is too small and has limitations in processing, memory capacity, and power supply. These limitations result in some problems which are the source of much research in this field. The most prominent concern in this network would be the information gathered by network sensors. Meanwhile, the network nodes can no longer be used after its energy consumption. Therefore, optimization of energy consumption in order to increase the network lifetime is one of the challenging discussions in this type of networks.

Recent breakthrough in radio short wave technology and micro-electro-mechanical systems led to the invention of intelligent sensors that are capable to sense and process data in wireless communications. These small sensors can be used to gather the ideal information and deliver the sensed data to central units called Sinks. Sending the sensed data to Sinks might take several steps which is a typical trait for data diffusion in wireless sensor networks.

In recent years [4], [5], [6], many algorithms and protocols have been suggested to increase efficiency and reliability for data diffusion in wireless sensor networks. For instance, flooding diffusion is the most reliable method for transmitting data from sensors to the sinks. The implementation is quiet easy as flooding protocols do not use any recondite algorithms. However, the main problem in this protocol is the amount of overhead due to the repetitive messages sent. This problem causes inefficiency in energy consumption. To solve this problem, a new data centric method has been suggested which is quiet different from the traditional routing mechanisms used in wireless sensor network. Data centric protocols are query-based and are dependent on proper data naming (including attribute and value).

Therefore, redundant transmission of many packets will be blocked; and consequently, it will increase the energy consumption. Directed diffusion is one of the data centric protocols which might be the most common data centric protocol due to its energy saving characteristics. For example, using interest message for query as well as data gathering, and using gradient mechanisms also reinforces routing. The gradient is a direction toward the neighbor nodes through which sink is accessible. In most data packets of a set, the sensor is sent toward sink. Hence, each sensor node tacitly has the direction of sink for the purpose of transmission of the sensed data toward it. The important issue here is that the amount of gradient should be built and maintained within each node. Generally, managing the amount of gradient by primary and periodically flood diffusion of controlling packets is performed via the sink. Notice that this periodical flood diffusion will result in too much overhead in sensor networks. In addition, in case of a change in network topology due to being dynamic or the decay of wireless connections,

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

the value of some gradients will be invalid. Hence, the repetitive flood diffusion is required.

Throughout data discovery phase, directed diffusion encounters is that when discovered data is sent from source to the sink, each node sends this message to all neighbors. It happens because the node does not know which neighbor node has the ability of sending this data to the sink. Consequently, even if the source and sink are very close to each other, (Figure 1) many other irrelevant nodes will be involved in this connection.

The main reason for designing directed diffusion is increasing the efficiency of energy and thereafter, increasing the network lifetime.

This can be done by the help of two methods of data aggregation and in network processing to decrease the overhead of data transmission and consequently, lower energy consumption. However, due to the use of flooding diffusion, this has some limitations that cause too much overhead turnout in this algorithm. In this paper, by classifying the interests and dividing the network into separable areas, we can have a decline in energy consumption and an increase in network lifetime.

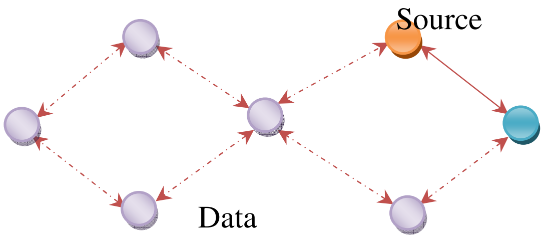


Figure 1. operation of diffusion in forwarding the discovery data

This paper includes different sections. In Section 2, we will study the works already done to improve the directed diffusion for the purpose of increasing the lifetime of network. Section 3 is regarding the suggestive method. In this part, there is a suggestion to improve the directed diffusion to increase the lifetime of network in regard to energy parameters. Afterwards, we will have the simulation in NS-2 and finally the result of simulation will be presented in the last section.

II. REVIEW OF RELATED WORK

There has been a great deal of research in wireless sensor network routing. One of the outstanding protocols is directed diffusion. In this protocol, a sink is responsible for making the return route from all sensed sources to sink by sending an interest packet. This return route is called “Gradient”. Using the interest packets and Gradients the route(s) between the source and the sink will be established. Directed diffusion uses reinforcement mechanisms to choose a high quality route among the several accessible routes to transmit the data. Based on the fact that in directed diffusion, each node sends a packet to a designated neighbor in next step through reinforced route, many of the tasks are done over the basic directed diffusion and each of them improves the overall turnover.

Basic Directed diffusion algorithm is also called diffusion algorithm with the attribute of a two-phase pull. In the first phase of this method, the destination node sends an interest packet to the network and whenever a node senses the collision of the gathered-data by its sensors, it sends a data discovery to the destination; while forwarding the interest message, gradients try to choose a gradient with the best responding time. And this process continues until it reaches the destination. After that, the source node forwards the gathered data to the destination. However being weak in a small number of situations, the two-phase pull is a suitable method for most applications.

Another improvement in directed diffusion is the hexagonal diameter based method. In this method, the sensor nodes similar to a beehive regularize a constant topology. And the main data will be transmitted over the diameter of the hexagonal. In fact, the hexagonal diameter is supposed to be the main route. In this method, the main routes change periodically to distribute the energy consumption among the sensors (Figure 2).

In order to decrease the energy consumption, ‘active’ status of sensors is used in the time of forwarding data and ‘inactive’ status is used in case the sensors are idle.

The rate of energy consumption and deferment in responding to an interest has a wonderful efficiency in comparison with basic forwarding.

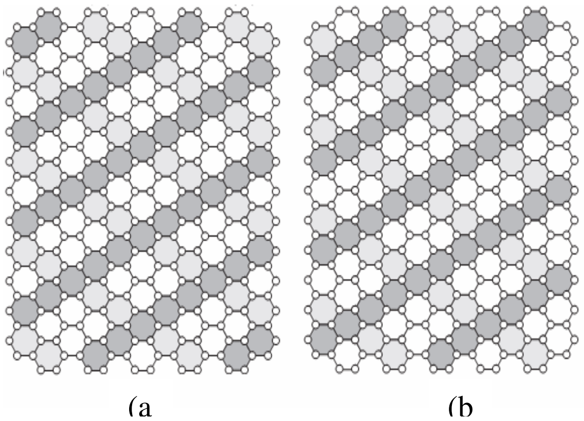


Figure 2. The periodic change to distribute the energy consumption among the sensors

Regarding the high cost of directed diffusion in flooding diffusion of data to energy source, a method has been devised to prevent the waste of energy in [9] entitled passive clustering. The main duty of clusters is to optimize the exchange of flood messages that prevents a high level of overhead. These clustering are forwarded with flooding of the controlling messages which occurs when there is no traffic within the network; as a result, it causes waste of energy for maintaining the structure of the cluster. Passive clustering, on the contrary

ISBN: 978-1-61208-144-1

to classic clustering algorithm, is dynamic and is created by the first flood message.

In this method, long frequency of cluster adjustment is prevented. To do this, controlling messages for maintaining cluster are not used explicitly. And all the related information, for maintaining clusters in data messages exchanged in network, is carried. In fact, passive clustering is an appropriate mechanism addressed to increase the efficiency of these crucial diffusion phases and consequently, the efficiency of the whole protocol. The protocols share some common properties:

1. They are on-demand mechanisms that maintain the operation only in case there is application traffic in need of their services.

2. They purely rely on local information for performing their functions. Figure 3 illustrates the structure of directed diffusion routing when passive clustering is used. The simulation results show that directed diffusion along with passive clustering in transmission rate actually increase the density of the network, and deferment has a better performance in comparison with basic forwarding.

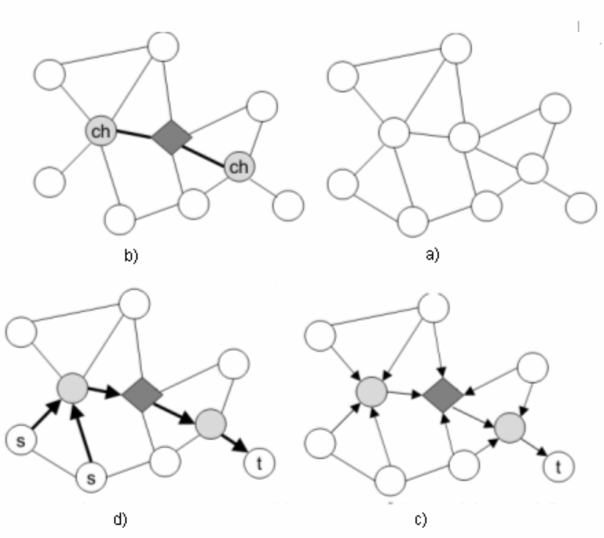


Figure 3. structure of clustered directed diffusion (Gray circles are head of groups and diamonds are the gates)

Based on the type of applications, sensor network have different needs. Directed diffusion is a data centric protocol whose performance in optimizing the energy consumption will be decreased significantly when a proper application is not used. In [12], a compatible directed diffusion routing protocol have been suggested based on their application. In this plan, a general message is used which has the potential of being applied to different practical scenarios just by a slight change. In this method, different versions of directed diffusion are defined in a general message. In fact, according to different applications, adaptive directed diffusion changes are made to result in the desired version. The results of simulation demonstrate that this idea in comparison with other directed diffusion in time of change has a good performance and application.

In [15], a method have been suggested which shows using two filters with the capability of forwarding increase energy. The first filter is a real-time filter which is used to decline the end-to-end deferment in real-time traffics. The second filter named as the best-effort delivery has been suggested in order to reach the global dominant energy as a result of network lifetime growth. Besides, a repairing mechanism for restoring node decay and connections is implemented for real-time traffic. The results of simulation illustrate that the suggested method has different service diffusion for real-time traffic and best-effort as well as low deferment for real-time traffic and totally provides more network lifetime in comparison with the basic diffusion.

In [16], in order to decrease the overhead of network resulted by the exchanged packets for periodical diffusion of interests, delivering the data discovery to sink will be declined. As a result, the energy consumption decreases due to the reduction in the number of exchanged controlling messages; therefore, the network lifetime increases. In the light directed diffusion; first of all, a spars logical topology will be produced through simple, local rules. Then, the directed diffusion will be implemented on this topology.

In the mentioned protocols, this section covers only energy efficiency using various mechanisms. Following that, in the next section, a method is suggested for increasing the network lifetime.

III. SUGGESTIVE METHOD ALGORITHM

A. *Clustering interests*

In order to overcome the limitations discussed in the previous section, in suggestive algorithm, we have clustered the interests based on demand; for example, clustering the temperatures over 35°; by clustering and naming the interests will be clustered; as a result, it will be quite easy to recognize the source.

B. *Network segmentation*

Network is segmented geographically; nodes store the vector and specifications of the neighbor nodes in a table and are informed about the neighbors. And their own sink node is defined as the vector of the source; and by the help of the neighbors' vectors, the network is divided into necessary segments. A variable to each segment is assigned and in case of finding the source in that area, the variable will be reported.

C. *Selection of Diffusion filter*

After clustering, segmentation and assigning a variable to each class and area, in the diffusion phase, the variables will be analyzed and filter selection will be done based on value.

Moreover, before transmission, first of all the sink will check its variables, when one variable is more than the other variables, sink uses flooding phase otherwise, it uses the rumor phase.

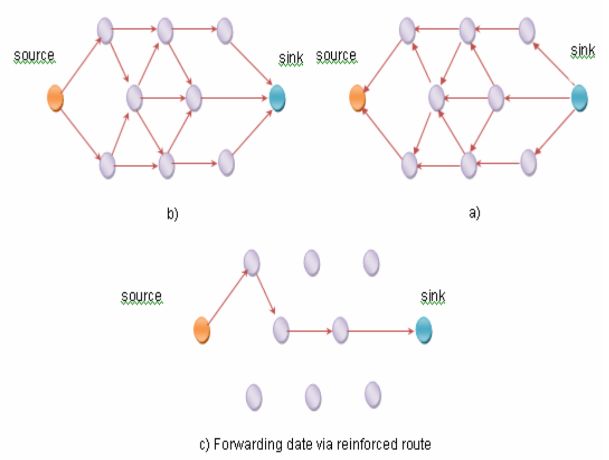
The filters used in suggestive method are divided into two types by accuracy in search and the rate of energy consumption.

1. Flooding
2. Rumor

1. Flooding filter

In Phase One of this method, the destination node will forward an interest to the network. As soon as a node senses the adaptation of the gathered data using its sensors with the forwarded interest, a discovery data (packet) will be sent to the destination and this data will try to select the best gradient with the highest quality and the best response time. It will continue until it reaches the destination, afterwards, the source node will forward the gathered data toward the destination.

The main station will request the data with an interest broadcast. Interest will describe the work that must be done by the network. Interest will be forwarded all over the network hop by hop. At the same time, each node will forward it to its neighbor nodes. As the interest is broadcast throughout the network, the routes for forwarding the acceptable data to the applicant node will be created. Each sensor receiving the interest will create a gradient 3 to the sensor nodes that had received the interest. This will continue until the time the gradient from source to the main station is created. Generally, a gradient defines an attribute and a route. Stability of gradient may vary from one neighbor to the other. Consequently, the rate of data stream will be different. Figure 4 shows the directed diffusion when the interests match the gradients. The data stream routes are a combination of some routes, and then the best route will be reinforced by local rules to prevent the flooding diffusion. In order to decrease the cost of connections, the data will be aggregated in a route. The main goal is to find a proper aggregate tree that receives the data from the sources and sends it to the main station. The main station periodically will be refreshed and will send interest to make sure the reliable transmission of data when it begins to receive data from the source(s).



c) Forwarding date via reinforced route

Figure 4. directed diffusion operation

2. Rumor filter

Directed diffusion uses flood forwarding to inject the query to the whole network, but in some cases there are only a few requests from the nodes; therefore, there is no need for the flood diffusion in this case. The flood diffusion is suggested when there are few events and many queries. The main goal is to clarify the routes for queries from the receiving nodes instead of drowning into the whole network to restore information about the events. In order to flood diffusion of events in the network, the algorithm of Rumor routing packets with higher lifetime called 'agent' is used. When a node discovers an event, it will add that event to its local table called as the 'Events table' and will produce an agent. The agent surveys the network to forward the local events to the far nodes. When a node produces a query for an event, the nodes that know the path might respond to this query by referring to their Events-table. Therefore, there is no need to flood the whole network. Consequently, it will cause a decline in the cost of connections. In other words, Buzz routing against two-phase pull keeps only one route between the source and the destination. As we can see in Figure 5, the steps decrease with the cutout of route with source searching, and the energy will be saved. On behalf of events, an agent with a definite lifetime will be created to survey the network. Meanwhile, the sink will deliver its interest to one of the neighbors and the survey will continue until the route of agent and diffusion cutout each other. With the cutout of the two routes, the table will be updated and the new path with fewer steps will be replaced.

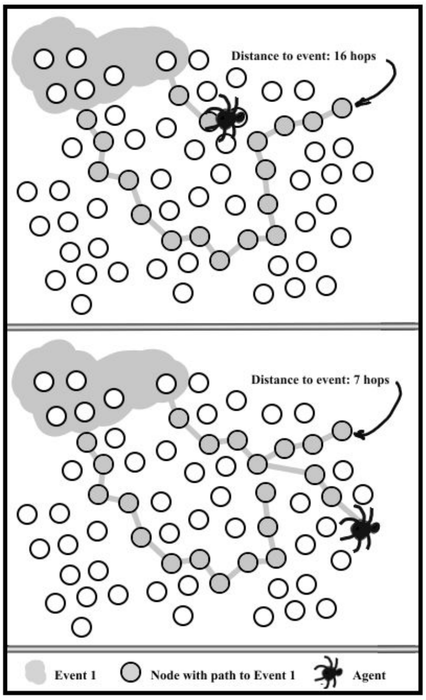

Distance to event: 16 hops
Distance to event: 7 hops
Event 1
Node with path to Event 1
Agent

Figure 5. Operation of propagation rumor

D. *Diffusion filter management*

Before the discovery stage, first the counter variable is analyzed. When the network is initiating, all the counters are set by the value of zero. When the values of the counters are equal, the two-phase pull method is used. In other words, at the time of network start due to the equality of the counters, the whole areas of the network will be searched thoroughly to discover the source in cases where the values of the counters are less than the other ones. In other words, the Rumor filtering with low energy consumption rate is used wherever there is less probability of source presence.

IV. Evaluation of Suggestive Method

E. Implementation

In order to implement the algorithm, we use the Diffusion 3.2.0 code which is given out along with ns-allionone-2.29 software. In these two packs, there are two versions of directed diffusion algorithm including diffusion and diffusion3. The diffusion version is for implementation of simple algorithm and therefore takes less detail. In this paper, with a change in filter of two-phase pull which is in Diffusion 3.2.0 algorithm, the suggestive method is implemented and using API, it is diffused as a filter toward the core.

F. Simulation scenarios

In this experiment, the number of the nodes is 250 at most, and is distributed throughout 160 * 160 square meters. The protocol 802.11 is used to simulate the wireless scenario Diffusion 3.2.0 in Ns-allinone-2.29. According to energy consumption in PCM-CIA WLAN card as ns2 and the nodes are expanded in grid.

Table I. Simulation Scenarios

Parameter	Value
Routing Protocol	Diffusion
MAC Protocol	IEEE 802.11b
Radio Transmission Power	0.660mw
Radio Reception Power	0.395mw
Radio Idle Power	0.0375mw
Sensing Power	0.0325mw
Radio Propagation Model	Two-Ray
Packet Size	100 bytes
Data Rate	1Mbps
Radio Range	90 meter
Sensing Range	13 ~ 48 meter
Area	160 m * 160m
Number of nodes	250

V. Simulation Results Survey

In this section, the results of simulation for each of the discussed scenarios in previous section are presented. The graph regarding the results is illustrated. Then the results of each simulation will be studied.

We have compared our suggestion algorithm with pervious algorithms, Two-Phase-Pull and passive clustering directed diffusion (pcdd).

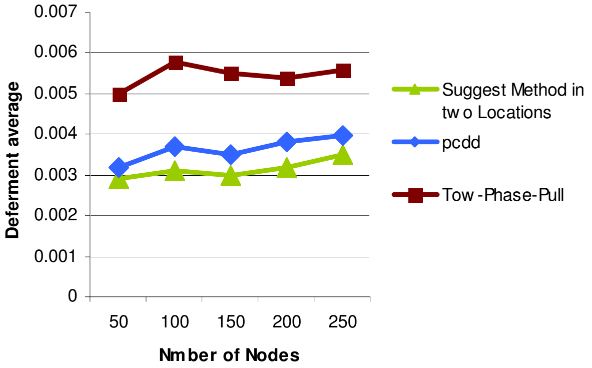


Figure 6. comparison of the amount of delay in two locations

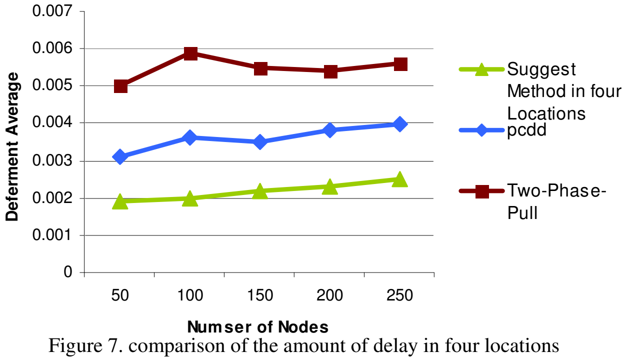


Figure 7. comparison of the amount of delay in four locations

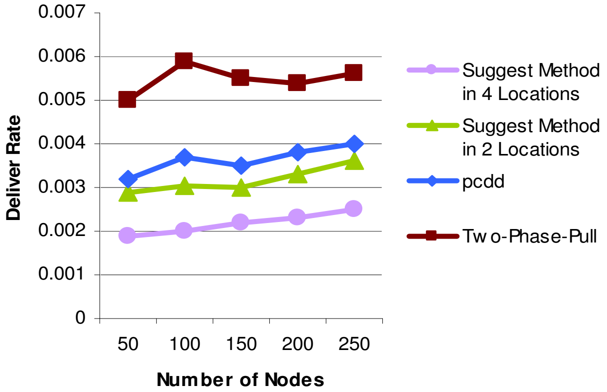


Figure 8. comparison of the amount of delivery

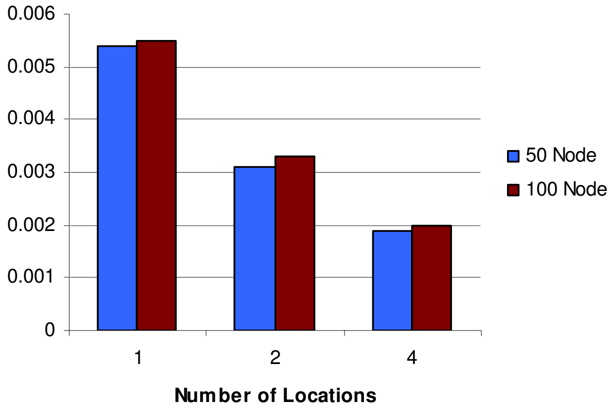


Figure 9. Comparison of the average energy consumption in one, two, and four locations

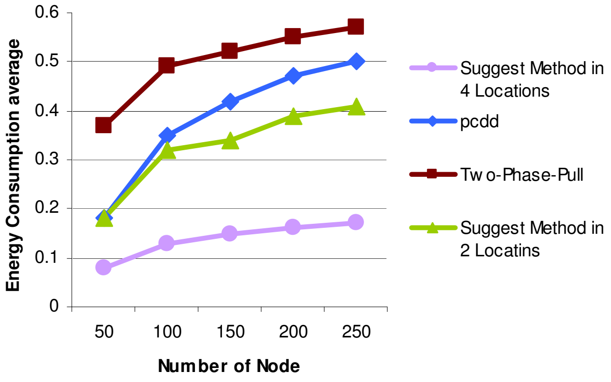


Figure 10. Comparison of the amount of energy consumption average

When we increased the number of locations, network's delay decreased. By raising the number of regions, sink could find the source sooner than pervious methods and Figure 6 and Figure 7 demonstrate this matter. Although pervious methods had fluctuations, our algorithm rose slightly; when the number of regions increased, the average energy consumption decreased. Figure 9 and Figure 10 also present that the amount of delivery decreased.

VI. Conclusion and Future Work

With increasing the number of locations in each area, fewer nodes are used and in case an interface node is spoiled, source would not be accessible. As a result, with increasing the number of locations when the node is spoiled, access to the source might not be possible. However, if the network encounters a lot of decay with the addition of locations, the delivery rate will get worse. If the locations reach the point that there is a node in each location with minimal overhead and traffic and 100% delivery, through increasing locations in case of presence of an interface node, accessing the source may not be possible. Hence in future works, this problem could be solved. For example, if eight locations were not found and the decay happened, access to the source could be improved by decreasing the number of locations to four segments. Our goal is to decrease the overhead. Using the limited Directed Diffusion, we tried to increase the network lifetime by reducing energy consumption.

Reference

[1] C. Intanagonwiwat, et al.,"Directed Diffusion: A scalable and robust communication paradigm for sensor networking," MobiCom, Rome, Italy, 2004, pp. 438-449.

[2] I.F. Akyildiz, et al., "Wireless Sensor Networks: a Survey" Computer Networks, Vol. 38, March 2002, pp. 393-442.

[3] J. Heidemann, F. Silva, and D. Strin, "Matching Data Dissemination Algorithms to Application Requirements," Proceedings of The First ACM Conference on Embedded Networked Sensor Systems (Sensys), November 2004, pp. 218-229.

[4] Y. ShyanChen, Y. WenNian, and J. PingSheu, "An Energy Efficient Diagonal-Based Directed Diffusion for Wireless Sensor Networks," Proceedings of the Ninth International Conference on Parallel and Distributed Systems (ICPADS), 2006 IEEE, pp. 445-450.

[5] R.D. Pietro, L.V. Mancini, Y.W. Law, S. Etalle, and P. Havinga, "LKWH: A Directed Diffusion-Based Secure Multicast Scheme for Wireless Sensor Networks," Proceedings of the International Conference on Parallel Processing Workshops (ICPPW), 2005 IEEE, pp. 56-67.

[6] C. Wong, M. Gouda, and S. Lam, "Secure Group Communications Using Key Graphs," IEEE/ACM Transactionson Networking (TON), 2004, pp. 16-30.

[7] D. Wallner, E. Harder, and R. Agee, "Key management formulticast: Issues and architectures," RFC 2627, IETF, June 2002.

[8] W. Ding, S.S. Iyengar, R. Kannan, and W. Rummler, "Energy Equivalence Routing in Wireless Sensor Networks," Microprocessors and Microsystems, 2006 Elsevier, pp. 467-475.

[9] V. Handziski, A. Kopke, H. Karl, C. Frank, and W. Drytkiewicz, "Improving the Energy Efficiency of Directed Diffusion Using Passive Clustering," EWSN, 2007 Springer, pp. 172-187.

[10] D. Bein and A. K. Datta, "A Self-Stabilizing Directed Diffusion Protocol for Sensor Networks," Proceedings in International Conference on Parallel Processing Workshops (ICPPW), 2008 IEEE, pp. 393-422.

[11] Z. Y. Yuan, X. Bo, and Z. Z. Ming, "Adaptive Directed Diffusion Routing in Wireless Sensor Networks Based on Application," CCECE/CCGEI, 2005 IEEE, pp. 51-58.

[12] W. Marc Lee, V. Wong, "LPT for Data Aggregation in Wireless Sensor Networks", IEEE Globecom, 2005, pp. 2069-2074.

[13] W. Marc Lee and V. Wong, "E-Span and LPT for Data Aggregation in Wireless Sensor Networks", Computer Communications, Elsevier, 2006, pp. 2506–2520.

[14] C. Min and K. Taekyoung, C. Yanghee, "Energy-efficient Differentiated Directed Diffusion (EDDD) in wireless Sensor Networks", Computer Communications, Elsevier, 2006, pp. 231-245.

[15] M. Alessia, N. Michele, P. Chiara, and V. Andrea, "Directed Diffusion Light: Low Overhead Data Dissemination in Wireless Sensor Networks", IEEE, 2005, pp. 148–159.

[16] C. Sun, H. Lee, and T. Cho "A Path Selection Method for Improving the Detection Power of Statistical Filtering in Sensor Networks", 2009, pp. 1163-1175.

[17] L. Thanh and J. Levendovszky, "A Novel Reliability Based Routing Protocol for Power Aware Communications in Wireless Sensor Networks", IEEE, 2009, pp. 1163-1175.

LRD2: *Low Resource Device Description for Energy Efficient Device Discovery*

Juan Pablo Suarez
Grenoble University
Grenoble, France
Juan-Pablo.Suarez-Coloma@imag.fr

Levent Gürgen
French Alternative Energies and Atomic Energy Commission
CEA-LETI Minatec Campus
Grenoble, France
levent.gurgen@cea.fr

Abstract—**The success of the large scale deployment of sensor actuator devices lies in their plug&play capabilities: they should be automatically discovered and ready to be used when they join to an environment. Self-description and discovery gain therefore a particular importance. Nevertheless, there is currently no largely adopted energy efficient device description and discovery standard. Existing protocols use proprietary device description models and discovery mechanisms that are incompatible between the two. This paper presents LRD2 (Low Resource Device Description), a generic description model capable of describing different kinds of device information. LRD2 implements a compression algorithm to reduce the size of description documents, thus saving energy by reducing the number of messages sent in the network. Experimental results about the performance of LRD2 are also presented.**

Keywords - WSN; Device description; plug&play; low power; energy conservation; compression algorithm

I. INTRODUCTION

Wireless sensor and actuator networks (WSAN) are used increasingly by numerous applications from various domains such as home, industrial, environmental and medical. They bring us one step closer to the “internet of things” paradigm. However, WSAN are still application-specific networks, contrary to the “general purpose” nature of the current internet. In fact, many standardization groups or industrial private solutions define stacks of protocols, targeting a particular set of domains, which are in general incompatible with the ones defined by others. This heterogeneity is the main obstacle for dynamic plug&play WSAN that is essential for successful large scale deployment of cross-domain solutions [1]. In fact in such dynamic large scale networks, devices should be self-described and self-discovered in a dynamic manner, with minimum human intervention.

Some recent efforts aim at taking on a layer-based approach, thus defining standards at different OSI layers; e.g., IEEE 802.15.4 [2] at the link layer, IETF RPL [3] at the network layer, or IETF COAP [4] at the application layer. This is an important progress towards a plug&play interoperable WSAN solution. However, at the application layer we still need to define what the devices are capable of doing and how to interact with them.

In this paper we propose LRD2, a device description model that aims to fill this gap. The main goal is to obtain a generic device description model for self-discovery of device capabilities, while taking into account energy constraints of the tiny sensor/actuator devices. The description model is based on an extensible hierarchical structure. It also implements a compression (and decompression) process. Besides being generic and extensible, the model therefore aims at being lightweight to keep the description size as small as possible. We apply a compression algorithm to the description in order to reduce its size, thus consuming less energy during its transfer to other devices in the network.

We have performed experiments to measure the efficiency of LRD2 in terms of code size, number of exchanged discovery messages, and the energy consumption for the discovery. We also compared these results against another compression mechanism, namely EXI (Efficient XML Interchange), which is a compact representation model for XML documents [5].

The paper is organized as follows: Section II presents the related work in the domain, in particular making a synthesis of related existing standards. Section III presents our proposition, LRD2, with its description model and compression algorithm. Section IV provides the results of some experiments that we have performed with LRD2. Finally, Section V concludes the paper.

II. RELATED WORK

Several WSAN standards have been defined targeting different OSI levels (from physical to application). For instance, industrial alliances such as LonWorks [6], EnOcean [7], Z-Wave [8] and Insteon [9] build solutions with a holistic approach covering requirements from the physical to the application layer. Some other standards focus on only a few layers. For example IEEE 802.15.4 [2] is a PHY/MAC layer protocol targeting low resource devices. Zigbee alliance [10] defines a set of protocols from the network to the application layer and is based on the 802.15.4 protocol. Similarly, WirelessHART [11] defines a protocol stack from network to application layer, which is also based on the 802.15.4 protocol.

	LonWorks	EnOcean	Z-Wave	Insteon	Wireless HART	ZigBee	IEEE 802.15.4	6LoWPAN	UPnP	DPWS	SensorML
Application	■	■	■	■	■	■			■	■	■
Network/ Transport	■	■	■	■	■	■		■			
Link/ Physique	■	■	■	■			■				

TABLE I. WWSAN PROTOCOLS AND OSI LAYERS THEY DEAL WITH

At the network layer, IETF specifies 6LoWPAN [12] bringing the IPv6 features to low power personal area networks on top of the 802.15.4 protocol. It is likely to be the “standard” at the network layer for personal area networks. For instance, the Smart Energy v2 of Zigbee will be agnostic to MAC/PHY protocol and will be based on 6LoWPAN.

UPnP [13] is an application layer protocol defining how to interact home appliances, in particular media devices. Similarly, DPWS (Device Profile for Web Services) [14] brings the service-oriented approach over the resource-constrained devices. UPnP and DPWS uses XML based device descriptors that are self-descriptive and expressive, yet greedy for sensor devices. SensorML [15] aims to describe powerful sensor devices rather than low power tiny wireless sensor devices.

The protocols dealing with the application layer adopt different approaches to describe devices. We can classify them into two groups according to the level of the structure of the data: **structured and semi-structured**.

The well-structured format uses fewer resources to express the information, but requires that the description receiver knows the strict structure format. On the other hand, in the case of the semi-structured format, a high level schema is enough to use the description information. However, devices need more resources to express the same information in the latter case. Protocols such as Insteon, EnOcean, Zigbee and LonWorks use a highly structured format for device description, with a few bits to express the sensor device identity and other information small in size. WirelessHART, UPnP, DPWS and SensorML provide a flexible description using a semi-structured format, mostly based on XML. The description size for these protocols is quite high to be stored and shared by very low-power sensor devices. Figure 1 shows the relation between the device description size versus the flexibility for different protocols and the position where LRD2 aims to achieve.

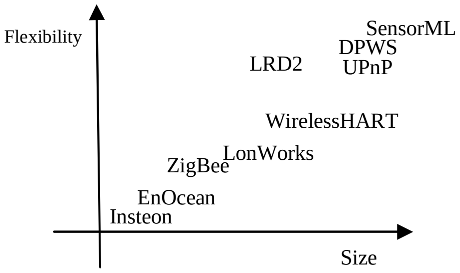


Figure 1. Description size versus model flexibility

The goal of a recent effort from the W3C, namely EXI (Efficient XML Interchange) [5], is to reduce the size of XML documents by compressing them. The idea is to assign a small binary code to the most-used elements tags, and a larger one for the less-used (instead of assigning the same quantity of bits to represent each different tag) in order to achieve a higher compactness.

The objective of LRD2 is to provide an extensible yet small in size description model. It is based on a hierarchical structure like XML that gives certain flexibility, while being less verbose in order to reach the same size of the structured data models. Similarly to EXI, LRD2 proposes a compression process for the device description with the goal of reducing the description size in the description exchange phase, therefore reducing the number of exchanged messages and saving energy consumption at devices.

III. LRD2

This section describes LRD2 device model, its compression mechanism and theory of operation of a discovery mechanism.

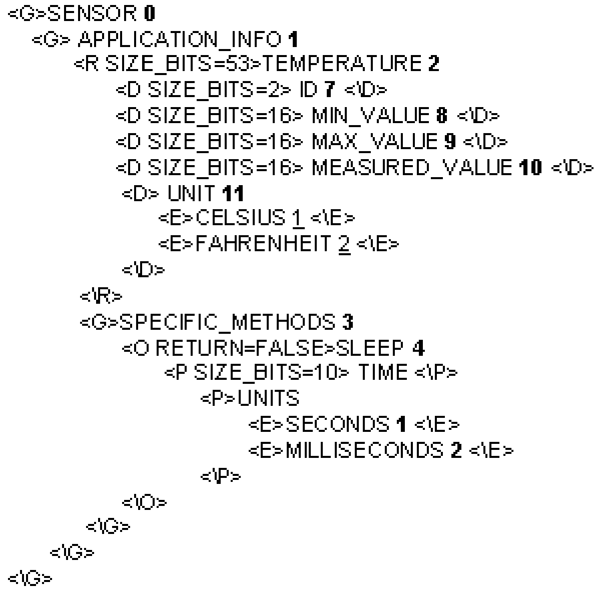
A. *Device description*

LRD2 is a flexible and hierarchical description model capable of describing different kinds of device information. The model schema is composed of the following elements:

- The **group** (G) tag defines a set of information under a common group, e.g., application specific information, network parameters, system information.
- The **resource** (R) tag represents the values of the resources in a group, e.g., temperature value, network address, OS name.
- The **detail** (D) tag allows the model to give some more detail on the resource information in terms of attributes, e.g., the minimum temperature measurable by a temperature sensor, network addressing type, OS version.
- The **operation** (O) tag is used to describe the specific operations implemented by the sensor device, such as sleep, reboot or ping. It defines a **parameter** (P) attribute is used to determine the input parameters of operations.
- The **enumValue** (E) tag is used to define possible values for resources or parameters.

All the tags are accompanied by a specifically chosen **qname** (qualified name) and an **ID** that allows the interaction between devices. The model also defines a *size-bits* attribute which gives the size used for the *resource* or *parameter* of operations.

LRD2 attempts to obtain small-sized yet extensible descriptions. While being strict within a group, it allows flexibility by giving the possibility of adding groups. Within groups the order of appearance of the information in the description has to be strictly respected using the specified number of bits (8 by default) in the *size_bits* tag attribute. Respecting the order allows us to formulate highly structured messages with a small size. Figure 2 gives a description example.



```
<G>SENSOR 0
  <G> APPLICATION_INFO 1
       <R SIZE_BITS=53>TEMPERATURE 2
            <D SIZE_BITS=2> ID 7 <\D>
            <D SIZE_BITS=16> MIN_VALUE 8 <\D>
            <D SIZE_BITS=16> MAX_VALUE 9 <\D>
            <D SIZE_BITS=16> MEASURED_VALUE 10 <\D>
             <D> UNIT 11
                 <E>CELSIUS 1 <\E>
                 <E>FAHRENHEIT 2 <\E>
             <\D>
        <\R>
        <G>SPECIFIC_METHODS 3
             <O RETURN=FALSE>SLEEP 4
                  <P SIZE_BITS=10> TIME <\P>
                      <P>UNITS
                           <E>SECONDS 1 <\E>
                           <E>MILLISECONDS 2 <\E>
                      <\P>
             <\O>
         <\G>
     <\G>
<\G>
```

Figure 2. A simple description example

B. Description compressing/decompressing

Once the description is complete, an ID is designed for each word used. To achieve a higher compactness, the words used in the description are uppercase represented in 6 bits. The compressed description is composed of three parts:

The first part is the new character codification; instead of using 16 bits as UTF-8 or 8 bits as ASCII, we choose the minimal amount needed to code all the used characters in the description. In order for this message to be understandable by other devices, the first part is built as follows: 7 bits representing a digit *k* in ASCII, *k* being the minimal number of bits used for the chosen character codification, for each used character, the 7-bits ASCII representation followed by the new character codification in *k* bits. If *k* is 5, that means that the used characters are less than 64, if A and B are used characters coded as 00001 and 00010 respectively, the first codification message could start as <u>0110101</u> <u>1000001</u> 00001 <u>1000010</u> 00010 (followed by all the other coded characters), being the underlining bits an ASCII representation.

The second part of the compressed description is the dictionary. The dictionary is a stream of bits. For separating two different words (the words have a variable size) the character ":" is used. The three first words in the dictionary message are: *n1*, *n2* and *n3*; *n1* being the number of bits assigned to code the ID used in the interaction messages to access the *group*, *resource*, *detail* or *operation*; *n2* is the number of bits used to represent the size of the biggest enumValue group; and *n3* is the number of bits used to code each word in the description. After these 3 first words, each other word in the dictionary is written followed by its respective ID in hexadecimal mode and separated also by the character ":". All the characters used in the second compression part are written using the selected character codification of the first compression part.

For the third and last part of the compression process, the complete description is written using *n3* bits with each word representing the word ID assigned in the second compression part. The compression process is done only once (or each time that the description changes) directly in the description owner device. Another approach is to make the description compression and store it compressed in development time into the device, thus saving space. The compressed description is shared with the interested devices and decompressed only when it arrives at its destination.

When a device joins a network, or when it is contacted by another device, a presentation message is sent containing the three parts of the compressed description. Using these three parts, the description can be decompressed and rebuilt. The information supplied in the description is enough to interact with the sensor device containing sensor parameters and the list of operations that can be executed on the device, besides simple get/set operations to retrieve/modify parameter values. For the construction of the interaction messages, the *n1* bits ID is used to specify the information to be affected (group, resource, detail, operation). The *n2* bits are also used in the case where *enumValue* is specified. The answer interaction messages come with the *n1* bits ID to match with the interaction message.

IV. Experimentation

For the validation of the LRD2 approach we have constructed device descriptions of different sizes and measured various values such as compressed document size, compression duration, and energy consumed on sensors to perform the compression.

We used EXI as a reference and compared the measured values with the ones obtained by EXI. We used the EXIficient v0.5 [16] implementation of the EXI. We conducted the tests over a desktop PC computer with 3 GB of RAM and processor of 2.66 GHz. Figure 3 shows the size of documents after compression by LRD2, EXI Schema-less and EXI Schema-Informed. In fact, EXI can function in 2 different modes: i) schema-less where there is no knowledge on the document's XML schema; ii) schema-informed where the schema of the document is used to optimize the compression. In the latter case, the schema is compressed using EXI schema-less algorithm, and then the schema is decompressed and used for decompressing the schema-informed compressed description.

We observed compression ratios varying from 0.71 to 0.26 depending on the size of the documents we used. The ratios we obtained are very close to the ones of the EXI's schema-less compressed documents. EXI Schema-Informed documents have greater size at the beginning as they also contain the schema of the document.

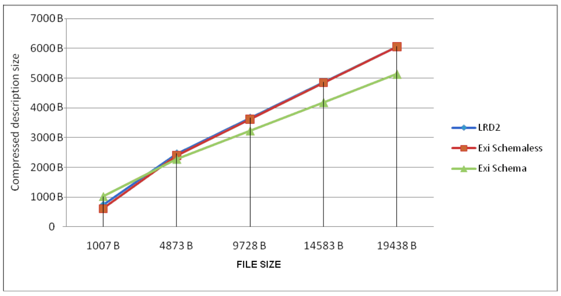


Figure 3. Compressed documents size

We also measured the description compression times with both LRD2 and EXI. Figure 4 shows the values we obtained. Even if the execution time for LRD2 compression is slightly longer than that of EXI, considering the code size of the two solutions (50 KB for LRD2 and 1500KB for EXI), LRD2 is preferable for resource-constrained devices having a small memory capacity. LRD2 is simple and lightweight and worked seamlessly over all the tested Java environments (Standard and Micro Editions).

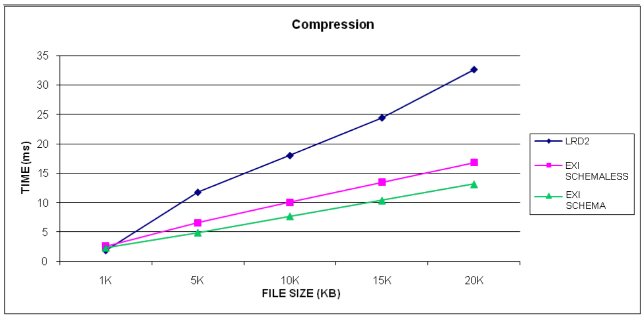


Figure 4. Compression time (milliseconds)

For the battery consumption experimentation on sensors, we used SunSPOT sensors with the following processor board specifications: 180 MHz 32 bit ARM920T core - 512K RAM/4M Flash; 2.4 GHz IEEE 802.15.4 radio with integrated antenna; 3.7V rechargeable 720 mAh lithium-ion battery. We measured on a SunSPOT node the battery capacity difference between before the start of the compression process and after the transmission of the compressed description. Figure 5 shows the measurements we obtained. We observe that energy consumption is linearly increasing w.r.t the description size and consumes between 0.04 % and 0.5 % of overall battery for a description size of between 1KB and 20KB.

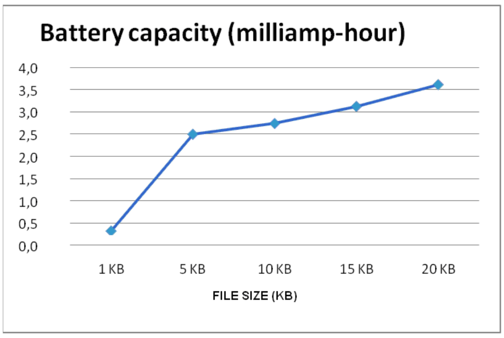


Figure 5. Battery consumption of compression (milliampere-hour)

V. Conclusion and Future Works

For successful large-scale deployment of WSAN applications, sensor and actuator devices need to be automatically discovered and ready to be used once they are discovered. Device descriptions have an important role in defining generic yet extensible descriptions to take into account new devices and technologies appearing every day in the market, as well as their constraints in terms of resources.

We have proposed a simple device description model and a compression mechanism. The semi-structural and hierarchical model lets us achieve a tradeoff between the flexibility and the size of descriptions. Thanks to its compression mechanism, for descriptions ranging in size from 1KB to 20KB we were able to obtain compression ratios from 0.71 to 0.26; compression times from 1.86ms to 32.6ms; and energy consumption values from 0.32mAh to 3.61mAh. These are reasonable values considering the fact that the compression is performed (mostly) only once. The values are close to the ones we obtained with the EXI implementation; while furthermore the code size of LRD2 being 3.3% of the one of EXI.

Our next plan is to further evaluate the energy consumption, not only at the device level but also at the whole network level. In such multi-hop networks, the energy consumed at a node to compress a description may be largely dominated by the potential energy gain in the multi-hop network, thanks to the decreasing number of exchanged messages.

Acknowledgment

This work has been partially supported by the European project OUTSMART FP7-2011-ICT-FI- 285038.

References

[1] L.Gürgen, J.Nyström-Persson, A.Cherbal, C.Labbé, C.Roncancio, and S.Honiden. Plug&manage heterogenuous sensing devices. In Proceedings of the 6th International Workshop on Data Management for Sensor Networks (DMSN'09), in conjunction with VLDB'09. 2009.

[2] IEEE P802.15^{TM} Working Group, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)", IEEE Standard, March 2009

[3] RPL: IPv6 Routing Protocol for Low power and Lossy Networks, draft-ietf-roll-rpl-19 (work in progress), March 2011

[4] Constrained Application Protocol (CoAP), draft-ietf-core-coap-05 (work in progress), March 2011

[5] Efficient XML Interchange (EXI) Format 1.0, W3C Recommendation, March 2011

[6] LonMark. Device Interface File Reference Guide, Revision 4.402, 2009

[7] EnOcean Alliance – Technical Task Group Interoperability, "EnOcean Equipment Profiles (EEP)", Version 2.1, January 2011

[8] Z-wave Protocol Overview, Version 4, May 2007

[9] P. Darbee, Insteon Command Tables, revision 20070927a, 2007.

[10] ZigBee Specifications, http://www.zigbee.org/

[11] WirelessHart , "Wireless Command Specification", September 2007

[12] IPv6 over Low Power WPAN (6LoWPAN). November 2009

[13] UPnP Device Architecture, Version 1.1, October 2008

[14] Devices Profile for Web Services (DPWS), Version 1.1, July 2009

[15] OpenGIS Sensor Model Language (SensorML), Implementation Specification", July 2007

[16] Open source implementation of W3C Efficient XML Interchange (EXI). http://exificient.sourceforge.net

Energy-aware Clustering with Variable Ranges in Wireless Sensor Networks

Faruk Bagci
Department of Computer Science and Engineering
German University in Cairo
Cairo, Egypt
Email: faruk.bagci@guc.edu.eg

***Abstract*—Traditional wireless devices communicate directly with a beacon or base station located in range of the device. Multi-hop messages are still a rarity in wireless communication. Wireless sensor network protocols often are based on peer-to-peer infrastructure, where messages traverse long distances through the network to reach a certain destination. A flat infrastructure is hard to manage regarding routing and scalability, if number of nodes increase drastically. Because of battery operated nodes, energy-effective mechanisms are very important in wireless sensor networks. Clustering brings hierarchy into the network and can save energy since nodes within a cluster usually communicate locally in a short range. Messages sent to large distances are handled by cluster-heads routing them through an inter-cluster backbone. A heterogeneous infrastructure with a large number of simple and cheap sensor nodes and only a small percentage of more powerful cluster-heads is beneficial but not necessary. This paper presents a new clustering approach called *Variable Ranges Protocol* that provides basic features to modify the range of each node. A dynamic transmission range adaption protocol substantially prolongs the lifetime of the nodes through energy efficient communication without significantly decreasing the node connectivity. The VR protocol is implemented combining several MAC protocols for local communication within a cluster.**

***Keywords*-Wireless sensor network; cluster architecture; energy efficiency; connectivity; wireless communication;**

I. Introduction

The study of sensor networks, while being a research field in and of itself, forms a basis for various areas where the collection of environmental data through sensors is essential (*e. g.*, security, traffic control, ubiquitous computing, *etc.*). The combination of sensing/sensoring, computational aspects, and communication solutions provides for a broad range of applications such as smart hospitals, intelligent battlefields, earthquake response systems, and learning environments [1] [2]. Generally, the term *sensor network* has come to describe a dynamically self-organizing collaborative network of widely distributed, tiny, low-cost, sensoring nodes (“smart dust”) that are capable to cover an area and automatically communicate the collected data to a beacon or base station over multi-hop paths.

Sensor nodes are usually tiny, self contained, battery powered devices. Under normal circumstances it is impossible to replace or recharge these batteries, therefore the lifetime of a wireless sensor network is intrinsically restricted by the initially available power in each individual node, making power consumption considerations an essential part of any new protocol design. Similarly, very small memory, low processing power, and a limited communication bandwidth, all in comparison to traditional wireless devices, further restrict the options. Also, high failure rates, occasional shutdowns, and sporadic communication interference force continuous dynamic changes upon the topology. Finally, the sheer number of individual sensing devices of a sensor network, ranging from hundreds to thousands, make it infeasible to rely on previous solutions of ad-hoc networking protocols such as. For example flooding-based standard routing schemes for ad hoc networks simply do not scale adequately [3].

In [4] we proposed a security architecture for wireless sensor networks called *SecSens* which fulfills security requirements on multiple levels. SecSens focuses mainly on three security aspects: key management, secure routing, and verification of sensor data. The sensor network in SecSens consists of clusters, each containing a number of simple sensor nodes and one powerful node that acts as a cluster-head. Sensor nodes connect directly to the cluster-head, i.e. routing in clusters is not necessary. A node can be a member of several clusters at the same time. All cluster-heads form together an inter-cluster network used for sending messages to base stations. We assume that sensor nodes do not change their position once they are attached to a location. SecSens works with multiple base stations in order to avoid single-point-of-failures (see Figure 1). In the first version of SecSens clusters were built in the initial phase and remained unmodified for the whole lifetime of the network. Furthermore, all nodes used the same sending configuration, i.e. transmission power and range were set to maximum on each node. This paper describes an extension of this approach with dynamic features. The new variable ranges (VR) protocol optimizes the communication range of each node. This optimization results in more efficient usage of energy throughout the overall sensor network.

The next section describes related energy efficient communication approaches for sensor networks. Section III introduces the variable ranges protocol. We evaluated the new architecture in a wireless sensor network simulator. Section IV presents the evaluation results. The paper ends

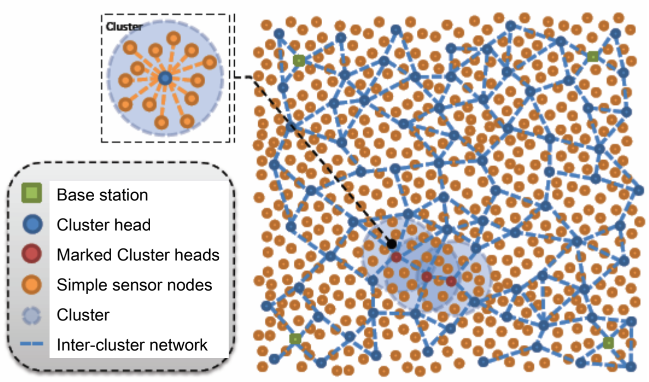


Figure 1. Basic sensor network architecture

with the conclusion.

II. Related Work

The Sensor-MAC (S-MAC) protocol [8] is an energy efficient communication protocol for wireless sensor networks. S-MAC is a slot-based protocol where each sensor node has alternating sleep and awake phases. The network is divided into clusters and all members of a cluster are awake or asleep at the same time. All cluster nodes exchange schedules in an initial phase. Within a cluster only one schedule is used, i.e. the schedule of the first node that sent a schedule. If a node receives multiple schedules it follows all of them. Such nodes have a higher energy consumption. Within an awake phase all cluster nodes contend with each other for medium access. The contention mechanism of S-MAC is the same as that in IEEE 802.11, i.e., using RTS (Request To Send) and CTS (Clear To Send) packets. S-MAC needs a strict timer synchronization in order to achieve correct functionality. Periodic synchronization among neighboring nodes is performed to correct their clock drift. An extension of S-MAC by *adaptive listening* is described in [9]. If a node A notices an ongoing communication of node B whom it wants to send a message, it sleeps the time until B is ready.

A modification of S-MAC called Timeout-MAC or T-MAC is introduced in [10]. In S-MAC all nodes need to be awake in the contention phase even if they have nothing to send or receive. T-MAC uses a specific timer T_A to shorten the awake phase if the node does not need to communicate. Obviously the T_A is smaller than the contention phase, thus the energy consumption is reduced. But if the timer T_A is chosen too small, the node sleeps early missing possible message requests of other nodes. This *early sleeping problem* could even lead to unfairness. In further extensions of T-MAC this problem is solved by *future request to send (FRTS)* messages, but this increases again the energy consumption. Nevertheless T-MAC gains better energy results compared to S-MAC.

Token ring [11] can be classified as a combination of TDMA and contention-based. Each node has its own time slot (token holding time) where it has to manage the communication with multiple contending nodes. The *Wireless Token Ring Protocol (WTRP)* [12] was developed for mobile adhoc networks. All nodes build a single ring in the initial phase. The aim of WTRP is to maximize the throughput and minimize the latency without restraining the mobility. Energy efficiency is not considered in WTRP because the nodes are mobile devices with strong energy resources like Laptops or PDAs. A mapping of WTRP on wireless sensor networks is E^2*WTRP* that is described in [13]. E^{2}WTRP aims to enhance the energy balance by dynamic adaptation of the token holding time. An active node can send more messages if the token holding time is increased. The frequency of token hand-over is decreased at the same time that reflects in lower energy consumption. *ESTR* [14] is an energy saving token ring protocol for wireless sensor networks that introduces sleep periods for nodes which does not need to send or receive messages. This leads to a very good energy balance.

III. Variable Ranges Protocol

A sensor node consumes most of the energy in its active mode. The energy cost rises enormously if the node uses radio communication. The energy consumption of the microprocessor Texas Instruments MSP430F149, which is used in several sensor boards, is 1.6 μA in sleep mode and rises to 280 μA in active mode at 1 MHz. In [8] the energy rate between active:receive:send is determined as 1:1.05:1.4, i.e. the energy consumption of sending a message outweighs other tasks. It is plausible that the energy consumption can be highly decreased by establishment of sleep intervals and avoidance of unnecessary packet sending. Using sleep intervals can lead to a contrary effect, i.e. the data exchange is reduced to a shorter time, which can cause higher packet collisions.

The energy consumption of sensor nodes that send with maximum transmission power lies significantly higher than with reduced power. Decreasing transmission power results in exponentially decreased energy consumption. Therefore, the Variable Ranges (VR) protocol saves energy by adjusting and optimizing the signal strength to particular circumstances of the sensor network in order to extend the lifetime of nodes. Additionally, the initial state of reduced transmission power of nodes ensures that complexity of network is low. With high signal strength nodes are confronted with frequent interferences and redundant paths within the network. Low signal strength means that number of neighboring nodes is less, i.e. probability for message collisions decreases.

Regarding the security architecture, the number of neighbors is also an important parameter. Each cluster-head has to manage several keys with each other neighboring sensor node and cluster-head for securing the communication

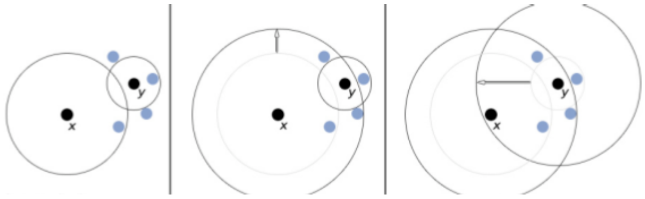


Figure 2. Range adjustment in VR protocol

and ensuring authentication. More neighbors means higher management effort and more storage space, as well as more encryption and decryption processing. All of this result again in increased energy consumption. Therefore, it is essential that the security architecture works hand in hand with the underlying communication protocol.

In the initial phase of the VR protocol, the nodes search for neighboring nodes starting with a minimum signal strength. For this reason, each node sends a *DiscoverNodes* message containing its own range parameter and ID. Then the node waits a certain time to get a response. The waiting time is also dynamic, i.e. the time is low, if the range is low and increases, if the range increases. The reason for this is that a node with a low range will reach less neighbors. Therefore, there is no need to wait a long time for a response. The nodes increase stepwise their transceiver power and send new discovery messages until a pre-configured number of nodes is found. A node which receives a *DiscoverNodes* message of an unknown node, extracts the range information of its seeking new neighbor. In the next step, the node compares the received range value with its own range. If the own range is lower, the node increases its range and sends a *DiscoverNodesReply* message back. Since the signal strength of the nodes would increase in this way until all nodes would settle at the range of the largest distance between two nodes, the VR protocol performs only a temporary range adjustment. This means that nodes discard the adaptation after a certain time and return again to their previous values.

Figure 2 illustrates this adjustment scheme. Assuming a maximum number of neighbors is set to three in this figure, it would be unfavorable for node *y* to use the range parameter of node *x*, since it can reach three neighbors with a much lower signal strength. For this reason, *y* will only increase range temporarily to answer node *x* and return to its previous range, in order to proceed with its own search. Cluster-heads find in this way a minimum number of other cluster-heads and as well as sensor nodes.

Actually, the aim of each cluster-head is to be reachable through the inter-cluster network by at least one base station. After the initial phase, each base station sends a broadcast through the new built network. This sink message is also important to generate new keys for further authentication. In [4] this key generation is described in detail. Therefore, reachability of base stations is essential to establish the security architecture in the sensor network. If a cluster-head does not receive a broadcast message after a certain time, it starts a new search phase to find new cluster-heads. The cluster-head uses this time a different message *DiscoverNewNodes*. It first uses the current range, since there could be cluster-heads which are in range, but not discovered. Cluster-heads who receive such a message, adjust their signal strength to answer, but keep their new range value this time. If the node does not get any answers, it increases its range and repeats the procedure until it finds new connections. Figure 3 shows the *TryToConnect* phase. You can see on the right side of the figure, that after the initial search phase, several local cluster networks are established, but not all of them can reach a base station marked here as green squares at the four corners. On the left picture you can recognize that the connectivity is enhanced after the *TryToConnect* phase, but nevertheless there are still local clusters remaining unreachable by any base station. The reason is that nodes are deployed randomly. There is a small probability that some nodes cannot reach a base station, even if transmission power is set to the maximum or due to message collisions in the initial phase.

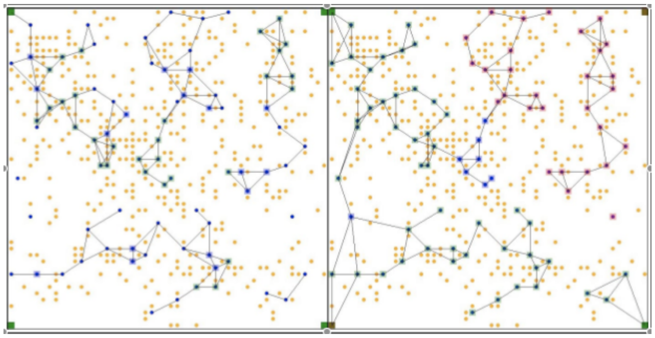
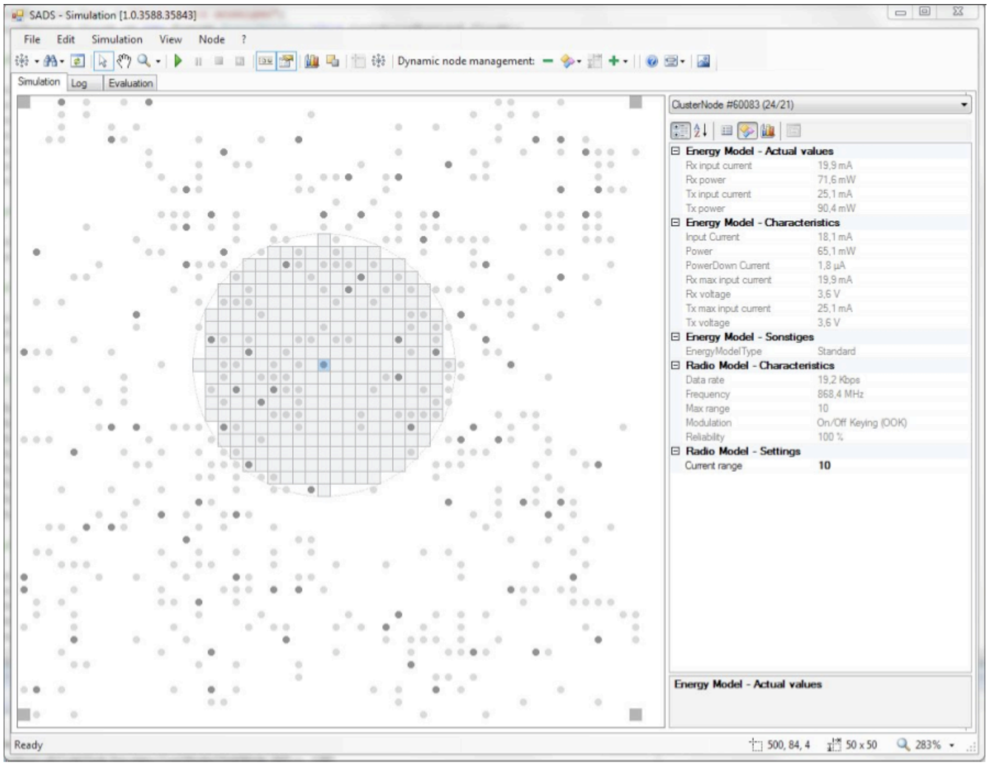


Figure 3. Initial phases of VR protocol: a) neighbor search b) TryToConnect

Cluster-heads check periodically their neighborhood for node losses or new arriving nodes. During lifetime the VR protocol ensures that nodes can dynamically adapt to changes in their environment. This network adaptation goes hand in hand with security adjustments.

Additionally, routing information is updated by cluster-heads after each VR adjustment phase. Simple sensors do not need routing capability, because they exclusively communicate with the cluster-head. Routing is used only within the inter-cluster network established by cluster-heads. Our architecture uses probabilistic multi-path routing based on the level values to forward messages from cluster-heads on the way to the corresponding base station. Cluster-heads build up a trust matrix, where each transmission to its neighbors is recorded. Based on this trust information, cluster-heads calculate a probability value and write it into the packet header. This value is used to decide in which direction the packet has to be send. Each cluster-head



ClusterNode #60083 (24/21)

Energy Model - Actual values	
Rx input current	19,9 mA
Rx power	71,6 mW
Tx input current	25,1 mA
Tx power	90,4 mW
Energy Model - Characteristics	
Input Current	18,1 mA
Power	65,1 mW
PowerDown Current	1,8 µA
Rx max input current	19,9 mA
Rx voltage	3,6 V
Tx max input current	25,1 mA
Tx voltage	3,6 V
Energy Model - Sonstiges	
EnergyModelType	Standard
Radio Model - Characteristics	
Data rate	19,2 Kbps
Frequency	868,4 MHz
Max range	10
Modulation	On/Off Keying (OOK)
Reliability	100 %
Radio Model - Settings	
Current range	**10**

Energy Model - Actual values

Figure 4. The Simulator GUI

modifies the probability value and sends the message over the most trustworthy route. Furthermore, our architecture provides passive participation, i.e. sensor nodes listen to packet transmissions of their neighbors. If cluster-head *u* detects a packet addressed to its neighbor *v*, and recognizes that *v* is not forwarding the message, *u* takes responsibility with a certain (low) probability. Also, if *u* assumes that *v* forwards the message to a non-existent node, *u* takes care of transferring.

IV. Evaluation

To evaluate the efficiency of our variable ranges security architecture we implemented a simulation tool where it is possible to establish different sizes of sensor networks. Figure 4 shows the GUI of the simulator. The simulation is divided into three phases: node distribution, initialization of network, and report sending. In the first phase, a predefined number of nodes is distributed randomly over a given area. Sensor nodes and as well as cluster-heads are deployed after setting for each a maximum transceiver range.

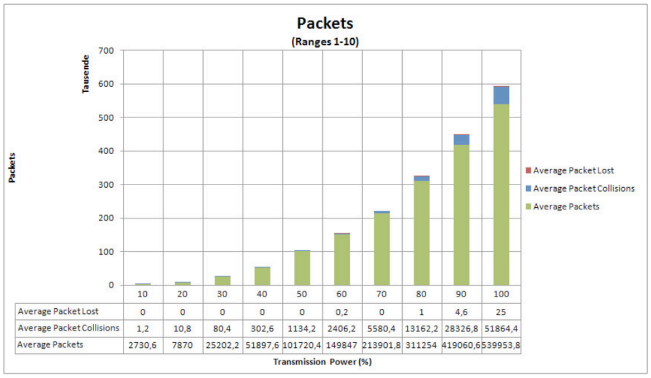
Basic parameters for the network are total number of nodes, initial node range, initial node energy, and network density. Type, range, and position of nodes can be changed easily using the simulator GUI. Furthermore, new nodes can be added or existing nodes can be deleted before the next phase of the simulation is started. Figure 4 shows a screenshot after the first phase. Dark circles are cluster-heads whereas light dots are simple sensor nodes. The squares at the corners represent again four base stations. In this case one cluster-head is selected and you can see the communication range of the current node.

The second phase initializes the network based on a communication protocol. We implemented three protocols that can be selected by the user in the beginning of this phase. These are the SMAC protocol, the energy saving token ring protocol (ESTR), and the variable ranges (VR) protocol. The user can change a set of parameters depending on the selected communication protocol, e.g. cluster size, timer settings, update periods. In this phase security and routing information is exchanged, too.

At the end of initialization, the network is established and nodes can start to exchange secured messages. This is simulated in the last phase by randomly generated reports that are sent to base stations. The user can halt the simulation at any time, in order to change parameters for nodes. For example, one can turn off a node to simulate a node loss. It is also possible to simulate a compromised node that sends false reports into the network.

Nodes consume energy for processing data, like encryption and decryption, and sending or receiving messages. For some communication protocols nodes can switch to a sleep mode, where energy consumption is minimal. Our simulator bases on an energy model that uses specifications of real sensor boards: ESB 430/1 and MSB-430 of Freie University Berlin [15].

In a first evaluation we measured the number of messages sent in the initialization phase using the VR protocol. We performed several simulations where we modified the maximum range of nodes in order to get average number of sent packets, collisions, and lost packets. Figure 5 shows the results of a network with 500 nodes. Traffic load increase with higher range of single nodes, because nodes can reach more neighbors to exchange messages with.

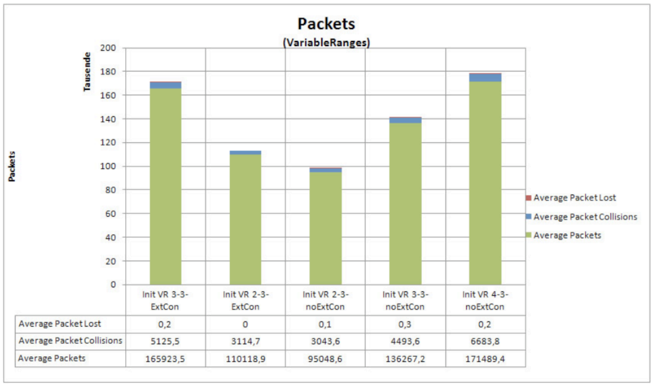


	10	20	30	40	50	60	70	80	90	100
Average Packet Lost	0	0	0	0	0	0,2	0	1	4,6	25
Average Packet Collisions	1,2	10,8	80,4	302,6	1134,2	2406,2	5580,4	13162,2	28326,8	51864,4
Average Packets	2730,6	7870	25202,2	51897,6	101720,4	149847	213901,8	311254	419060,6	539953,8

Figure 5. Traffic load in relation to signal strength

The VR protocol can be initialized with several parameters. As mentioned in the previous section, the VR protocol continues to search for new neighbors by increasing the transmission range. Using the simulator the user can set the maximum number of neigboring cluster-head and sensor for each node in the network, e.g. setting the number to three would stop the search after finding three cluster-heads in the neighborhood. In some cases, this would lead to a low connectivity, since nodes which could not join a cluster group would be disclosed. Therefore, VR protocol offers a second optimization step that was described in the previous section (*TryToConnect*). Figure 6 shows number of sent packets, packet collisions, and lost packets using different configurations of VR protocol. The notation *Init VR 3-3-ExtCon* means that each cluster-head searches for new neighbors until 3 other cluster-heads and 3 sensor nodes are found and the *TryToConnect* mode is turned on. It is clearly seen that packet collision in VR protocol is very low and there are nearly no packet losses, since the communication range is very reduced. The less packets are sent, the less energy is consumed. In Figure 6 the configuration *Init VR 2-3 noExtCon* seems to be the optimal configuration.

But regarding the connectivity this is not the best choice. Figure 7 shows the connectivity for each configuration. It is clearly seen that the connectivity for the configuration *Init VR 2-3 noExtCon* is only %14.15, i.e. only a small number of cluster-heads can actually reach a base station.



	Init VR 3-3-ExtCon	Init VR 2-3-ExtCon	Init VR 2-3-noExtCon	Init VR 3-3-noExtCon	Init VR 4-3-noExtCon
Average Packet Lost	0,2	0	0,1	0,3	0,2
Average Packet Collisions	5125,5	3114,7	3043,6	4493,6	6683,8
Average Packets	165923,5	110118,9	95048,6	136267,2	171489,4

Figure 6. Traffic load for different configurations of VR protocol

Turning on the *TryToConnect* modus brings only an increase of %10 in connectivity. Only after increasing the number of neighboring cluster-heads leads to reasonable results. Even without the second optimization phase, VR protocol can reach nearly %90 connectivity.

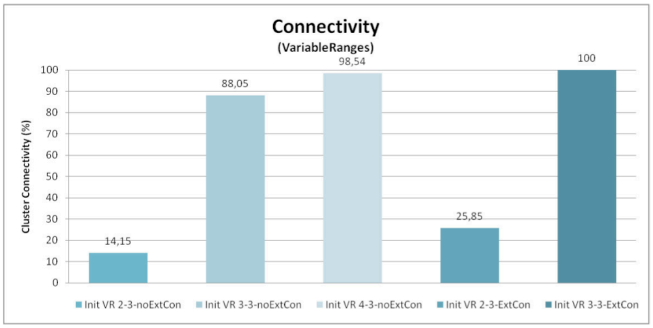


Figure 7. Connectivity for different configurations of VR protocol

As mentioned in the previous section, the complexity of sensor network is much lower using VR protocol. On the left side of Figure 8 the network was established with maximum node range. It is clearly seen that in dense areas of the network, the number of different connections is rather high. In a second simulation, we used the VR protocol to establish the network. As seen on the right part of Figure 8 using the VR protocol lowers the complexity of the network.

The optimal usage of signal strength in VR protocol shows its advantages also in energy consumption. Figure 9 illustrates the energy consumption for sending reports from a sensor node to the base station. Level represents here the distance between sending node and nearest base station, e.g. level 6 means that messages traverse six intermediate cluster-heads until they reach the base station. We performed the energy measurement in four different networks with the same size. For the first three networks the maximum range parameter of each node was set to a fixed value, i.e. range 6 stands for %60 maximum signal strength. The last

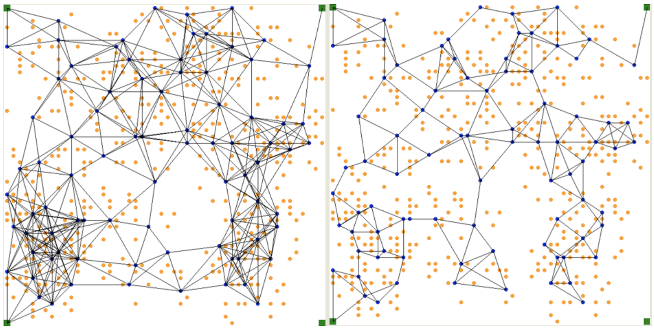


Figure 8. Network complexity without (left) and with VR protocol (right)

network used the VR protocol with at least three cluster-head and three sensor node neighbors and a further optimization step to increase the connectivity (*VR 3-3 ExtCon*). One can clearly see that the VR protocol has the best energy balance leading to a longer lifetime of the network.

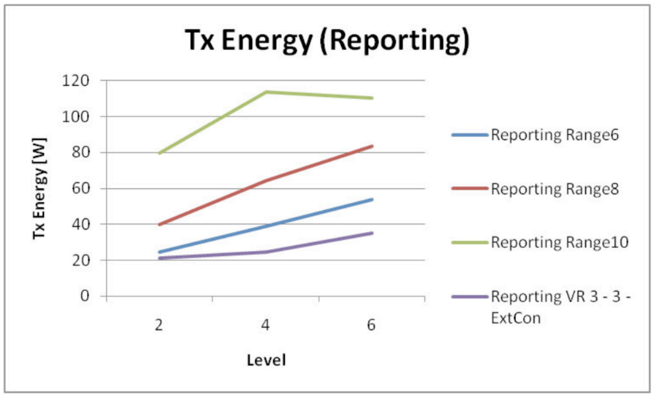


Figure 9. Energy consumption for reporting

V. Conclusion

This paper presented the Variable Ranges protocol for wireless sensor networks. Cluster architectures offer a good basis for scalable and energy-efficient protocols. Using hierarchy of cluster-heads and sensor nodes, it is possible to limit the range of each node and to exploit multi-hop communication. By dynamically adapting the range of each node, the network can be established with low complexity, but still with high connectivity. Since nodes do not sent messages with full transmission power, the energy consumption decreases considerably. This results in an extended lifetime of the overall sensor network.

References

[1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta, GA, USA, September 2002.

[2] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[3] P. Downey and R. Cardell-Oliver, "Evaluating the Impact of Limited Resource on the Performance of Flooding in Wireless Sensor Networks," in *Proceedings of the 2004 international Conference on Dependable Systems and Networks*, Washington, DC, USA, June 2004.

[4] F. Bagci, T. Ungerer, and N. Bagherzadeh, "Multi-level Security in Wireless Sensor Networks," *International Journal On Advances in Software*, vol. 4, no. 6, 2010.

[5] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, and David E. Culler, "SPINS: Security Protocols for Sensor Networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[6] Donggang Liu and Peng Ning, "Multilevel μTESLA: Broadcast authentication for distributed sensor networks," *Trans. on Embedded Computing Sys.*, vol. 3, no. 4, pp. 800–836, 2004.

[7] Sencun Zhu, Sanjeev Setia, and Sushil Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," in *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, New York, NY, USA, 2003, pp. 62–72, ACM Press.

[8] Wei Ye, John Heidemann, and Deborah Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, New York, USA, June 2002, vol. 3, pp. 1567–1576.

[9] W. Ye, J. Heidemann, and D. Estrin, "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, June 2004.

[10] Tijs van Dam and Koen Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, Los Angeles, California, USA, Nov. 2003, pp. 1567–1576.

[11] IEEE, *IEEE CS, Token Ring Access Method and Physical Layer Specifications. ANSI/IEEE Standard 802.5*, 1985.

[12] M. Ergen, D. Lee, R. Sengupta, and P. Varaiya, "WTRP - Wireless Token Ring Protocol," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 6, pp. 1863–1881, Nov. 2004.

[13] Zhenhua Deng, Yan Lu, Chunjiang Wang, and Wenbo Wang, "E^{2}WTRP: An Energy-Efficient Wireless Token Ring Protocol," in *Proceeding of the IEEE conference on Personal, Indoor, and Mobile Radio Communications*, Barcelona, Spanien, 2004, pp. 398–401.

[14] F. Bagci, T. Ungerer, and N. Bagherzadeh, "ESTR - Energy Saving Token Ring Protocol for Wireless Sensor Networks," in *Proceedings of the International Conference on Wireless Networks (ICWN '08)*, Las Vegas, NV, USA, July 2008.

[15] http://www.scatterweb.com, *ScatterWeb Homepage*, 2007.

Distributed Search of Various Backbones in Wireless Sensor Networks

V. Boudet
LIRMM
University
Montpellier 2
Montpellier,
France
boudet@lirmm.fr

S. Durand
LIRMM
University
Montpellier 3
Montpellier,
France
sdurand@lirmm.fr

L. Gönczy
MIT
BME
Budapest, Hungary
gonczy@mit.bme.hu

J. Mathieu
IUT Rodez
University
Toulouse 1
Rodez, France
jerome.mathieu@iut-rodez.fr

J. Palaysi
LIRMM
University
Montpellier 2
Montpellier,
France
palaysi@lirmm.fr

Abstract— **In this paper, we are interested in enhancing lifetime of wireless sensor networks trying to collect data from all the nodes to a "sink"-node for non-safety critical applications. Connected Dominating Sets are used as a basis for routing messages to the sink. We present a simple distributed algorithm, which computes several CDS trying to distribute the consumption of energy over all the nodes of the network. The simulations show a significant improvement of the lifetime of the network.**

Keywords - sensor network; lifetime; distributed algorithm

I. Introduction

In this paper, we investigate the communication efficiency in wireless sensor networks (WSN), which are consisted of sensor with a limited energy resource (batteries). Each sensor is able to communicate with a few other sensors in its neighborhood within its communication range, which we assume to be roughly the same for all sensors (however, this assumption is not restricting the application of our algorithm). The sensors regularly perform measurements and measured data is collected to a single special node of the network called the sink (this operation is called gathering). Our goal is to maximize the number of gatherings that can be done by the network. For each gathering, the set of nodes used for transmissions have to be connected and to dominate the graph associated to the network (backbone). We also assume that there is no central entity to compute optimal routing for communication, therefore we are interested in developing distributed algorithms for this purpose.

In Sections II and III, we present related work, the model and the main assumptions we made in order to get realistic results. Then a distributed algorithm is proposed in Section IV to compute backbones, which distributes the use of sensors for transmissions to maximize network lifetime. Section V shows experimental results achieved by simulations about the lifetime of the network on grid and random topologies. We conclude and give tracks for future works in Section VII.

II. Related Work

Because of the critical importance of energy saving in WSN, literacy about this subject is extensive. In order to increase the lifetime of a WNS, one typical way is to use Connected Dominating Sets (CDS) also called *backbones* to route the messages where only the nodes belonging to the backbone use energy to forward messages. The goal is then to minimize the number of nodes of the backbones [1]. Because of the need of robustness and scalability of the solution, most algorithms are operating in a distributed manner with local election of the nodes belonging to the backbone [2]. Since this problem is NP-hard, some authors work to find guarantees on the approximation ratio [3].

Unfortunately, with this strategy, the nodes belonging to the backbone will consume more energy. Thus after a certain period of time, the network will be disconnected while the other node may still have a lot of energy. In order to increase the lifetime one can compute several backbones, trying to find a set of CDS such that the maximum number of CDS a node belongs to is minimized. Such a distributed algorithm is proposed in [4]. Nevertheless, this model does not take into account the real consumption of energy of the nodes, which depends (among others) on the number of received messages i.e. on the degree of the node. In [5], the authors dynamically construct backbones taking into account the remaining energy of each node. In our case, all data have to be gathered to a fixed sink. We thus only have to compute a directed in-tree rooted at the sink, and the sink may initiate this computation (the algorithm is not localized but only distributed). This specification allows us to need only a small number of messages to compute a backbone.

III. The Model

A WSN can be modeled by a graph $G=(V, E)$ where V is the set of sensors and an edge $e=(v, w) \in E$ if v and w can communicate. We suppose that if v can communicate with w, the contrary is also (G is not directed). Results on WSN are highly dependant on several parameters of the network (density, model of energy consumption, measurement frequency, etc.). We thus have to make several assumptions in order to specify the global framework of our work.

We suppose that the frequency of the measurements is small enough so that there is enough time to collect data from all sensors to the sink without new measurements being sent. If we compute a routing to collect the data, we then can *aggregate* them at each node. So to do a gathering, we just have to find a directed in-tree rooted at the sink. We also assume that the size of the data is small enough so that even after several aggregations the size of the messages sent will be less than one packet. Thus for each gathering, the emission cost will be the same for all sensor. This is our unit to measure the energy consumption. Note that this assumption is just done to fix the conditions of the experiments: our algorithm still works if the size of the messages (and thus the energy needed to send them) is not constant and the sensors know their remaining energy. The cost for a reception cost will not be supposed to be equal to zero. Although this cost is not often taken into account especially in theoretical models, this seems to be a reasonable assumption regarding for example [6] or [5]. Both emission and reception costs depend on the sensors type and on the transmission range but they usually have the same order of magnitude.

The energy needed for measurement depends on the kind of measurement performed and the device used to do so. Taking into account this energy in the model would make it highly dependent on the application. In order to avoid this dependency, we will suppose that auxiliary batteries provide the energy needed to make the measurements and thus the cost for measurement is null. One can remark that this cost would not be difficult to take into account in our model and since for each gathering, each node will make exactly one emission, we just need to increase the cost of emission of this value for the messages containing data.

IV. Distributed Search of Various CDSs (DSVB)

A. Main Algorithm

The principle of our algorithm is that each node will choose a father in the backbone the first node from which he received an "invitation" message. Since all the nodes but the sink send their invitation after receiving one and waiting a certain time, this ensures that we built a directed in-tree rooted at the sink. More formally, for each search of a backbone *b*, the algorithm works as follows:

```
• The sink $s$ sends invitation <INV> with its id and $b$'s
  id (broadcast to all of its neighbors)
• For all other nodes $v$ do
    o Father ← sender's id in the first <INV> received
    o Send acceptation <ACC> (with $v$'s id, Father's id
      and $b$'s id)
    o Chose a delay $w$
    o Wait $w$
    o Send <INV> (with $v$'s id and $b$'s id)
    o If (number of received <ACC> = number of
      neighbors) and ($v$ is a father in none of the
      received <ACC>) then $v \notin b$
    o If, in a received <ACC>, (Father's id = $v$'s id)
      then $v \in b$
```

For each node, computing a backbone with this algorithm needs only to send two messages and so a number of receptions equal to twice the degree of the node.

B. Computing the Delay

Fine-tuning of the algorithm is done by the computation of the delay each sensor has to wait before sending an invitation. The influence of this delay obviously depends on the time needed for one node to run the algorithm (if the algorithm needs 100 μs to be run, adding a delay of 2 μs will not have much influence). Let *t* be an upper bound of the time a node needs to run the algorithm (receive and process a message, compute the delay and send a message). Our unit to measure the energy of a node will be the amount *e* of energy needed to send or receive a message. Let $e{\cdot}E_i$ be the initial energy of the node (thus E_i is the number of messages a node can send), $e{\cdot}E_r$ its remaining energy after a certain duration of use. This remaining energy may by known by the sensor or evaluated considering the numbers of emissions and receptions already done (in this second case, the battery model is supposed to be linear). Let n_c be the number of CDS already used and n_b the number of CDS a node already belongs to. A "penalty" is computed for each node. It has to increase when:

- The proportion of CDS a node belongs to increases in order to discriminate the nodes even at the beginning of the process. This parameter is especially important for regular graphs.
- The remaining energy of a node decreases.

The penalty is thus computed using formula (1)

$$p = t \times \left(f\left(n_c / n_b ; E_r \right) \right) \tag{1}$$

where the function *f* is an increasing function of its parameters. In order to differentiate nodes that would have the same penalty, each delay is randomly chosen in a range [0, *exp*p*] where *exp* is an expansion factor that increases the differences of penalty ensuring that the delay is not too long. Note that if the delay is constant, then our algorithm is formally identical to a Breadth First Search (BFS).

V. Main Results

In order to validate our approach and be as near as possible of the functioning of a real network, we use WSNET simulator. Despite NS-2 is more often use in the literacy, WSNET is reported to have a more realistic model for transmission [8]. We simulate the effective communications between the nodes with an autonomous functioning of the nodes. The sink is supposed to have an unlimited energy. In our simulations, $t \approx 2$ μs. Those values are chosen considering the devices Micaz of MEMSIC on which we plain to make experiments on in further works.

For the delay, *f* is chosen according to equation (2).

$$f\left(n_c / n_b ; E_r \right) = \left(\frac{\max(n_c;1)}{\max(n_b;1)} \times \frac{E_i}{1+E_r} \right)^k . \tag{2}$$

The value of $f(n_c/n_b;E_r)$ is lower or equal to 1, but usually very smaller than 1. In order to know the highest penalty when a backbone is computed, each node has to transmit the maximal current penalty it knows (from its penalty and its children) when returning a data. The highest penalty p_{max} is then included in the invitation message. The expansion factor is given by $exp = c / p_{max}$ where c is a coefficient depending on the size of the network. The latency to built a backbone is at most |V|**exp*. In our experiments, c is set so that the latency is lower or equal to 10 ms.

A. Networks

Two kinds of networks are used to do the simulations.

1) Grids

In many potential applications, sensors are not randomly spread and the network has a "grid-like" shape (deployment in fields, cities, building, containers on a boat). Furthermore, grids have interesting properties for our studies since they both have low density (which make the computation of disjoint CDS difficult) and relatively high connectivity (which helps to avoid degenerated cases that may occur because of the presence of isthmus or lowly connected parts).

We made simulations on two kinds of grids. The first one $G_R(p, q)$ is the usual $p\times q$ grid. In the second one $G_{R\sqrt{2}}(p, q)$, each node cannot only communicate with its 4 neighbors but is also connected on the diagonal and thus has 8 neighbors. The sink is always the center of the grid. The main part of the simulations are made using $p = q = 11$.

2) Random networks

The networks are unit disk graphs. For those network, $|V|$=100 and the density (average number of neighbors) is 10. We choose the 100 first connected networks generated.

B. Results

1) Setting the parameters

In order to see how the different parts of the delay influence the construction of the CDSs, we try several combinations of the parameters. The best value for k (the exponent in formula 2) is 6.

In TABLE I. we present the number of gatherings that can be collected to the sink using different parameters for the delay computation and for different graphs. In the first line, the penalty is randomly chosen between 0 and 1. In the second line, E_r is set to E_i-1. In the third line, n_c is set to n_b and in the last line, both parameters are taken into account. The values in the 2 first columns are means for 16 runs on the same graph. The values in the last column are means for 16 random graphs with various density and connectivity 2 or 3. A new CDS is computed every 116 gatherings.

TABLE I. NUMBER OF GATHERINGS COLLECTED AT THE SINK (EI=8000)

Penalty	**Network**		
	G_R*(11, 11)*	**$G_{R\sqrt{2}}$*(11, 11)***	***Random graphs***
Random	1578	911	593
Frequency	1849	1358	949
Energy	2059	1615	1182
Both	2215	1735	1159

These results show the interest in using both parameters to compute the delay on regular graphs. For random graphs, taking into account the frequency does not seems to improve the lifetime, but the loss is negligible so both parameters will be used for the next simulations on random networks.

2) Frequency of CDS computation

Figure 1. presents how the initial energy and the frequency of the CDS computations influence the ratio of gatherings achieved (number of gatherings / intial energy). We give average values obtained considering 16 runs on $G_{R\sqrt{2}}(11, 11)$.

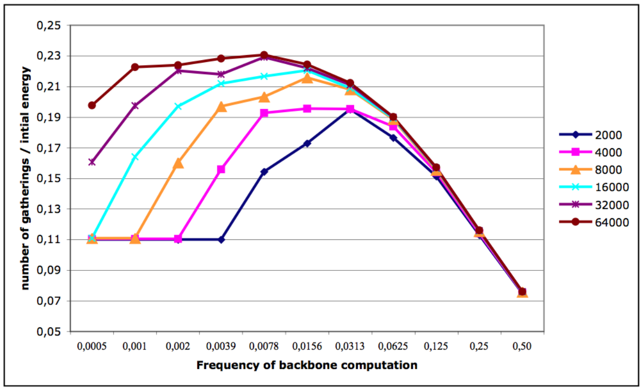


Figure 1. Ratio of gatherings collected at the sink regarding the frequency of backbone computation for different initial energy

As expected, when the number of gatherings per backbone is higher than the number of gathering collected (low frequency), the ratio is constant and corresponds to the use of a single backbone. On the contrary, computing many backbones has a cost, which becomes excessive when the frequency is too high.

3) Efficiency of DSVB

In order to know how our algorithm performs, we compute lower and upper bounds for the number of gathering that may be achieved.

The number of gathering made using a single CDS (for example, found by a BFS) C gives a lower bound. In this case, for a node v in the CDS, the energy used for each gathering is $(d(v)+1).e$ (where $d(v)$ is the degree of v in G): this node receives messages from all its neighbors (cost $d(v).e$) and transmits its aggregated data to its father (cost e). Using this strategy, the maximum number of gatherings allowed is $\min_{v\in C}(E_v(v)/(d(v)+1))$. We thus have to find a CDS C such that $\max_{v\in C}(d(v))$ is minimum.

To compute an upper bound for the number of gatherings we extend the idea proposed in [7] to our model. For each vertex-cut S that disconnect G (for convenience, we will suppose that S does not contain any leaf of G), we need, for each gathering, to use at least one of its vertices. The energy used by a node v for a gathering is $(d(v)+1).e$ if v is in a CDS and e else. So if x_i is the number of CDSs a node i belong to, for each node i we must have $(d(i)+1)x_i + \Sigma_{j\text{-}i}x_j \leq E_i(i)$. The number z of gathering that can be supported by the set S is then: $z = \Sigma_i x_i$. To obtain an upper bound, we can solve the relaxation of this linear program. If every constraint is tight $(x_i = (E_i(i) - z)/d(i)$ it leads to:

$$z = \sum_{i\in S} \frac{E_i}{d(i)\left(1 + \sum_{j\in S}\frac{1}{d(j)}\right)}. \qquad (3)$$

Since it is possible to find a solution of the dual having the same value (set $y_l = 1/(d(i)(1+\Sigma_j(1/d(j)))$ for all l in S), this solution is optimal.

Finding the set S of vertices disconnecting G such that (3) is minimized gives an upper bound for the maximum number of gatherings. For example, for a network $G_R(p, q)$, a set S achieving the minimum number of gatherings is composed of the two neighbors of a corner.

In order to evaluate the efficiency of our algorithm, TABLE II. shows the number of gatherings collected using DSVB compared to a lower and an upper bound (the conditions are the same as for TABLE I.

TABLE II. EFFICIENCY OF DSVB

Network	**Number of gatherings achieved**		
	Lower Bound	***DSVB***	***Upper bound***
$G_R(11, 11)$	1600	2215	2666
$G_{R\sqrt{2}}(11, 11)$	888	1735	2754
Random graphs	554	1182	1763

Although the actual result using a single CDS would be the lower bound, the upper bound may be overestimated, which suggests a better performance of our algorithm.

4) "real" lifetime

Lifetime of WSN is not a well-defined notion [8]. Especially in the case of networks where the measures made by the sensors are redundant, one may accept a reasonable ratio of loss of messages (i.e., lifetime is not equal with the guaranteed message transfer). This logically influences the lifetime of the network. We have seen that in usual grid networks, the main (theoretical) problem occurs for the corners since they are only connected to the rest of the network by 2 nodes of degree 3. Nevertheless, if we accept a loss rate of 4% those nodes are not any more limiting as soon as the grid has more than 100 nodes. Figure 2. shows how the number of transmitted measurements (i.e. connected nodes) decreases regarding the number of gathering for two representatives runs on $G_R(11, 11)$.

We fix a threshold of 90% such that a new backbone is computed either if 100 gatherings are done or if less than 90% of data are collected. For 16 runs, the first failure occurs in mean at the gathering number 2215, whereas we achieve 2341 gathering collecting more than 90% of data. For random networks, the lifetime increases of 10% with a threshold of 95% and 13% with a threshold of 90%.

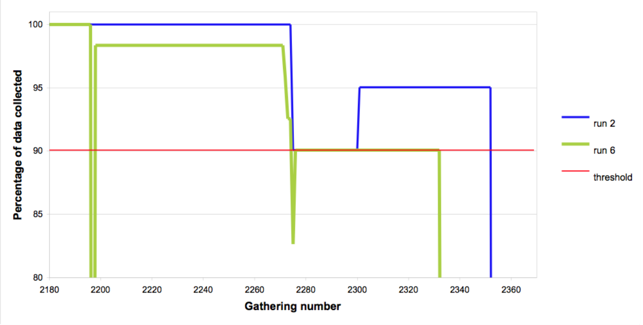


Figure 2. Percentage of data effectively collected regarding the number of gatherings on $G_R(11, 11)$

VI. CONCLUSION

In this paper, we presented a distributed algorithm to collect data in a Wireless Sensor Network. Easy to implement, it computes several Connected Dominating Sets sharing out the use of the sensors for the transmissions. Simulations have shown a significant improve of the network's lifetime. One of the next steps is to validate our approach on real sensors. In several applications, measures are redundant and some algorithms exist to optimize the energy used for measuring. In a future work, we will try to combine our technique with an efficient k-coverage of the network. We aim to save energy globally for both measuring and communicating.

ACKNOWLEDGMENT

This work was supported in part by the Hungarian–French Intergovernmental S&T Cooperation Program under the "Dependable Services in Sensor Networks" (F-6/2008) project.

REFERENCES

[1] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets," *Algorithmica*, vol. 20, pp. 374–387, Apr. 1998.

[2] C. Adjih, P. Jacquet, and L. Viennot, "Computing Connected Dominated Sets with Multipoint Relays," *Adhoc & Sensor Wireless Networks*, vol. 1, n°. 1, pp. 27-39.

[3] K. Islam, S. G. Akl, and H. Meijer, "A Constant Factor Localized Algorithm for Computing Connected Dominating Sets in Wireless Sensor Networks," in *International Conference on Parallel and Distributed Systems*, pp. 559-566, 2008.

[4] K. Islam, S. Akl, and H. Meijer, "Distributed Generation of a Family of Connected Dominating Sets in Wireless Sensor Networks," in *Distributed Computing in Sensor Systems*, vol. 5516, B. Krishnamachari, S. Suri, W. Heinzelman, and U. Mitra, Ed. Springer Berlin / Heidelberg, 2009, pp. 343-355.

[5] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, pp. 481–494, Sep. 2002.

[6] T. Val, A. Van Den Bossche, and E. Campo, "Technologie ZigBee / 802.15.4 - Protocoles, topologies et domaines d'application," *Techniques de l'ingénieur*, vol. 7508, May. 2008.

[7] E. Ben Hamida, G. Chelius, and J. M. Gorce, "Impact of the Physical Layer Modeling on the Accuracy and Scalability of Wireless Network Simulation," *SIMULATION*, vol. 85, n°. 9, pp. 574 -588, 2009.

[8] J. Champ, C. Saad, and A. Baert, "Lifetime in Wireless Sensor Networks," presented at the CISIS : International Conference on Complex, Intelligent, and Software Intensive Systems, Fukuoka, Japan, 2009.

A Mobile Beacon Based Localization Approach for Wireless Sensor Network Applications

Yifeng Zhou and Louise Lamont
Communications Research Centre Canada
Email: yifeng.zhou, louise.lamont@crc.gc.ca

***Abstract*—In this paper, a novel localization approach based on the use of a mobile beacon is proposed for wireless sensor networks. The localization system consists of a mobile beacon and beacon receiving modules on each sensor node for measuring distance. The localization is based on the application of multidimensional scaling technique and a local map registration approach. The approach is designed to operate without requiring path planning for the mobile beacon. It estimates the relative coordinates of the sensor nodes, and does not require the location information of the mobile beacon, making them attractive for applications where access to GPS signals is not available. Finally, computer simulations are used to evaluate the localization performance of the proposed approaches under different scenarios.**

***Index Terms*—wireless sensor network, sensor localization, mobile beacon, multilateration, MDS, map registration, path planning, GPS, least squares**

I. INTRODUCTION

Sensor node localization is a highly desirable capability for wireless sensor network applications. Localization refers to the process of estimating the coordinates of the sensor nodes in a network based on various types of measurements and with the aid of a number of beacon or anchor nodes that know their locations. A beacon node broadcasts beacon signals with limited information content. Anchors are required for sensor localization in a global coordinate system. The location information of an anchor or beacon node can be hard-coded or acquired by using localization systems such as a Global Positioning System (GPS) receiver. There are a number of reasons why sensor localization is important. For example, sensor location information can be used for tagging sensory data, which is important for environmental monitoring and military surveillance applications. The operation of a sensor network relies on sensor location information for uncovering and healing coverage holes in the network. Sensor location information can also be used to perform efficient spatial querying or tasking, *e.g.*, scoping the query or task propagation to sensor nodes in specific locations or geographic regions without the need to flood the whole network, significantly reducing the network overhead and minimize consumption of energy and resources in the network.

Recently, many sensor localization techniques have been developed for wireless sensor network applications [1][2]. In this paper, we focus on mobile beacon-based sensor localization approaches. Localization using mobile beacons has many advantages over those that use static beacons. The use of mobile beacons pushes the hardware complexity and power consumption requirement on the mobile beacon, which is less resource constrained and has access to the required power for repetitive message transmission to sensor nodes to be localized. In addition, the use of mobile beacons can significantly reduce the cost of sensor deployment. A mobile beacon transmitting at different locations can be considered equivalent of multiple static beacon deployment. Mobile beacons can move and easily avoid environmental obstructions. Using mobile beacons can also avoid the problem of interference and collision of beacon signals due to uncoordinated beacon transmissions of static beacons. In [3], Sichitiu *et al.* proposed a mobile beacon based localization method, in which the received signal strength indicator (RSSI) was used for ranging. A mobile beacon traverses the deployment area while broadcasting beacon signals. Sensor nodes that receive beacon signals infer proximity constraints to the mobile beacon, and their positions are estimated using a Bayesian approach. In [4], a solution called the Walking GPS was proposed. In this approach, a mote equipped with a GPS receiver (mobile beacon) is carried by a sensor deployment person and periodically broadcasts its location. A sensor node being deployed infers its position from the location broadcast by the GPS mote. This approach is simple and cost effective. Its disadvantage is also obvious: the localization results are directly determined by GPS accuracy. Galstyan *et al.* [5] proposed a distributed online localization algorithm based on a moving beacon, in which sensor nodes use geometric constraints induced by both radio connectivity and sensing to reduce the uncertainty of their positions. The authors then generalized the approach to use a moving target with *a priori* unknown coordinates. In [6], a refined approach is proposed, which uses mobile anchor scenarios for anchor information distribution. Statistical techniques are adopted for localization with inaccurate range data. In [7], a walking beacon-assisted localization is discussed and two distributed localization methods are proposed where sensor nodes compute their position estimates based on the range-free technique. The first method uses the arrival and departure information of a walking beacon and the second method exploits the variance of the RSS measurements from the beacon.

In this paper, we propose a novel localization approach, referred to as the MAP approach, based on the use of a mobile beacon for wireless sensor network applications. The localization hardware includes a mobile beacon and a beacon

ISBN: 978-1-61208-144-1

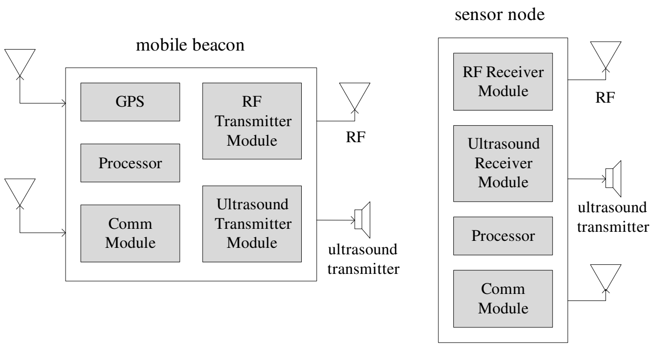


Fig. 1. Block diagram of the mobile beacon and sensor node.

receiving module on each sensor node. The mobile beacon moves in the sensor deployment area while broadcasting beacon signals to the network of sensor nodes. When a sensor node receives a beacon signal, it can estimate its distance to the mobile beacon at the time of beacon transmission. In the proposed approach, the user carries the mobile beacon and deploys sensor nodes. When a sensor node is deployed, the user turns on its power and set the mobile beacon to transmit a set of beacon signals. Any previously deployed nodes that are within the transmission range of the mobile beacon will receive the beacon signals and estimate their distances to the sensor node being placed. The inter-node distance measurements are obtained by using the mobile beacon, and the mobile beacon does not need a pre-planned path. All sensor nodes then pass the data back to a central node for localization using a map registration algorithm developed by Zhou *et al.* [10][11]. The rest of the paper is organized as follows. In Section II, the mobile beacon system is discussed with a detailed description of the hardware systems on both the mobile beacon and the sensor nodes. The ranging mechanism of the mobile beacon is also discussed. The proposed MAP localization approach is also presented in Section III. In Section IV, computer simulations are used to demonstrate and compare the performance of the proposed mobile beacon based localization approaches. Finally, the simulation results are analyzed and conclusions are presented.

II. System Hardware and Ranging

Fig. 1 is the the block diagram for the mobile beacon and sensor node. The mobile beacon consists of a GPS receiver and an RF/ultrasound transmitter in addition to a processor and a communication module. All sensor nodes in the network are equipped with an RF/ultrasound receiver module. Note that the GPS receiver is required for the mobile beacon when global coordinates of sensor nodes are required. A beacon signal contains an RF message followed by an ultrasound pulse that is synchronized in time. The RF message contains the beacon sequence number, time stamp and the current location of the mobile beacon (optional and obtained through the onboard GPS receiver).

When a sensor node receives a beacon signal, it can estimate its distance to the mobile location at the time of beacon transmission based on the TDOA between the RF signal and the ultrasound pulses. There are different ranging techniques including those based on time-of-flight (TOF), TDOA, and RSSI *etc.* [2]. They typically use either RF, acoustic or optical signals. The extreme fast propagation speed of RF signals makes them impractical for TOF ranging due to the tight time synchronization requirements for sensor nodes that often operate at a low clock frequency. The relatively inexpensive and simple RSSI based ranging tends to be highly susceptible to environmental interference and is known to be unpredictable for distance estimation. The ranging approach used in this study is similar to the one used by the Cricket system [12]. The underlying principle of RF/ultrasound ranging is to use their different propagation speeds in the air. The fact that ultrasound waves propagate at a much slower speed than RF in the air makes it possible for low cost implementation of accurate ranging. The TOA of the RF signal is used as a reference assuming that it is negligible. Then, the TDOA between the RF and the ultrasound represents the TOF of the ultrasound signal traveling from the mobile becon to the sensor node. Ranging based on RF and ultrasound signals can achieve an accuracy of a few centimeters over a short distance [12][13]. It is also able to eliminate the requirement for tight time synchronization between the mobile beacon and deployed nodes. However, it should be noted that since the speed of ultrasound in air varies with ambient temperature, humidity and atmospheric pressure, the impact of these factors on the speed of ultrasound should be accounted for in practice. This is typically done by installing temperature and humidity sensors on the sensor nodes.

III. Mobile Beacon Based Localization

The proposed localization approach starts from the stage of sensor node deployment. After the sensor deployment planning process, the user that carries the mobile beacon starts to deploy sensor nodes. When a sensor node is placed, the user turns its power on, and sets the mobile beacon to transmit a set of beacon signals. Any previously deployed nodes that are within the transmission range of the mobile beacon will receive the beacon signals and estimate their distances to the current mobile beacon location (or the location of the sensor node under deployment). The user then moves to deploy the rest of the sensor nodes and repeats the same procedures until all sensor nodes are deployed. After all sensor nodes are deployed, each sensor node will have a set of distance measurements to its neighbors, which are then passed to a central node for localization using the MAP localization algorithm.

The MAP localization algorithm is proposed to overcome the problem of mismatching of the shortest distances in the MDS method. The MDS localization method requires that the full Euclidean matrix be known. In practice, due to power constraint, the mobile beacon may have a limited transmission range. When two nodes are out of the transmission range of the mobile beacon, the distance measurement between them becomes unavailable and needs to be estimated or approximated

ISBN: 978-1-61208-144-1

[8]. A common approach is to replace the unavailable inter-node distances by their shortest path distances. In a network of regular topology, a shortest path distance is found to match its corresponding Euclidean distance well. However, in a sparse network or a network of irregular topology, a shortest path distance may not match its Euclidean distance, and the use of the approximated distance matrix will result in localization errors [10][8]. In MAP, the network is divided into many small sub-groups of nodes, where adjacent groups share common sensor nodes. A commonly used approach is to form a sub-group for each sensor node, which involves the node and its neighbors with a given number of hops (*e.g.*, one or two hops). One hop is determined by the transmission range of the mobile beacon rather than the radio range of the sensor nodes. A local map is built for each sub-group of sensor nodes using the MDS method [8]. The local maps are then merged into a global map based on the common sensor nodes shared by different groups. In [8], an incremental greedy algorithm was proposed for merging the local maps in a sequential manner. One local map is randomly selected as the core map, which is grown by merging the local maps one by one. During the merging process, an optimal rigid transformation is determined, which minimizes the conformation difference between the locations of the common nodes in the core map and those of the local map subject to the rigid transformation. The incremental greedy algorithm is seen to be locally optimal since it only explores the commonalities of the shared sensor nodes in two maps. In practice, the common sensor nodes are often shared by more than two local maps. The sequential merging process can also lead to error propagation and perhaps unacceptable errors as the network grows. In [10], the MAP approach was introduced to counter the problems of the incremental greedy approach. In MAP, instead of using a sequential pairwise approach for merging local maps, the construction of the global map is considered at a global level. An affine transformation is defined for each local map. The set of optimal affine transformations are obtained simultaneously by considering all available nodes that are shared by various local maps. The set of optimal affine transformations are found, which minimize the location discrepancies of sensor nodes subject to their corresponding affine transforms in the global map. The discrepancy is represented by the sum of the squared distances of all nodes to their respective geometric centers in the global map. The resulting coordinates of the sensor nodes in the global map are relative coordinates. If desired, they can be transformed into their global coordinates based on the use of a few selected beacon nodes.

Since the proposed local map registration algorithm minimizes the overall discrepancies of the locations of all sensor nodes, it is able to counter the problems associated with approaches based on pairwise map merging, and achieve the global optimal performance. The problem of finding the optimal rigid transformation for two maps based on common nodes has closed-form solutions [14]. The approach by Arun *et al.* [15] is shown to have provable optimality and the advantage of computational efficiency over other methods. Arun's approach minimizes the squares error between two sets of matched points under rotation and translation, and the optimal transformation is obtained using a singular value decomposition (SVD). The problem of finding a set of optimal transforms for multiple local maps, however, is not trivial, and analytic solutions do not exist due to the highly nonlinear optimization criterion involved. In this study, a gradient projection algorithm is developed for finding the optimal transforms for transforming local maps to a global map [10]. The algorithm is developed based on the general idea by Jennrich in [16][17] and is suitable to the constrained optimization problem of coordinate transformation. In spite of the iterative nature of the algorithm, it has faster convergence and is computationally more efficient than many general numerical optimization techniques [19][18] for nonlinear programming.

The proposed localization approach does not rely on the knowledge of the mobile beacon locations, which is important for applications where access to GPS satellites is not available. However, it is necessary to point out that the localization results from MAP are relative sensor locations, *i.e.*, the estimated sensor locations are given in an arbitrary coordinate system. In many applications, relative location information is sufficient. For example, in applications such as detecting and tracking an intruder, the user is concerned about the location of a target relative to the network rather than its global coordinates. Relative locations of the sensor nodes will suffice for wireless network functions such as routing for communications and tasking. If global coordinates of the sensor nodes are desired, then, a number of anchors are needed to determine a rigid transformation of the relative coordinates into global coordinates.

IV. Computer Simulations and Performance Analysis

In this section, we use computer simulations to demonstrate the performance of the proposed mobile beacon based localization techniques. Four types of network shapes and sensor deployment scenarios are used in the simulation: rectangular random network, rectangular grid network, C-shape random network, and C-shape grid network. A C-shape area is defined as a rectangle that contains a concave on one side. In this study, the concave is located at the center of the rectangle's bottom line.

The mobile beacon is simulated to have a transmission range of 25 meters. The user moves at a speed of 0.694 meters per second and spends 4 seconds to place a sensor node. When in periodic broadcasting mode, the mobile beacon broadcasts beacons to the network regularly every 10 seconds. For a grid network, the deploying person starts from the left-most column of the sensors, and places the sensor nodes from bottom to top. The next column of sensor nodes are placed from top to bottom along the y-axis. This process continues until all sensor nodes are placed. For a random network, the sensor deployment area is divided into multiple segment of equal length on the x-axis. In the first (left-most) segment, sensor nodes are placed from bottom to top along the y-axis. In the

ISBN: 978-1-61208-144-1

next segment, the sensor nodes are placed from top to bottom along the negative direction of the y-axis. This process repeats itself until all sensor nodes are placed.

The distance estimates computed by a sensor node from receiving the mobile beacon signals is assumed to contain additive errors, which are modeled as a random variable that follows a uniform distribution with boundaries (both positively and negatively) proportional to the actual distance. For an actual distance d between the mobile beacon and sensor node, the distance estimate error is uniformly distributed in the interval $[-\kappa d, \kappa d]$, where κ is the proportionality constant. The mobile beacon is assumed to carry a GPS receiver to acquire its own coordinates (this information is only needed for multilateration based approaches). The mobile beacon location errors are simulated to be additive Gaussian distributed with zero mean in both x and y coordinates. The errors in x and y coordinates are assumed to be statistically independent and have a same standard deviation of $1/\sqrt{2}$ meters. For each scenario, we vary the constant of proportionality κ and use Monte-Carlo simulations to compute the root mean squares errors (RMSE) of the sensor location estimates. In the simulation, κ varies from 0 to 0.1 with a step size of 0.01. For each value of κ, 100 tests are repeated to obtain the RMSEs of the location estimate of each sensor node. An averaged RMSE is then computed by averaging the RMSEs of all nodes.

Five other localization approaches, referred to as MLE, LLS, MDS PATH-MLE and PATH-LLS, respectively, are included for comparisons. MLE and LLS are based on the use of multilateration. The user carries the mobile beacon and deploys sensor nodes. The mobile beacon broadcasts beacon signals periodically. If a sensor node receives beacon signals from more than three locations, it can apply multilateration to find its coordinates. The only difference between MLE and LLS is that MLE uses a nonlinear least squares formulation of multilateration while LLS is formulated as a linear solution. MDS uses the same deployment strategy as the proposed MAP approach. PATH-MLE and PATH-LLS are the nonlinear and linear least squares solutions of multilateration, respectively. They differ from MLE and LLS in that the mobile beacon moves along a planned path around or in the sensor deployment area. In this study, simple paths are used in evaluating PATH-MLE and PATH-LLS, which are along the perimeter of the sensor deployment area.

A. Rectangular random network

30 sensor nodes are deployed in a rectangular area of 100 meters by 20 meters. The mobile beacon moves on a Z-shape path in the deployment area. Four anchor nodes are selected. The anchor nodes are required by MAP and MDS to transform the resulting relative coordinates of the sensor nodes into global coordinates for comparison. For PATH-MLE and PATH-LLS, the mobile beacon moves on a Z-shape path in the deployment area.

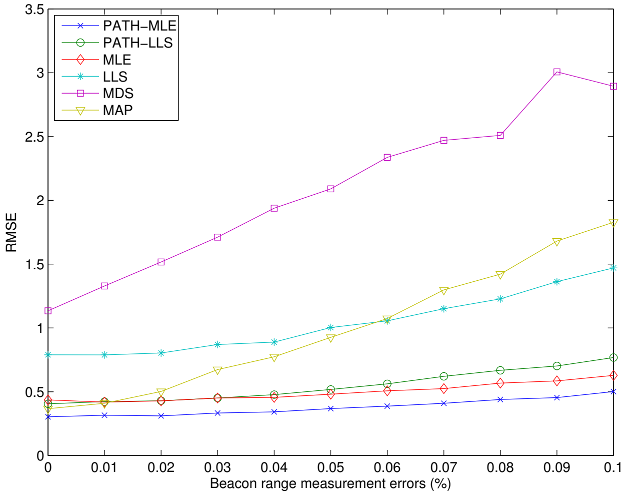


Fig. 2. RMSEs of location estimates via κ: rectangular random network.

Fig. 2 shows the RMSEs of the location estimates versus κ. Among all the approaches, the MDS approach performs the worst in terms of RMSEs for all κ. Even when $\kappa = 0$, *i.e.*, there are no ranging errors at all, MDS still shows an RMSE of 1 meter. This phenomenon is due to the use of approximated distances in the Euclidean matrix in MDS. As discussed before, due to the limited transmission of the mobile beacon, all inter-node distance estimates are not available. For node pairs that have separation distances greater than the mobile beacon transmission range, the distance estimates will be approximated by their shortest path distances. As the number of unavailable distance estimates increases, the localization performance deteriorates. We use connectivity level to characterize the availability of inter-node distances, which is defined as the averaged number of nodes that a node can receive mobile beacon signals from. Note that this connectivity is defined based on the mobile beacon transmission range instead of the radio range of the sensor nodes. In Fig. 2, the connectivity level is computed as 11.2, which means that each node can directly measure its distances to about 11 nodes instead of the 29 nodes in the ideal case. In general, the connectivity level increases with node density of a network and the mobile beacon transmission range. PATH-MLE, Path-LLS and MLE outperform the others. The LLS approach outperforms MAP only for low values of κ ($\kappa < 0.06$). It is interesting to note that, for relatively small κ ($\kappa < 0.06$), the RMSEs for PATH-MLE, PATH-LLS, MLE, LLS, and MAP, are all smaller than 1 meter, which is the simulated RMSE for GPS location errors. This indicates that these approaches are able to suppress errors in the mobile beacon locations, and provide better localization performance than by using GPS alone.

B. Rectangular grid network

In this scenario, 30 sensor nodes are placed on a rectangular grid with 20% placement errors. The unit length of the grid is $r = 8$ meters. The placement errors are simulated as additive and uniformly distributed in the interval $[-0.2r, 0.2r]$ in both the x and y coordinates of the node's original grid position. For PATH-MLE and PATH-LLS, the mobile beacon moves on

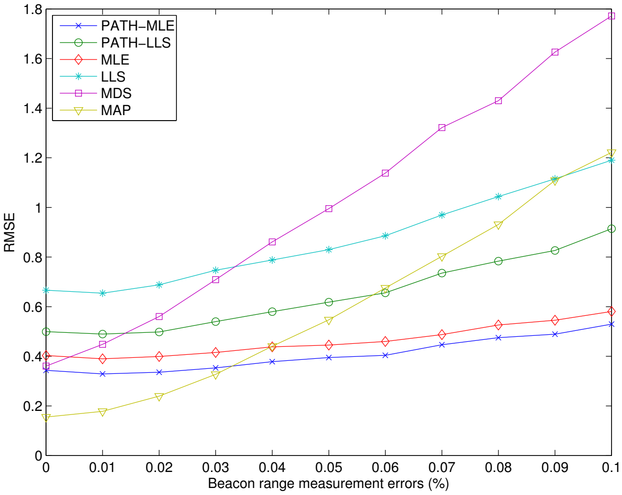


Fig. 3. RMSEs of location estimates via κ: rectangular grid network.

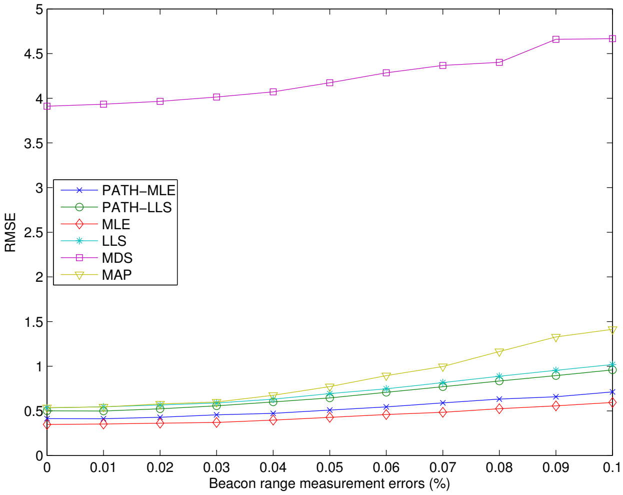


Fig. 4. RMSEs of location estimates via κ: C-shape random network.

a Z-shape path in the deployment area. Four anchor nodes are selected.

Fig. 3 shows the RMSEs of the location estimates versus the beacon ranging errors (κ). The MAP approach performs better than PATH-LLS and LLS when $\kappa < 0.06$. The performance of MDS lags behind the other approaches. All the approaches have better localization performance in the rectangular grid network than in the rectangular random network. In particular, MDS sees significant improvement in the rectangular grid network, which may partially be attributed to the increased connectivity level of the grid network and its relatively uniform node density. The simulated grid network has a connectivity level of 14.1. For a rectangular grid network, the shortest path between a pair of nodes corresponds well with their Euclidean distance.

C. *C-shape random network*

The C-shape random network is simulated by randomly placing 45 sensor nodes in a C-shape area. The rectangle size is 100 meters by 40 meters, and the concave size is 60 meters by 20 meters. The placement of the sensor nodes follows a uniform distribution. Five anchor nodes are selected. For PATH-MLE and PATH-LLS, the mobile beacon moves along the perimeter of the C-shape area.

Fig. 4 shows the RMSEs of location estimates versus κ. PATH-MLE and MLE have similar performance and perform best among all approaches for all κ. PATH-LLS and LLS are close in their RMSEs for all κ, and they are slightly outperformed by PATH-MLE and MLE. The MAP approach performs reasonably well and has RMSE values that are less than 1 meter when $\kappa < 0.07$. On the other hand, the MDS performs poorly and fails to provide satisfactory localization performance. The RMSE values for MDS are larger than 3.5 meters for all κ. The failure may be due to the irregular shape of the C-shape network as well as the low connectivity level of the network. As discussed in [10], for a sensor network of irregular shape, the shortest paths between pairs of sensor nodes usually do not correlate well with their Euclidean distances. The simulated C-shape random network has a connectivity level of 13.5.

D. *C-shape grid network*

In the simulation of the C-shape grid network, 38 sensor nodes are placed on a C-shape grid with 20% placement errors. The gird has 10 sensor nodes in the x direction and 5 sensor nodes in the y direction. The concave contains 4 and 2 sensor nodes in the x and y directions, respectively. The unit length of the grid is 8 meters. Five anchor nodes are selected. The anchor nodes are selected by dividing the sensor deployment area into five sub-areas and randomly selecting one sensor node from each area. The mobile beacon moves along the perimeter of the C-shape area.

Fig. 5 shows the RMSEs of location estimates versus κ. The RMSE curves for the C-shape grid network are similar to those for the C-shape random network. The RMSEs for all six approaches increase as κ increases. PATH-MLE and MLE perform best and have close RMSE for all κ. PATH-LLS and LLS are slightly outperformed by PATH-MLE and MLE, but have close RMSE values for all κ. The MAP approach outperforms PATH-LLS and LLS for $\kappa < 0.4$. The MDS approach fails to produce satisfactory results with RMSE values larger than 3 meters for all κ. Note that MDS has slightly smaller RMSE values than in the C-shape random network. As the simulated C-shape grid network has a connectivity level of 16, which is larger than the connectivity level for the C-shape random network, this may explain the improved RMSEs of the sensor location estimates for the MDS approach.

V. CONCLUSIONS

In this paper, the MAP localization approach based on the use of a mobile beacon has been presented for wireless sensor network applications. The use of a mobile beacon has the advantage of flexibility and can greatly simplify the process of sensor localization. In addition, it is able to overcome many of

ISBN: 978-1-61208-144-1

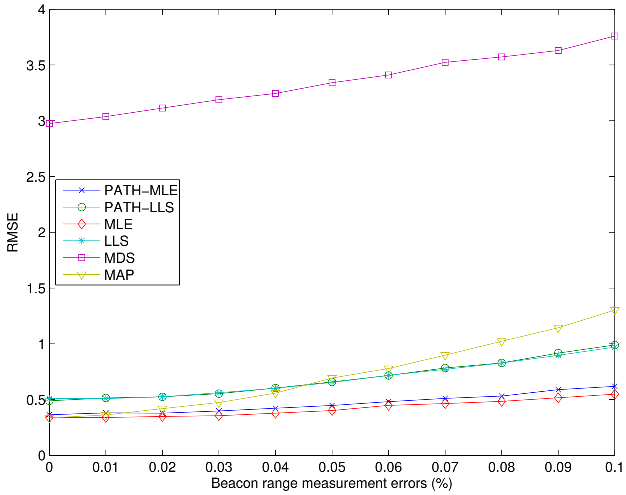


Fig. 5. The RMSE of location estimates versus κ: C-shape grid network.

the difficulties that would be encountered by using static beacons or localization approaches based on inter-node ranging. The performance of the MAP approaches was evaluated using computer simulations and compared with other approaches that are also based on the use of mobile beacons. Four types of network topology and sensor node placement were simulated. The simulation results show that the MAP approach has significant improvement in terms of RMSEs of sensor location estimates in comparison with the MDS approach. It outperforms or performs as well as LLS for low κ values. In the case of rectangular grid network, MAP outperforms LLS for all κ. It is observed that MAP performs better for grid networks than for random networks partly due to the relatively large and more balanced connectivity levels of grid networks. In general, MAP is a practical sensor localization techniques that can provide satisfactory localization results. Although the simulation results showed that maximum likelihood based approaches (*e.g.*, MLE and PATH-MLE) are able to provide the best localization performance, they are not considered practical due to the nonlinear optimization procedures. MAP is a GPS-less approach, *i.e.*, it does not need to know its own location coordinates. However, if global coordinates of the sensor nodes are desired, then a sufficient number of anchor nodes with known global coordinates is required.

ACKNOWLEDGMENT

The work described herein was funded by Defence Research and Development Canada (DRDC).

REFERENCES

[1] J. Bachrach and C. Taylor, *Localization in Sensor Networks*, Chapt. 2, *Handbook of Sensor Networks: Algorithms and Architectures*, *ed.* I. Stojmenović, Wiley Interscience, 2005.

[2] G. Mao, B. Fidan and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, issue 10, pp. 2529-2553, July 2007.

[3] L. Sichitiu and V. Ramadurai, "Localization of wireless sensor networks with a mobile beacon," *Proc. the First IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004)*, pp. 174-183, Fort Lauderdale, FL, USA, October 2004.

[4] R. Stoleru, T. He, and J. A. Stankovic, "Walking GPS: a practical solution for localization in manually deployed wireless sensor networks," *Proc. the First IEEE Workshop on Embedded Networked Sensors (EmNetS-I)*, pp. 480-489, Tampa, FL, USA, November 2004.

[5] A. Galstyan1, B. Krishnamachari, K. Lerman, and S. Pattem, "Distributed online localization in sensor networks using a moving target," *Proc. the 3rd International Symposium on Information Processing in Sensor Networks (IPSN 2003)*, pp. 61-70, Berkeley, California, USA, April 2003.

[6] T. Parker and K. Langendoen, "Localisation in mobile anchor networks," Delft University of Technology Parallel and Distributed Systems Report Series, PDS-2005-001, Delft University of Technology, Delft, Netherlands, February 2005.

[7] B. Xiao, H. Chen, and S. Zhou, "Distributed localization using a moving beacon in wireless sensor networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 5, pp. 587-600, May 2008.

[8] Y. Shang and W. Ruml, "Improved MDS-based localization," *Proc. IEEE INFOCOM 2004, The 23rd Annu. Joint Conf. of the IEEE Computer and Communications Societies*, Hong Kong, China, March 2004.

[9] Y. Shang, W. Ruml, Y, Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Trans. Parallel Distri. Sys.*, vol. 15, no. 11, pp. 961-974, November 2004.

[10] Y. Zhou and L. Lamont, "An optimal local map registration technique for wireless sensor network localization problems," *Proc.* 11*th International Conference on Information Fusion (FUSION08)*, Cologne, Germany, June 30-July 03, 2008.

[11] Y. Zhou and L. Lamont, "Optimal local map registration technique for wireless sensor network localization problems," *in Advances in Wireless Sensors and Sensors Networks*, *ed.* S. C. Mukhopadhyay and H. Leung, *Springer Lecture Notes in Electrical Engineering*, 2010.

[12] N. B. Priyantha, *The Cricket Indoor Location Systems*, PhD Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2005.

[13] A. Savvides, C. C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad hoc networks of sensors," *Proc. the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'01)*, pp. 166-179, Rome, Italy, July, 2001.

[14] D. W. Eggert, A. Lorusso, and R. B. Fisher, "Estimating 3-D rigid body transformations: a comparison of four major algorithms," *Machine Vision and Applications*, vol. 9, pp. 272-290, 1997.

[15] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d points sets," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 698-700, September 1987.

[16] R. I. Jennrich, "A simple general procedure for orthogonal rotation," *Psychometrika*, vol. 66, no. 2, pp. 289-306, June 2001.

[17] R. I. Jennrich, "A simple general method for oblique rotation," *Psychometrika*, vol. 67, no. 1, pp. 7-19, March 2002.

[18] D. P. Bertsekas, *Nonlinear programming*, Athena Scientific, *2nd ed.*, 1999.

[19] J. E. Dennisand and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Englewood Cliffs, NJ: Prentice-Hall, 1983.

ISBN: 978-1-61208-144-1

Rate Adaptation for Slepian-Wolf Coding in Presence of Uncertain Side Information

Reza Parseh and Farshad Lahouti
Wireless Multimedia Communications Laboratory
School of Electrical and Computer Engineering
College of Engineering, University of Tehran
{*rparseh,lahouti*}*@ut.ac.ir,* URL: *http://wmc.ut.ac.ir*

***Abstract*—In this paper, a rate adaptation framework is proposed to address the problem of Slepian-Wolf coding with uncertain side information at the encoder. The uncertainty arises due to the time-varying nature of the correlation between source and side information in settings such as wireless sensor networks and distributed video coding. The proposed framework is set up based on a multi-mode Slepian-Wolf coding scheme which is designed to minimize the average rate. The presented solution utilizes the feedback channel judiciously to select the best encoder mode and substantially reduces the delay and decoding complexity compared to the previous methods which rely on frequent retransmissions for successful decoding. The designs based on both practical and ideal Slepian-Wolf codes are considered, where the latter serves as the corresponding theoretical performance bound. Simulation results based on LDPC codes show that by using sufficient number of modes, a desirably small average rate gap from the theoretical bound with no uncertainty can be achieved.**

Keywords-Slepian-Wolf coding; rate adaptation; uncertain side information; rate-limited feedback channel.

I. Introduction

Distributed Source Coding (DSC) and especially Slepian-Wolf (SW) coding has been the subject of substantial research interest recently. This is mainly due to its applications in data compression for wireless sensor networks and distributed video coding.

The SW coding theorem as introduced in [1], states that the ultimate lossless compression rate for a source with a given correlated Side Information (SI) only available to the decoder, is the same as that when SI is also available to the encoder. Practical coding schemes appeared later and may be categorized as parity-based and syndrome-based coding approaches [2][3]. These schemes are constructed based on capacity approaching codes such as LDPC and Turbo codes [4][5]. As indicated in [1], the joint Probability Distribution Function (PDF) of source and SI must be available to the encoder for compression based on SW coding.

In many practical scenarios, the joint PDF of source and SI may not be available perfectly to the encoder. For instance, in distributed video coding [6] for wireless video sensor networks, correlation model cannot be estimated at the encoder due to its associated computational cost for the encoder, usually being a mobile device. Also, in wireless sensor networks for environmental monitoring, the correlation varies in time as a result of natural phenomena.

Simple feedback schemes are suggested to overcome this uncertainty problem in the joint PDF of source and SI. For instance, for distributed video coding based on Turbo or LDPC codes, decoding failure is first detected. Next, with requested additional syndrome or parity bits via a feedback channel, subsequent decodings are performed until a probability of error constraint is satisfied [6]. The aforementioned procedure has two major drawbacks; (1) the delay due to the number retransmissions and (2) the computational cost of multiple decodings each time more syndrome or parity bits are fed to the decoder. Different approaches have been proposed to tackle the inefficient use of feedback usually resulting in increasing encoder complexity. Recently in [7], the relay nodes in a wireless video sensor network are utilized to reduce the said retransmission delay by incorporating network coding. However, the delay is still non-negligible.

For flexible rate SW compression, [8] and [9] provide methods to construct multi-rate LDPC and serial and parallel concatenated convolutional codes from a parent code to efficiently handle possibly varying correlation of source and SI. However, these works are focused on code design and either do not present a code selection mechanism or rely on simple ACK/NACK feedback schemes discussed before.

The rate-distortion performance bounds for Gaussian Wyner-Ziv (WZ) coding (lossy DSC) with uncertain SI at the encoder is studied in [10]. In [11], for the case when the SI quality has two different states, unknown to both encoder and decoder, a two-layer WZ coding scheme using transform coding and ideal SW coding is presented.

In this work, we propose a framework to address the problem of SW coding with uncertain SI at the encoder. The framework consists of a multi-mode encoder working with a carefully designed feedback from the decoder. The proposed scheme utilizes the feedback channel effectively and consequently performs a one-time-only decoding for each data frame to substantially reduce decoding delay and computational cost of multiple decodings. It is assumed that the joint PDF of source and SI is fixed over each source frame but varies from frame to frame. In line with [10], we assume that this PDF is controlled by a single parameter σ, which is known to the decoder. We partition the range of σ and assign each interval to a unique mode of the encoder. For each frame, the decoder sends the mode index to the encoder

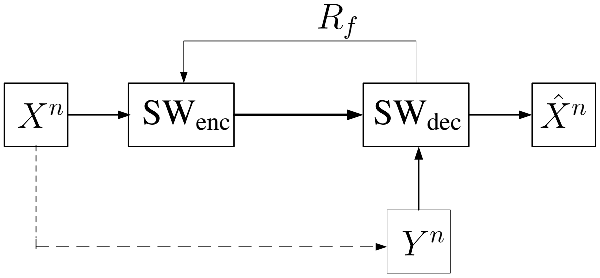


Figure 1. Rate-adaptive SW coding with a rate-limited feedback channel

using the feedback channel. To derive the corresponding performance bounds, we consider ideal (lossless) SW codes and obtain the optimized partitioning and code rates such that the average rate is minimized. Next considering practical SW codes with finite block length and discrete rates, we present an algorithm for the design of the optimized partitioning to minimize the average rate while a given probability of error constraint is satisfied. Simulation results based on LDPC SW codes using syndromes are provided which quantify the performance for different number of modes and demonstrate the effectiveness of the proposed algorithms and designs.

The rest of this paper is organized as follows. Section II discusses the preliminaries on SW coding and introduces the system model. In Section III, performance bounds for SW coding with uncertain SI are derived and in Section IV, effective solutions based on practical codes are presented. Section V is dedicated to simulations and numerical results and finally Section VI concludes the paper.

II. Preliminaries

In this section, we first briefly discuss the original SW coding theorem. We then present the proposed system model and introduce the parameters used in the rest of the paper.

A. Slepian-Wolf Coding and Related Limitations

Consider the i.i.d sequence $(X^n, Y^n) = \{x_i, y_i\}_{i=0}^{\infty}$ with joint probability distribution $P_{xy}(x,y) = P_x(x)P_{y|x}(y|x)$. It may be assumed that X^n and Y^n are correlated via a virtual innovation channel model $P_{y|x}(y|x)$. It was shown in [1] that the two sources X^n and Y^n may be separately encoded and jointly and losslessly decoded at a minimum sum rate of $H(X,Y)$, where $H(.)$ is the Shannon entropy function. This is referred to as distributed source coding in symmetric setting. Interestingly, the said rate coincides with that when both sources are collocated.

If Y^n is available to the decoder as SI, X^n can be encoded at the rate of $H(X|Y)$. This scenario is known as distributed source coding in asymmetric setting. The situation in which X^n is a discrete random variable but Y^n has a continuous alphabet, usually arises in compression of data in sensor networks [12] or compression of quantized indexes in WZ coding of correlated data [13]. A good model for these scenarios is that X^n is a sequence of binary uniform random variables, i.e., $\mathcal{X} = \{-1, 1\}$ and $P_{y|x}(y|x)$ represents an AWGN channel. If the variance of the AWGN channel is equal to σ^2, then the necessary rate for compression of X^n as given in [1] equals $R_{\mathrm{SW}} = H(X|Y)$. This rate can be calculated as

$$R_{\mathrm{SW}} = H(X|Y) = H(X) + \frac{1}{2}\log 2\pi e\sigma^2 - h(\sigma)\,, \quad (1)$$

where

$$h(\sigma) = -\frac{1}{2}\int_{-\infty}^{\infty}\Big(\frac{1}{\sigma\sqrt{2\pi}}\big(e^{\frac{-(y-1)^2}{2\sigma^2}} + e^{\frac{-(y+1)^2}{2\sigma^2}}\big)\Big)$$
$$\log\Big(\frac{1}{\sigma\sqrt{2\pi}}\big(e^{\frac{-(y-1)^2}{2\sigma^2}} + e^{\frac{-(y+1)^2}{2\sigma^2}}\big)\Big)dy\,. \quad (2)$$

Clearly, the compression rate is a function of σ^2 and if the correct compression rate is not known at the encoder, lossless decoding is not possible at the decoder.

Practical SW coding is based on finite rate and finite length codes as follows. According to [3], if the parity check matrix for this code is H_p of size $(n-k) \times n$, first the syndrome S_l for a block X_l of source is created by calculating $S_l = H_p X_l$ and then S_l is sent to the decoder. This indicates a compression rate of $(n-k)/n$. At the decoder, exploiting SI, an estimate $\hat{X}_l$ is found such that $S_l = H_p \hat{X}_l$ and the Hamming distance between X_l and $\hat{X}_l$ is minimized. This is ideally done by Maximum-Likelihood decoding, but iterative decoding algorithms based on Turbo or LDPC codes are usually used in practice. In this paper, the method of [5] has been used for practical implementations in Section V.

B. System Model

Figure 1 shows the general system model in this paper. $X^n = \{x_i\}_{i=0}^{\infty}$ is the source to be encoded where x_i are binary random variables with uniform distribution. $Y^n = \{y_i\}_{i=0}^{\infty}$ is the correlated SI available to the decoder. To model the correlation, we have $y_i = x_i + z_i$, where z_i is Gaussian distributed with variance σ^2. The source is encoded in frames of length n_f and it is assumed that σ is constant for each separate frame, but varies from one frame to another. The random variable σ follows the PDF $g_\sigma(\sigma)$ and cumulative distribution function $\mathcal{G}(\sigma) = \int_0^\sigma g_\sigma(x)dx$, where the range of σ is denoted by Σ, usually including all non-negative real numbers.

Encoding is performed using a multi-mode encoder with a set $\mathcal{R}$ of pre-designed SW codes for compression, where $|\mathcal{R}| = M$. The decoder is assumed to estimate σ perfectly in each frame [10], and based on which select an encoder mode or equivalently the SW code to be used. The decoder then sends the index $i, i \in \{1, 2, \ldots, M\}$ of the selected mode to the encoder via a rate-limited feedback channel with the rate $R_f = \lceil \log_2(M) \rceil$ without delay or error. Therefore, Σ is partitioned into M disjoint intervals (modes) in an optimized manner and each interval is associated with one code. The partitions are in the form $[T_{i-1}, T_i), i = 1, 2, \ldots, M$. The probability of each mode being used is then equal to $p_i = \int_{T_{i-1}}^{T_i} g_\sigma(x)dx$.

Given a set of SW codes $\mathcal{R}$ with rates $R_i, i \in \{1, \ldots, M\}$, the multi-mode SW design problem is to determine the partitioning of Σ such that the source coder average rate $R_{\text{avg}} = \sum_{i=1}^{M} p_i R_i$, is minimized (maximum compression). Without loss of generality, we assume that $R_i > R_{i-1}, 2 \leq i \leq M$. To the best of the authors' knowledge, this paper is the first in the context of Slepian-Wolf coding to formulate the rate control mechanism as the solution to an optimization problem. In Section III, performance bounds to this problem are obtained when ideal (lossless) SW codes are considered and their corresponding rates may also be designed. In Section IV, the design problem with practical SW codes and hence a practical lossless decoding constraint is addressed. This is accomplished by modeling the decoding error performance of each SW code by a function $f_i(.)$. Specifically, $f_i(\alpha)$ denotes the probability of bit error at the SW decoder, when $\sigma^2 = \alpha$ and α is known to both encoder and decoder.

III. Rate Adaptation for Ideal Slepian-Wolf Coding

Assuming ideal SW codes with infinite block length and zero error probability, one can obtain a lower bound for the average rate and use it as a benchmark for the performance of practical SW codes. In this direction, no constraint on the rate of the codes are assumed, i.e., $R_i \in [0, 1]$, and indeed, the code rates are design parameters. This is in contrast to practical SW codes with predetermined rates.

Formally stating the problem, the objective is to minimize the average rate (maximize the compression ratio) for lossless decoding. In this case, the probability of decoding error goes to zero as the block length goes to infinity.

Problem 1:

$$\min_{T_i \in \Sigma, R_i, i=1,2,\ldots,M} R_{\text{avg}} = \sum_{i=1}^{M} p_i R_i$$
$$\text{s.t.}\{\bar{P}_e \to 0 | n_f \to \infty\} \tag{3}$$

where $\bar{P}_e$ denotes the average probability of error and $R_i \in [0, 1]$. To solve this problem, we set

$$R_i = \mathcal{H}(T_i) = 1 + \frac{1}{2}\log(2\pi e T_i^2) - h(T_i)\,. \tag{4}$$

This is due to the fact that each T_i in fact corresponds to a value of σ and by definition of conditional entropy and as given in equation (1), $\mathcal{H}(T_i)$ is the minimum achievable rate for $\sigma = T_i$. It can be simply verified that if $R_i < \mathcal{H}(T_i)$ the probability of error cannot go to zero when $n_f \to \infty$. After setting R_i for given T_i, finding the optimal partitioning of Σ, or equivalently the values of T_i, completes the solution to problem 1. Algorithm 1 is proposed for this purpose.

Algorithm 1:

- Initialize $T_{i,0}$ for $i = 1, 2, \ldots, M-1$ as the interval thresholds at iteration 0 arbitrarily, but satisfying $T_{i,0} < T_{j,0}$ for any $i < j$. Set $T_{0,0} = \min(\Sigma)$ and $T_{M,0} = \max(\Sigma)$.
- Choose the values of η and k_{max} as predefined constants to control the number of iterations.

Now, for each iteration $k = 1, 2, \ldots, k_{\text{max}}$, do the following,

1) For each $i = 1, 2, \ldots, M-1$, find $T_{i,k}$ such that

$$\frac{\partial R_{\text{avg},k}(T_{i,k})}{\partial T_{i,k}} = 0, \qquad T_{i-1,k} < T_{i,k} < T_{i+1,k}\,. \tag{5}$$

where

$$\frac{\partial R_{\text{avg},k}(x)}{\partial x} = g_\sigma(x)\big[\mathcal{H}(x) - \mathcal{H}(T_{i+1,k})\big] + \big[\mathcal{G}(x) - \mathcal{G}(T_{i-1,k})\big]\frac{\partial \mathcal{H}(x)}{\partial x}\,. \tag{6}$$

If there are multiple solutions for $T_{i,k}$, select the one which results in the smallest average rate $R_{\text{avg},k}$. Note that in calculating $T_{i,k}$ using (5), all other $T_{j,k}$, $j \neq i$ are kept fixed.

2) Calculate the average rate, $R_{\text{avg},k}$, using the final partitioning at iteration k. If $(R_{\text{avg},k-1} - R_{\text{avg},k})\ /\ (R_{\text{avg},k-1}) < \eta$ or $k > k_{\text{max}}$, stop.

More details on Algorithm 1 are presented below.

Proposition 1: The average rate computed using Algorithm 1 is reduced in each run of its step 1.

Proof: From (3) and (4), we have

$$R_{\text{avg}} = \sum_{i=1}^{M} \int_{T_{i-1}}^{T_i} g_\sigma(x)\mathcal{H}(T_i)dx \tag{7}$$
$$= \beta + \int_{T_{i-1}}^{T_i} g_\sigma(x)\mathcal{H}(T_i)dx + \int_{T_i}^{T_{i+1}} g_\sigma(x)\mathcal{H}(T_{i+1})dx\,,$$

where β is a constant independent of T_i. As evident in (6) for $x = T_i$, noting that $\mathcal{H}^{'}(T_i) = \partial\mathcal{H}(T_i)/\partial T_i$ is differentiable for $T_i > 0$, $\partial R_{\text{avg}}/\partial T_i$ exists for all values of T_i and continuous $g_\sigma(x)$.

In the following, we show that Algorithm 1 is able to find at least one minimum for R_{avg} in each step. It can be verified that

$$R_{\text{avg}}|_{T_i \to T_{i-1}} = R_{\text{avg}}|_{T_i \to T_{i+1}} = \tag{8}$$
$$= \beta + \mathcal{H}(T_{i+1})(\mathcal{G}(T_{i+1}) - \mathcal{G}(T_{i-1})).$$

Also from (6), we have

$$\lim_{T_i \to T_{i-1}^{+}} \frac{\partial R_{\text{avg}}}{\partial T_i} = g_\sigma(T_{i-1})(\mathcal{H}(T_{i-1}) - \mathcal{H}(T_{i+1})) < 0\,, \tag{9}$$

$$\lim_{T_i \to T_{i+1}^-} \frac{\partial R_{\text{avg}}}{\partial T_i} = \mathcal{H}^{'}(T_{i-1})(\mathcal{G}(T_{i+1}) - \mathcal{G}(T_{i-1})) > 0, \quad (10)$$

which follow as both $\mathcal{H}(.)$ and $\mathcal{G}(.)$ are increasing functions of their arguments or consequently $\mathcal{H}(T_{i-1}) < \mathcal{H}(T_{i+1})$, $\mathcal{H}^{'}(T_{i-1}) > 0$, and $\mathcal{G}(T_{i+1}) - \mathcal{G}(T_{i-1}) > 0$. Equations (8), (9) and (10) are sufficient conditions that $R_{\text{avg}}(T_i)$ has at least one minimum in the interval $[T_{i-1}, T_{i+1})$, which is found using (6). This indicates that R_{avg} is in fact reduced in each run of the step 1 of Algorithm 1. ■

Remark 1. $R_f \to \infty$ is associated with infinite number of encoder modes or having an ideal feedback channel with no rate limit. This eliminates any uncertainty at the encoder and the SW bound is then achieved for an equivalent encoder which is fully aware of SI statistics. This results in the minimum possible rate. In such case we have,

$$R_{\text{min}} = \int_0^{\infty} \mathcal{H}(x) g_\sigma(x) dx \quad (11)$$

which can be calculated numerically. This is referred to as SI-Aware SW Coding bound (SIA-SWC) and is used for comparison in Section V.

IV. Rate Adaptation for Practical Slepian-Wolf Coding

The performance bounds of multi-mode rate-adaptation scheme for SW coding with rate-limited feedback was studied in Section III using ideal (lossless) SW codes. For practical SW coding and in presence of uncertain SI, the encoder may only use a certain number of pre-designed codes with predetermined rates. These codes are not ideal and may involve possible decoding error.

A. Rate-Adaptation Based on Practical Codes

The following design problem formulates the minimization of the average rate when finite length and discrete rate SW codes are used and their probability of error performance are taken into account.

Problem 2:

$$\min_{T_i \in \Sigma, i=1,2,\ldots,M} R_{\text{avg}} = \sum_{i=1}^{M} R_i \int_{T_{i-1}}^{T_i} g_\sigma(x) dx \quad (12)$$

$$\text{s.t.} \quad \{\bar{p}_i = \int_{T_{i-1}}^{T_i} g_\sigma(x) f_i(x) dx < p_0 \quad | \quad R_i \in \mathcal{R}\}$$

where $\bar{p}_i$ denotes the average probability of error in mode i. Note that in this design, practical lossless compression has been interpreted as the average probability of error per mode being limited to a small p_0. Satisfying the stricter per-mode average probability of error constraint also satisfies the overall average probability of error.

In the following, we present an algorithm to solve problem 2. The proposed solution is general in the sense that it is independent of the selected set of codes, their rates, and their error probability performance.

Algorithm 2:
Set $T_i = 0$ and select ϵ a small number. For $i = 1, 2, \ldots, M$ do the following,

```
1) Set $T_i = T_{i-1}$.
2) Increase $T_i$ with a step size $\epsilon$.
3) Calculate $\bar{p}_i$, if $\bar{p}_i < p_0$, go to (2). Else reduce $T_i$ by $\epsilon$.
```

Intuitively, with $R_i > R_{i-1}$, in order to reduce R_{avg}, the lower rates are to cover as much of the range of σ as possible. Thus, the thresholds are updated from T_1 to T_M. Each one is increased until the probability of error constraint for the corresponding mode is met with equality. To show that R_{avg} is decreased at each step of Algorithm 2, we present Proposition 2.

Proposition 2: Suppose that Σ is partitioned with the intervals $[T_{j-1}, T_j)$, $j = 1, 2, \ldots, M$. If all thresholds except T_i are fixed, in order for R_{avg} to be reduced, T_i must be increased.

Proof: Suppose that all thresholds $\{T_j\}_{j=1}^{M}, j \neq i$ are fixed and only T_i is to be updated. Also suppose that $T_i^{'}$ refers to the modified T_i value, and define $R_{\text{avg}}^{'}$ as the updated value for R_{avg} when T_i is modified. Now, we have

$$\begin{aligned}
&R_{\text{avg}}^{'} - R_{\text{avg}} = \\
&+ \Big(\gamma + \int_{T_{i-1}}^{T_i^{'}} R_i g_\sigma(x) dx + \int_{T_i^{'}}^{T_{i+1}} R_{i+1} g_\sigma(x) dx\Big) \\
&- \Big(\gamma + \int_{T_{i-1}}^{T_i} R_i g_\sigma(x) dx + \int_{T_i}^{T_{i+1}} R_{i+1} g_\sigma(x) dx\Big) \quad (13a) \\
&= \Big(\int_{T_{i-1}}^{T_{i+1}} R_i g_\sigma(x) dx + \int_{T_i^{'}}^{T_{i+1}} (R_{i+1} - R_i) g_\sigma(x) dx\Big) \\
&- \Big(\int_{T_{i-1}}^{T_{i+1}} R_i g_\sigma(x) dx + \int_{T_i}^{T_{i+1}} (R_{i+1} - R_i) g_\sigma(x) dx\Big) \\
&= \int_{T_i^{'}}^{T_i} (R_{i+1} - R_i) g_\sigma(x) dx \quad (13b)
\end{aligned}$$

where $\gamma = \sum_{j=1}^{M} \int_{T_{j-1}}^{T_j} R_j g_\sigma(x) dx\,, j \neq i, i+1$. The term $(R_{i+1} - R_i) g_\sigma(x)$ is always positive as the code rates are assumed ordered and $g_\sigma(x)$ is positive as a PDF. Therefore, it is a necessary and sufficient condition for R_{avg} to decrease that $T_i^{'} > T_i$. ■

B. Mode Selection

In practice, due to limitations for the feedback channel rate R_f and for reduced encoder/decoder complexity, only a limited number of modes equal to $N = 2^{R_f} < M$ may be allowed. For that reason in the following, we propose the Algorithm 3 to select a suitable subset of modes with the desired size N and their associated rates and intervals from the original result obtained using Algorithm 2.

Algorithm 3:

- Initialization: Given the set $\mathcal{R}$, use Algorithm 2 to design a rate-adaptive multi-mode SW code. Consider the resulting thresholds in the sequel.
- Perform the followings for $m = 1, 2, \ldots, |\mathcal{R}| - N$.
- For $i = 1, 2, \ldots, |\mathcal{R}|$,
  1) Merge each two partitions in the form of $[T_{i-1}, T_i)$ and $[T_i, T_{i+1})$, into one single partition $[T_{i-1}, T_{i+1})$.
  2) Remove the code in $\mathcal{R}$ with the rate R_i, associated with $[T_{i-1}, T_i)$, temporarily from $\mathcal{R}$ and assign R_{i+1} to the partition $[T_{i-1}, T_{i+1})$.
  3) Calculate R_{avg} and store its value as R^i_{avg}.
- Select the merging of intervals corresponding to $\arg\min_i R^i_{\text{avg}}$.

Note that by selecting R_{i+1} for the merged partitions, the probability of error constraint is still satisfied because $R_{i+1} > R_i$ and SW code $(i+1)$ can be used for compression instead of code i without incurring more error. The presented algorithm provides a low complexity but suboptimal solution to select a subset of N codes out of M available codes for the multi-mode SW encoder. Using an exhaustive search to this end, involves running Algorithm 2 for each code subset, which equals $M!/(N!(M-N)!)$ subsets in total. However, Algorithm 3 has only $M-N$ steps (a small number). Also, the computational cost for each step, which only consists of merging partitions, is much smaller than cost of running Algorithm 2. Algorithm 3 is used for mode selection in simulations of Section V.

V. Simulations and Results

In this section, we present a simulation setting and corresponding results to validate and compare the designs of Sections III and IV. We used a set of high-performance LDPC codes from the DVB-S2 standard for SW compression. The selected set consists of 11 SW codes with discrete rates as in Table 1. The bit error rate performance of this set was obtained via extensive simulations. The probability of error function $f_i(\sigma^2)$ for the code with rate R_i is modeled by

$$f_i(\sigma^2) = \frac{1}{(1 + e^{c_i(1/\sigma^2 - b_i)})^{d_i}}\,, \tag{14}$$

for small (less than 10^{-2}) bit error rates, where b_i, c_i, d_i are constants that are obtained using fitting techniques as presented in Table 1. It is noteworthy that a similar error performance model as in (14) has been used to model channel decoding error in [14].

For the simulations, it is assumed that the parameter σ^2, as introduced in Section II, is Rayleigh distributed with parameter θ^2, i.e, if $u = \sigma^2$, then $f_u(u) = \frac{u}{\theta^2} e^{-\frac{u^2}{2\theta^2}}$, $u > 0$. A greater θ^2 implies more uncertain SI. We also set $p_0 = 1 \times 10^{-4}$.

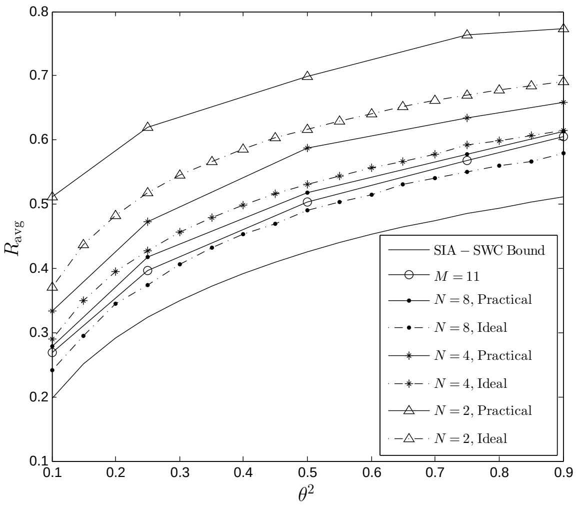


Figure 2. R_{avg} for different levels of SI uncertainty controlled by θ^2

Figure 2 depicts the performance of the proposed multi-mode SW coding quantified by R_{avg}, as a function of θ^2 for different number of modes. Using ideal SW codes and Algorithm 1, a lower bound for R_{avg} is presented. As expected, adding to the number of modes or equivalently the feedback rate, reduces the average rate. Using ideal SW codes with feedback rates $R_f = 1, 2, 3$ bps and for values of $\theta^2 > 0.2$ the gap from SIA-SWC bound (ideal feedback) approximately equals 0.188, 0.104, and 0.060 bps, respectively. For practical SW codes and using 2, 4, 8, and 11 modes, this gap is approximately 0.278, 0.145, 0.093, and 0.084 bits per symbol (bps), respectively.

The gap between ideal and practical SW code performance for $\theta^2 > 0.2$ equals 0.090, 0.047, 0.033 bps, respectively for $R_f = 1, 2, 3$ bps. For small values of θ^2, this R_{avg} gap is larger, e.g., for $\theta^2 = 0.1$ and $R_f = 1$ bps, it amounts to 0.141 bps. This small gap between practical and ideal code performance is more due to using finite length and discrete rate codes, and not to sub-optimality of Algorithm 3. This is supported by performance comparison of Algorithm 3 for mode selection and an exhaustive search solution as depicted in Figure 3. As evident, the incurred gap due to sub-optimality of the proposed Algorithm 3 is negligible over a wide range of values of R_f and θ. This is certainly outweighed by its much smaller complexity in comparison to an exhaustive search.

It is noteworthy that the gap between ideal and practical code performance decreases as R_f increases. This is due to the fact that as R_f and hence the number of modes (codes) increase, the set $\mathcal{R}$ approximately resembles a set of continuous rate codes. It is very interesting that the induced rate loss using 11 modes for compression is only 0.084 bps. This is comparable with the suggested SI aware

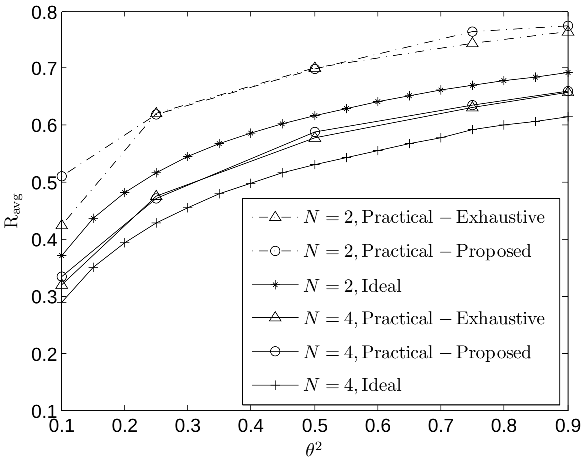


Figure 3. Performance comparison of the exhaustive search and the proposed method for mode selection (Algorithm 3)

Table I
FITTING PARAMETERS FOR MODELING THE PERFORMANCE OF LDPC-BASED SW CODES

R_i	b_{i}	c_{i}	d_{i}	R_i	b_{i}	c_{i}	d_{i}
1/10	4.249	4.18	11.9	2/5	1.898	9.22	90.4
1/9	3.803	4.04	6.8	1/2	1.183	40.78	9.5
1/6	3.124	14.24	7.0	3/5	0.923	28.35	50.1
1/5	2.820	11.73	11.5	2/3	0.680	50.11	12.2
1/4	2.474	12.45	15.5	3/4	0.505	60.18	16.0
1/3	2.042	14.59	31.4	-	-	-	-

schemes of [8] and [9] with LDPC and Turbo-based SW codes. In comparison to [6], the superiority of the proposed method is of course due to its judicious use of feedback and hence reduced decoding complexity and delay as discussed in section I.

VI. CONCLUSION AND FUTURE WORK

In this paper, a rate adaptation framework for the problem of Slepian-Wolf coding in presence of uncertain side information at the encoder is presented which uses a multi-mode encoder accompanied by a well-designed feedback scheme. SW compression rate and other mode parameters based on practical SW codes are designed such that the average rate is minimized. A lower bound is also derived considering ideal lossless codes. Simulation results based on LDPC codes show that by using sufficient number of modes, a desirably small average rate gap from the theoretical bound with no uncertainty can be achieved.

For the future work, the generalization of the proposed scheme for the context of Wyner-Ziv coding may be considered. This generalization requires introduction of distortion to the average rate minimization problem and the use of quantizers. The generalized scheme can then be adapted to be used in contexts such as distributed video coding and wireless sensor networks.

ACKNOWLEDGMENT

This project was funded in part by Iran Telecommunications Research Center (ITRC).

REFERENCES

[1] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[2] F. Cabarcas and J. Garcia-Frias, "Approaching the Slepian-Wolf boundary using practical channel codes," *Signal Processing*, vol. 86, no. 11, pp. 3096–3101, 2006.

[3] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS):Design and construction," *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 626–643, Mar. 2003.

[4] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes," in *Proceedings GlobeCom*, 2001, pp. 1400–1404.

[5] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, Oct. 2002.

[6] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, Jan. 2005.

[7] H. Zhang and H. Ma, "Delay-efficient rate control for Wyner-Ziv video coding in wireless video sensor networks using network coding," in *ICME*, 2010, pp. 243–248.

[8] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive distributed source coding using low-density parity-check codes," in *Asilomar Conf. on Signals, Systems and Computers*, 2005, pp. 1203–1207.

[9] M. Zamani and F. Lahouti, "A flexible rate Slepian-Wolf code construction," *IEEE Trans. Commun.*, vol. 57, no. 8, pp. 2301 –2308, Aug. 2009.

[10] C. Ng, C. Tian, A. Goldsmith, and S. Shamai, "Minimum expected distortion in Gaussian source coding with uncertain side information," in *ITW*, 2007, pp. 454–459.

[11] F. Bassi, M. Kieffer, and C. Weidmann, "Wyner-Ziv coding with uncertain side information quality," in *Europeen Signal Process. Conf.*, 2010, pp. 2141 – 2145.

[12] Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 80–94, 2004.

[13] Z. Liu, S. Cheng, A. Liveris, and Z. Xiong, "Slepian-Wolf coded nested lattice quantization for Wyner-Ziv coding: High-rate performance analysis and code design," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4358–4379, 2006.

[14] D. J. C. MacKay and C. P. Hesketh, "Performance of low density parity check codes as a function of actual and assumed noise levels," *Electronic Notes in Theoretical Computer Science*, vol. 74, pp. 91–98, 2003.

An Overview and Classification of Approaches to Information Extraction in Wireless Sensor Networks

Tariq A. A. Alsbouí, Mohammad Hammoudeh, Zuhair Bandar, Andy Nisbet

School of Computing, Mathematics & Digital Technology
Manchester Metropolitan University
Manchester, UK
Email: tariq.al-sboui@stu.mmu.ac.uk, {m.hammoudeh,z.bandar,a.nisbet}@mmu.ac.uk

***Abstract*—Recent advances in wireless communication have made it possible to develop low-cost, and low power Wireless Sensor Networks (WSN). The WSN can be used for several application areas (e.g., habitat monitoring, forest fire detection, and health care). WSN Information Extraction (IE) techniques can be classified into four categories depending on the factors that drive data acquisition: event-driven, time-driven, query-based, and hybrid. This paper presents a survey of the state-of-the-art IE techniques in WSNs. The benefits and shortcomings of different IE approaches are presented as motivation for future work into automatic hybridisation and adaptation of IE mechanisms.**

***Keywords*-Wireless Sensor Networks; Information Extraction**

I. Introduction

The main purpose of a WSN is to provide users with access to the information of interest from data collected by spatially distributed sensors. In real-world applications, sensors are often deployed in high numbers to ensure a full exposure of the monitored physical environment. Consequently, such networks are expected to generate enormous amount of data [1]. The desire to locate and obtain information makes the success of WSNs applications, largely, determined by the accuracy and quality of the extracted information. The principal concerns when extracting information include the timeliness, accuracy, cost, and reliability of the extracted information and the methods used for extraction. The process of IE enables unstructured data to be retrieved and filtered from sensor nodes using sophisticated techniques to discover specific patterns [2]. Practical constraints on sensor node implementation such as power consumption (battery limits), computational capability, and maximum memory storage, make IE a challenging distributed processing task.

In terms of data delivery required by an application, IE in WSNs can be classified into four broad categories: event-driven, time-driven, query-based, and hybrid. In event-driven, data is only generated when an event of interest occurs , while, in the time-driven, data is periodically sent to a sink every constant interval of time. With query-based, the data is collected according to end user's demand. Finally, the hybrid approach is a combination of one or more of the above.

The rest of this paper is organised as follows: Section 2 identifies what types of information needs to be reported to end users. Section 3 looks at event-driven IE approaches and presents sample developments. Section 4, describes time-driven IE and recent successful deployments. Section 5, describes query-based IE and present some of the recent approaches. Section 6, describes and identify recent advances in hybrid IE methods. In section 7, a summary on future research direction for IE is discussed. Section 8 concludes this paper.

II. What needs to be reported?

IE is one of the most vital efforts to utilise the ever burgeoning amount of data returned by WSNs for achieving detailed, often costly task of finding, analysing and identifying needed information. The process of IE involves the classification of data based on the type of information they hold, and is concerned with identifying the portion of information related to a specific fact. In the context of WSNs, the notion of fact can be defined as a property or characteristic of the monitored phenomenon at a certain point in time or during a time interval. Fact can also refer to an event or action. An event is a pattern or exceptional change that occasionally appears in the observed environment [3]. Events have some distinct features that can be used as thresholds, e.g. temperature > 50, to make a distinction between usual and unusual environmental parameters.

An event may arise in many other forms. It can be a continuous, gradually occurs over time (e.g. temperature does not change instantly), and has obvious limit with normal environment parameters. In [4], complex events are defined as sequences of sensor measurements over a period of time indicating an unusual activity in the monitored environment. In WSNs, the network owners may be unaware in advance what type of events may occur. This is because one of the ultimate goals of such networks is to discover new events and interesting information about the monitored phenomenon. For this reason, threshold based event detection methods are not always efficient to identify and extract event-based facts. From this deficiency arise the need for periodic, query-based, and hybrid IE approaches.

Events can be further classified into two categories: system events and environmental events. System events are concerned with architectural or topological changes, e.g. a mobile node entered a cluster area. Environmental events are concerned

Table I
OVERVIEW OF THE SELECTED APPROACHES TO IE

Event-driven	[3], [7]–[15]
Time-driven	[9], [16]–[20]
Query-based	[21]–[29]
Hybrid	[30]–[34]

with the occurrences of unusual changes across the monitored environment, e.g. spotting a moving target [4].

Nodes organisation plays an important role in IE because it defines, among other factors, the cost (amount of energy required to collect raw data), accuracy (level of coverage), reliability (e.g. timeliness) of extracted information. The organisation of nodes can be either centralised or hierarchical. In the centralised approach, data collected by all nodes are sent towards a sink node using single or multi-hop communication [5]. However, this approach does not provide scalability, which is a main design factor for WSN. Also, it causes communication bottlenecks and transmission delays due to congestions especially in areas around the sink [6]. To overcome the problems in the centralised approaches, hierarchical techniques has been proposed as an effective solution for achieving longer network lifetime and better scalability.

Since the number of existing IE approaches is significantly large, it will not be feasible to provide a detailed description of each approach. Instead, we have selected recent approaches that particularly represent directions of future research without focusing on the details of these approaches. However, characteristics of various approaches that are common for the approach they apply will be presented. Table I lists the reviewed approaches and some older approaches for the more interested readers. In Sections III to VI the approaches are presented based on the categorisation so as related subcategories are discussed in the common context. To make the analysis of different approaches more logical and to set up a common base for their comparison and connection we consider some qualitative criteria.

III. EVENT-DRIVEN IE

A. Description and Operation

In event-driven approaches to IE, the initiative is with the sensor node and the end user is in the position of an observer, waiting for incoming information. Any node may generate a report when a significant event (e.g, a change of state) or an unusual event (e.g, fire) occurs. Event-driven is valuable for detecting events as soon as they occur over a specified region. In the simplest form, sensor nodes are pr-econfigured with threshold values that when exceeded indicate an event.

Event-driven approaches incur low power consumption and requires low maintenance. Among the benefits of this class of approaches are: they reduce the amount of communication overhead by applying local filtering on collected data to determine whether to send new data or not; they implement local mechanisms to prevent multiple nodes reporting the same event; they exploit redundancy to reduce the number of false alarms; they allow timely responses to detected events; they are easy to implement and configure; they allow distributed processing at the node level or within a group of node to collaboratively detect an event; and they are suitable for time critical applications, e.g. forest fire monitoring or intrusion detection.

However, there are a number of limitations to the event-driven IE. First, it is difficult to capture events of spatio-temporal characteristics. Second, detecting complex event may require non trivial distributed algorithms, which require the involvement of multiple sensor nodes [9]. Third, due to the fact that events occur randomly, some nodes generate higher rates of data than other nodes. This will lead to unbalanced workloads among sensor nodes. Fourth, it is not suitable for continuous monitoring applications, where sensed measurements change gradually and continuously. Finally, due to sensors measurement inaccuracies, event-driven approaches may potentially generate false alarms.

B. Event-driven Approaches to IE

In earlier studies, events were detected with a user-defined threshold values [7], [8]. In such approaches sensor nodes are pr-econfigured with a static threshold value. When the sensor node reading deviates from the pre-defined thresholds, this indicates an event, which triggers the node to convey it is data back to the sink. To overcome some of the inherent problems in the threshold-based event-detection, [13] presented a data fusion tool to increase resilience of event detection techniques. They introduced two levels for event detection: at the first level, each sensor node will individually decide on detecting event using classifier (naive bayes). At the second level, fusion technique is placed in a higher level (e.g, cluster head) and used to distinguish between outliers. Outliers are measurements that differ from the normal pattern of sensed data occurring at individual nodes and events that more nodes agree upon [15]. The data fusion approach reduces the number of transmission, thus extends the network life time. It also reduces the number of false alarms, since cluster heads are able to distinguish between anomalies and event. Because of its distributed nature, this approach is scalable. However, processing data at the cluster head introduces delays in reporting an event. Moreover, the efficiency of the approach depends on the efficiency of cluster formation methods. For instance, many clustering algorithms result in unbalanced clusters.

Another threshold-based approach [12] introduced double decision mechanisms. A sensor may decide about the presence of an event of interest either directly or asking for additional data from nearby nodes. This approach minimises the energy consumption since the final process detection is activated only when it is needed, and there is no need for fusion centre to process the data as a fixed number of nodes will take the responsibility to make decisions about the occurrences of an event. However, it is always difficult to determine node's neighbours. Although these approaches can reduce communication overhead and report events promptly, however, it is difficult to define the optimal threshold values. Also, the complications of implementing an efficient sleep-wake cycles may dissipate the gained energy savings.

More advanced approaches, such as SAF [35] and Ken [18], exploit the fact that physical environments frequently exhibit predictable stable and strong attribute correlations to improve compression of the data communicated to the sink node. The basic idea is to use replicated dynamic models to reflect the state of the environment being monitored. This is done by maintaining a pair of dynamic probabilistic models over the WSN attributes with one copy distributed in the network and the other at the sink. The sink computes the expected values of the WSN attributes according to the defined prediction model and uses it extract information. When the sensor nodes detect anomalous data that was not predicted by the model within the required certainty level, they route the data back to the sink. This approach is subject to failure as basic suppression. It does not have any mechanism to distinguish between node failure and the case that the data is always within the error bound. Ken is not robust to message loss; it relies on the Markovian nature of the prediction models to presume that any failures will eventually be corrected with model updates, and the approximation certainty will not be affected by the missed updates. They propose periodic updates to ensure models can not be incorrect indefinitely. This approach is not suitable for raw value reconstruction; for any time-step where the model has suffered from failures and is incorrect, the corresponding raw value samples will be wrong. Finally, as the approach presented in [17], SAF and Ken can only handle static network models.

A decentralised, lightweight, and accurate event detection technique is proposed in [36]. The technique uses decision trees for distributed event detection and a reputation-based voting method for aggregating the detection results of each node. Each sensor node perform event detection using it is own decision tree-based classifier. The classification results, i.e. detected events, from several nodes are aggregated by a higher node, e.g, a cluster head. Each node sends it is detected events, called detection value, to all other nodes in it is neighbourhood. The detection value will be stored in a table. Finally, tables are sent to the voter (e.g. cluster head), which in turns decides to make a final decision among different opinion. The decision tree approach provides accurate event detection and characterised by low computational and time complexities. However, the processing of data at the cluster head will introduce further delays in reporting an event.

IV. Time-driven IE

A. Description and Operation

In time-driven approaches to IE, a sensor node periodically generates a report from the physical environment to give the end-user its current status. The reporting period may be preconfigured or set by the end-user depending on the nature of the monitored environment and applications requirements.

Time-driven approaches have the ability to enable arbitrary data analysis, they provide continuous monitoring of the WSN to reflect environmental changes, they scale to handle millions of nodes (through aggregation), they extend network life time by sending nodes to sleep between transmissions, they can reduce congestion and improve system reliability by scheduling nodes to transmit at different times, they explicitly incorporate resource capacity, and highlights unused resources. However, there are a number of limitations to the time-driven approaches. First, they are limited to specific set of applications where consistent changes occur across the network, e.g. agricultural applications. Second, a large portion of the returned data might be redundant and not useful for the end-user thereby resulting in wastage of resources. Third, nodes have to maintain global clock and deal with synchronisation issues. Finally, it is extremely difficult to define optimal time intervals.

B. Time-driven Approaches to IE

In time-driven IE, most of the published work in the literature is based on probabilistic models that attempt predict the next value that the sensor is expected to acquire. For example, Ken's [18] model exploits the spatio-temporal data correlations while guaranteeing correctness. It involves placing a dynamic probabilistic model on the sensor node and on the sink, and these models are always kept in synchronisation for periodic updates. Similar approaches to Ken have been suggested in [9], [37]. In contrast to Ken, these approach uses dynamically changing subset of the nodes as samplers where the sensor readings of the sampler nodes are directly collected, while, the values of non sampler nodes are predicated through probabilistic models that are locally and periodically constructed. All approaches in [9], [18], [37] save energy by reducing the number of transmitted messages. However, the additional cost to maintain models synchronised is not negligible.

Another approach called Cascading Data Collection (CDC) is presented in [19]. In CDC only a subset of sensor nodes are selected randomly to periodically transfer data back to the sink node. The mechanism is distributed and only utilises local information of sensor node. The CDC reduces communication cost by allowing only a subset of sensor nodes to periodically transmit readings back to the sink. However, the CDC uses packet aggregation at an intermediate node, which introduces undesirable communication delays. The work presented in [38] takes CDC one step further by enabling each node to use its local and neighbourhood state information to adapt its routing and MAC layer behaviour.

V. Query-based IE

A. Description and Operation

Query-based approaches to IE, typically involve request-response interactions between the end-user or application components and sensor nodes. End users issue queries in an appropriate language, and then each query is disseminated to the network to retrieve the desired data from the sensors based on the description in the query.

Query based approaches provide a high level interface that hides the network topology as well as radio communication from end users. Queries can be sent on demand or at fixed intervals. They provide a solution if the data needs to be retrieved from the entire network.

However, there are a number of limitations to the query-based approaches. First, most of existing query languages do not provide suitable constructs to easily articulate spatio-temporal sense data characteristics. Second, it is difficult to formulate queries using current languages that represent higher level behaviour, or specify a subset of nodes that have significant effect on the query answer. This may result in generating large amount of data of which big portion is not useful for the end user. Third, to the best of our knowledge, there is no published work that fully exploits all the potentials of different heterogeneous resources in WSN applications in a context-aware manner. Forth, approaches that take a database view of the network are inclined more towards the extraction of the reactive behaviour of the WSN and suggestions were made that the active database should be viewed as two end-points of the range of rule-based languages in databases [39]. Finally, though declarative languages are came into view in WSNs settings, the trigger that are the fundamental means for specifying the reactive behaviour in a database have not yet been maturely developed.

B. Query-based Approaches to IE

Query-based systems, applies techniques used in traditional database systems to implement IE. A query is sent to the network and data is collected according to the description in the query. COUGAR [21] was the first project that attempted to introduce the concept of WSN as a distributed database. It allows the end user to issue a declarative query (SQL) for retrieving information. The authors introduced a query layer between the application layer and the network layer. The query layer comprises a query proxy, which is placed on each sensor node to interact with both the application layer and the networking layer. The goal of the query proxy is to perform in-network processing. In-network processing increases efficiency in terms of power consumption, and reduces the amount of data that needs to be sent to the gateway node. The user does not need to have knowledge about the network, or how the data is retrieved or processed. However, COUGAR is incapable of capturing complex events, e.g. of spatio-temporal nature, or a produce queries that targets only a subset of the network [22].

A similar approach to COUGAR is proposed in [23]. TinyDB is a query processing system, which extracts information from the data collected by the WSN using the TinyOS operating system. TinyDB maintains a virtual database table called SENSORS. It disseminates the queries throughout the network by maintaining a routing tree (spanning tree) rooted at the end point (usually the user's physical location). Every sensor node has its own query processor that processes and aggregates the sensor data and maintains the routing information. TinyDB is extensible and complete framework with effective declarative queries. In-network processing reduces the amount of data that is required to be sent to the sink, thus, energy consumption is reduced. However, data does not include the georeferencing of sensor nodes for spatial quires, and tight correlation among routing and queries.

In [24], a new data collection algorithm that aims on reduce energy consumption by focusing on selective aggregate queries. The proposed algorithm, named, PDT (Pocket Driven Trajectories) deals with queries that aggregate data only from a subset of all network nodes. PDT is based on the logical assumption that spatial correlation in sensor values coupled with query selectivity gives rise to a subset of participating nodes formed by one or more geographically clustered sets (pockets). The algorithm starts by discovering the set of pockets for a given query. Then, the aggregation tree to the spatially optimal path connecting these pockets is aligned. The PDT algorithm reduces the amount of communication and is scalable for large WSNs. However, PDT introduces a delay in reporting data to a sink, because data is processed at an intermediate node.

In [25], a mobile sink moving through the sensing field issues a query to a specific area. The sensor node that is closest to the centre of the area of interest elects itself as a cluster head. The cluster head performs data collection and aggregation, then the aggregated data is sent back to the mobile sink. The proposed mobile sink approach saves energy by choosing an optimal time and location to disseminate query. The area-based querying and the mobile sink makes the approach scalable for large-scale WSNs. However, the proposed approach is limited to a set of applications, specifically, intelligent transportation system and environmental monitoring. Also, it introduces undesirable delay since the data is aggregated at the mobile sink.

The authors in [26], proposed a query processing algorithm, that allows the user to specify a value and time accuracy constraints based on an optimised query plan. Using these optimisation constraints, the algorithm can find an optimal sensing and transmission of attribute readings to sink node. Rather than sending sensors readings directly to the sink, the proposed algorithm report only updates. This results in considerable reduction in communication costs. However, the algorithm does not support dynamic adjustment of accuracy constraints.

More recently in [27], the authors designed and implemented a distributed in-network query processing, called Corona. Corona is composed of three components: the query engine that is executed on the sensors; a host system on the clients PC that is connected to the sink; and GUI that is connected to the host system via TCP/IP. The Corona query processing provides multi-tasking capabilities by running multiple queries concurrently, which in turns reduces processing delays and communications cost by applying data aggregation. However, the language can not easily capture spatio-temporal events.

VI. Hybrid-based IE

A. Description and Operation

A hybrid approach is an approach that combines the functionality of two or more algorithms from different IE categories. Hybrid approaches aims to minimise the effect of the disadvantages of individual IE categories described above.

B. Hybrid Approaches to IE

Many hybrid approaches to IE have been recently proposed in the literature. In [34], the authors proposed a hybrid protocol that adaptively switches between time-driven and event-driven data collection. A sensor node is triggered to detect an event of interest, and from the point when an event detected to the point when the event becomes no longer valid, the protocol switches to behave as a time-driven protocol. During this period sensor nodes continuously report data to the sink. This protocol reduces unnecessary data transmission and minimise event notification time. However, it is not guaranteed to work well for all applications due to limitations of the PAD algorithm, such as if sensor nodes detecting an event are located at the border between clusters, those nodes in other clusters can be included only when clusters at the same level have used time-driven data dissemination

More recently, in [33], the authors proposed a hybrid framework similar to [9], [18], which deploy both of event-driven and query-based approaches to IE. The idea is to process continuous group-by aggregate queries, and to allow each sensor node to check whether sensor readings satisfy local predicates based on a predefined thresholds. Then, nodes send only data that satisfy local predicates to their cluster heads, which in turns process the data to answer the query as accurate as possible. The proposed hybrid framework is able to target a subset of the network by using the group-by clause. It reduces communication cost by using one dimensional haar wavelets. However, it introduces a delay in reporting events since the data is processed at the cluster head.

In [32], the authors proposed energy-efficient hybrid data collection architecture similar to [25]. The aim is to enhance the network performance and reduce the total energy consumption by introducing mobile node entities. A mobile node is moving through the network deployment region to collect data from the static nodes over a single hop radio links. The mobile node visits the sink periodically to drop off the collected data. The proposed solution reduces energy consumption and communication overhead by moving the sink node near to the nodes to collect data. However, the mobile node introduces latency in transferring the data as it has to travel back to a sink.

VII. Discussion and possible future directions

Before concluding this paper, this section provides a discussion about research issues, and future directions in the area of IE in WSNs. This short survey revealed that most of the existing approaches to IE suffer from inherent problems that limit their applications including: they are application specific; characterised by poor spatio-temporal IE capabilities; consume high power; many approaches trade the amount and quality of returned information by energy consumption; they lack appropriate high-level interfaces that allow the user to set thresholds and issue queries; and the tight coupling between IE algorithms, applications, and hardware stacks leads to lack of code reuse. The lack of development frameworks means each new application has to be tackled from the ground up. These issues limit the usefulness of the developed IE approaches, making it hard to use them on anything other than the application it was designed for.

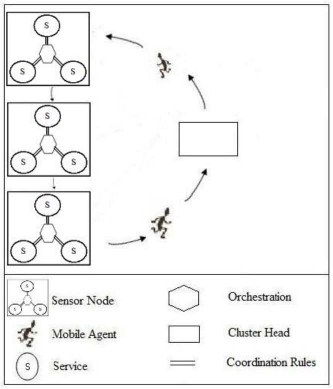


Figure 1. An integrated IE framework for WSNs.

The problems and limitations presented above are the opportunities we intend to follow in our future work. Possible solutions that we are currently investigating for the integration of the three IE approaches will be achieved through the use of coordination rules [40] and mobile agents [41].

Coordination rules are a set of modelling primitives, design principles and patterns that deal with enabling and controlling the collaboration among a group of software distributed agents performing a common task. If each algorithm in each IE category is viewed as a service, then the composition of these services will result in a complete IE framework. Service composition provides new services by combining existing services. The coordination rules specifies the order in which services are invoked and the conditions under which a certain service may or may not be invoked.

The mobile agent paradigm will be adopted to facilitate cooperation among services on different nodes. Mobile agent is a piece of software that performs data processing autonomously while migrating from node to node [41]. The agent can collect local data and perform any necessary data aggregation. Mobile agents can make decision autonomously without user input. They provide flexibility in terms of decision making, and reliability in terms of node failure [42].

Figure 1 shows an illustration of the described hybrid framework. It shows how services on one node are connected and how a service can access other services on remote node.

VIII. Conclusion

The main objective of this paper is to provide an understanding of the current issues in this area for better future academic research and industrial practice of WSNs IE. We have presented a review of the state of the art for IE approaches in WSNs. We discussed various approaches to IE. We also discussed the challenges as well as future research directions in developing a complete integrated WSNs IE framework.

REFERENCES

[1] M. Hammoudeh, R. Newman, and S. Mount, “An approach to data extraction and visualisation for wireless sensor networks,” *International Conference on Networking*, vol. 0, pp. 156–161, 2009.

[2] S. Sarawagi, “Information extraction,” *Found. Trends databases*, vol. 1, pp. 261–377, March 2008.

[3] C. Zhang, C. Wang, D. Li, X. Zhou, and C. Gao, “Unspecific event detection in wireless sensor networks,” *Communication Software and Networks, International Conference on*, vol. 0, pp. 243–246, 2009.

[4] R. Bhargavi, V. Vaidehi, P. Bhuvaneswari, P. Balamurali, and G. Chandra, “Complex event processing for object tracking in wireless sensor networks,” *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*, vol. 3, pp. 211–214, 2010.

[5] A. Iranli, M. Maleki, and M. Pedram, “Energy efficient strategies for deployment of a two-level wireless sensor network,” in *Proceedings of the 2005 international symposium on Low power electronics and design*, 2005, pp. 233–238.

[6] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 325 – 349, 2005.

[7] M. L. Segal, F. P. Antonio, S. Elam, J. Erlenbach, and K. R. De Paolo, “Method and apparatus for automatic event detection in a wireless communication system,” no. 6124810, September 2000.

[8] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees, “Deploying a wireless sensor network on an active volcano,” *IEEE Internet Computing*, vol. 10, pp. 18–25, 2006.

[9] B. Gedik, L. Liu, and P. S. Yu, “Asap: An adaptive sampling approach to data collection in sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, pp. 1766–1783, 2007.

[10] K. Pripužić, H. Belani, and M. Vuković, “Early forest fire detection with sensor networks: Sliding window skylines approach,” *Proceedings of the 12th international conference on Knowledge-Based Intelligent Information and Engineering Systems, Part I*, pp. 725–732, 2008.

[11] H. bin Wang, Z. Yuan, and C. dong Wang, “Intrusion detection for wireless sensor networks based on multi-agent and refined clustering,” in *Communications and Mobile Computing, 2009. CMC ’09. WRI International Conference on*, vol. 3, 2009, pp. 450 –454.

[12] S. Sardellitti, S. Barbarossa, and L. Pezzolo, “Distributed double threshold spatial detection algorithms in wireless sensor networks,” in *Signal Processing Advances in Wireless Communications, 2009. SPAWC ’09. IEEE 10th Workshop on*, 2009, pp. 51 –55.

[13] N. H. P. Bahrepour, M. Meratnia, “Sensor fusion-based event detection in wireless sensor networks,” *Mobile and Ubiquitous Systems: Networking Services, MobiQuitous, 2009. MobiQuitous ’09. 6th Annual International*, pp. 1–8, 2009.

[14] M. Bahrepour, N. Meratnia, and P. Havinga, “Use of ai techniques for residential fire detection in wireless sensor networks,” in *AIAI 2009 Workshop Proceedings*, vol. 475, July 2009, pp. 311–321.

[15] Y. Zhang, N. Meratnia, and P. Havinga, “Outlier detection techniques for wireless sensor networks: A survey,” *Communications Surveys Tutorials, IEEE*, vol. 12, no. 2, pp. 159 –170, 2010.

[16] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson, *Analysis of wireless sensor networks for habitat monitoring*, 2004, pp. 399–423.

[17] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, “Model-driven data acquisition in sensor networks,” in *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, 2004, pp. 588–599.

[18] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong, “Approximate data collection in sensor networks using probabilistic models,” in *Proceedings of the 22nd International Conference on Data Engineering*, 2006, pp. 48–.

[19] L. L. Li. Hong, Yu. Hongyi and A. Liu, “On the cascading data collection mechanism in wireless sensor networks,” *Wireless Communications, Networking and Mobile Computing, 2007.*

[20] L. Wang and A. Deshpande, “Predictive modeling-based data collection in wireless sensor networks,” in *Proceedings of the 5th European conference on Wireless sensor networks*, 2008, pp. 34–51.

[21] Y. Yao and J. Gehrke, “The cougar approach to in-network query processing in sensor networks,” *SIGMOD Rec.*, vol. 31, pp. 9–18, 2002.

[22] M. Bestehorn, K. Böhm, E. Buchmann, and S. Kessler, “Energy-efficient processing of spatio-temporal queries in wireless sensor networks,” in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS ’10, pp. 340–349.

[23] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “Tinydb: an acquisitional query processing system for sensor networks,” *ACM Trans. Database Syst.*, vol. 30, pp. 122–173, 2005.

[24] M. Umer, L. Kulik, and E. Tanin, “Optimizing query processing using selectivity-awareness in wireless sensor networks,” *Computers, Environment and Urban Systems*, vol. 33, no. 2, pp. 79 – 89, 2009.

[25] L. Cheng, Y. Chen, C. Chen, and J. Ma, “Query-based data collection in wireless sensor networks with mobile sinks,” in *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, 2009, pp. 1157–1162.

[26] J.-Z. Sun, “An energy-efficient query processing algorithm for wireless sensor networks,” in *Ubiquitous Intelligence and Computing*, 2008, vol. 5061, pp. 373–385.

[27] R. Khoury, T. Dawborn, B. Gafurov, G. Pink, E. Tse, Q. Tse, M. Gaber, U. Rohm, and B. Scholz, “Corona: Energy-efficient multi-query processing in wireless sensor networks,” in *Database Systems for Advanced Applications*, 2010, vol. 5982, pp. 416–419.

[28] P. Bonnet, J. Gehrke, and P. Seshadri, “Towards sensor database systems,” in *Proceedings of the Second International Conference on Mobile Data Management*, 2001, pp. 3–14.

[29] C. Chien-Chung Shen. Srisathapornphat, C. Jaikaeo, “Sensor information networking architecture and applications,” *Personal Communications, IEEE*, vol. 8, pp. 52–59, 2001.

[30] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8 - Volume 8*, 2000, pp. 8020–.

[31] T. E. Daniel, R. M. Newman, E. I. Gaura, and S. N. Mount, “Complex query processing in wireless sensor networks,” in *Proceedings of the 2nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, 2007, pp. 53–60.

[32] A. Rasheed and R. Mahapatra, “An energy-efficient hybrid data collection scheme in wireless sensor networks,” in *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, 2007, pp. 703 –708.

[33] C.-H. Lee, C.-W. Chung, and S.-J. Chun, “Effective processing of continuous group-by aggregate queries in sensor networks,” *J. Syst. Softw.*, vol. 83, pp. 2627–2641, December 2010.

[34] B.-D. Lee, “Adaptive data dissemination protocol for wireless sensor networks,” in *Security-Enriched Urban Computing and Smart Grid*, 2010, vol. 78, pp. 188–195.

[35] D. Tulone and S. Madden, “An energy-efficient querying framework in sensor networks for detecting node similarities,” in *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, 2006, pp. 191–300.

[36] M. Bahrepour, N. Meratnia, M. Poel, Z. Taghikhaki, and P. J. Havinga, “Distributed event detection in wireless sensor networks for disaster management,” in *Proceedings of the International Conference on Intelligent Networking and Collaborative Systems, INCoS 2010*, November 2010, pp. 507–512.

[37] G.-S. Ahn, “Information-driven tracking and access control in wireless ad hoc and sensor networks,” Ph.D. dissertation, Columbia University, 2009.

[38] R. Jurdak, P. Baldi, and C. Videira Lopes, “Adaptive low power listening for wireless sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 6, pp. 988–1004, 2007.

[39] G. Trajcevski and P. Scheuermann, “(reactive /+ proactive behavior) situation awareness in sensor networks,” in *Workshop on Research Directions in Situational-aware Self-managed Proactive Computing in Wireless Adhoc Networks*, 2009.

[40] J. Abreu and J. Fiadeiro, “A coordination model for service-oriented interactions,” in *Coordination Models and Languages*, 2008, vol. 5052, pp. 1–16.

[41] M. Chen, T. Kwon, Y. Yuan, and V. Leung, “Mobile agent based wireless sensor networks,” *Journal of Computers*, vol. 1, no. 1, 2006.

[42] D. Massaguer, C.-L. Fok, N. Venkatasubramanian, G.-C. Roman, and C. Lu, “Exploring sensor networks using mobile agents,” in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, ser. AAMAS ’06, 2006, pp. 323–325.

CPU-friendly Tracking in Wireless Sensor Networks

Neeta Trivedi, N. Balakrishnan
Supercomputer Education and Research Center
Indian Institute of Science
Bangalore, India - 560012
neeta.trivedi@gmail.com, balki@serc.iisc.ernet.in

***Abstract*—The problem of tracking multiple targets using Sequential Monte Carlo technique under the framework of Bayesian techniques for wireless sensor networks is discussed. Distributed filtering in wireless sensor networks is an active area of research owing to the high communication costs of centralized tracking. However, distributed filtering must carefully address the conflict between high correlation among signals picked up by neighboring sensors and detached sensing by far away nodes due to limited sensing radii. Further challenges relate to the processing and communication of large number of particles in resource-constrained sensor nodes. This paper proposes a novel integrated approach to network management and target tracking by which distributed tracking is achieved in a lightweight manner. Important contributions are 'Consensus Tracking' as a low-cost distributed solution for sensor tasking and 'Multitiling' as computationally efficient solution for managing and propagating particles.**

Keywords- multitarget tracking, Bayesian filtering, sequential Monte Carlo, particle filter, sensor networks.

I. Introduction

Advancements in electronics and communication technologies coupled with research in fusion techniques have made it possible to model the non-linear and non-Gaussian nature of most real-life problems more accurately, overcoming the limitations of Kalman filter and its variants. Applications are increasingly adopting the rigorous general framework of formal Bayes modeling for dynamic state estimation problems.

Wireless Sensor Networks (WSN) has been another interesting recent development. Large networks of small untethered nodes capable of sensing, communicating and computing have opened up entirely new possibilities.

In Bayes modeling of target-tracking applications, the goal is to estimate the density $f_{k|k}(x^k \mid z^{1:k})$ of the target set being in state x^k, given all observations up to time *k*. The estimate is performed recursively in two steps viz. prediction and update. Prediction amounts to obtaining the prior density

$$f_{k+1|k}(x^{k+1} \mid z^{1:k}) = \int f_{k+1|k}(x^{k+1} \mid x')\, f_{k|k}(x' \mid z^{1:k})\, dx' \qquad \ldots (1)$$

The update step uses the Bayes' rule to find the posterior

$$f_{k+1|k+1}(x^{k+1} \mid z^{1:k+1}) = \frac{f_{k+1}(z^{k+1} \mid x^{k+1})\, f_{k+1|k}(x^{k+1} \mid z^{1|k})}{\int f_{k+1}(z^{k+1} \mid x')\, f_{k+1|k}(x' \mid z^{1|k})\, dx'} \qquad \ldots (2)$$

Finite Set Statistics [1] extends Bayes single target tracking model to scenarios involving multiple sensors and multiple targets including target birth, death, merger and spawning by providing a unified, scientifically defensible probabilistic foundation for tracking.

The Bayes modeling problem has no closed form solution. For discrete state-spaces, exact inference is always possible, but may be computationally prohibitive [2]. A number of computationally tractable stochastic approximation techniques have been proposed. Sequential Monte Carlo (SMC) approximation or Particle filtering is one such technique that, given enough samples, guarantees to give exact answer [3]. The basic idea is to approximate the belief sate by a set of weighted particles or samples

$$f_{k|k}(x^k \mid z^{1:k}) \approx \sum_{i=1}^{N_s} w_{k|k}^i \theta(x_{k|k}^i) \qquad \ldots (3)$$

for any unitless function $\theta(x)$ of a state set variable x (likewise for sets X and Z in case of multitarget tracking). The particles approach the pdf 'in asymptotia'; the larger the number of particles, the better the approximation.

Tracking in the resource-constrained WSN poses even bigger challenges. Due to the limited field of sensing of individual nodes, the set of particles must be propagated not just temporally but also spatially for computing the belief state. Centralized solutions are not suitable due to latency and excessive energy requirement. Decision on the next node responsible for tracking a target as it leaves the sensing zone of one sensor must be taken in distributed manner. Managing and propagating the particles, typically large in numbers, is highly computation and communication intensive.

This paper makes two important contributions. First, it proposes 'Consensus Tracking', a method that works in integrated mode with network management functions to designate the node responsible for tracking a target in distributed and lightweight manner. Second, it proposes 'Multitiling', a method to reduce cost of managing and propagating the particles. To the best of our knowledge, no work has been reported addressing these problems in an integrated manner.

Handling of target birth, death, merger, spawning, missed detections, false alarms, track initiation and maintenance are the scope of a related work and are not discussed here.

The rest of the paper is organized as follows. Section II discusses related prior work. Section III describes the network setting. Consensus tracking and Multitiling are discussed in Sections IV and V respectively. Simulation results are shown in Section VI. Conclusions are drawn in Section VII.

II. Related Prior Work

Many researchers have addressed the many different facets of distributed particle filtering. Liu, Chu and Reich [4] provide a survey of techniques for tracking multiple targets in distributed sensor networks. A good study on distributed target tracking is also provided in [5].

Fang, Zhao and Guibas [6] have discussed the problem of target enumeration and aggregation in sensor networks. Zhao, Liu, Liu, Guibas, and Reich [7] have considered leader-based tracking with mutual information based handing over of target to neighboring node, non-parametric storage of belief state and also a real-life implementation of distributed tracking. Coates [8] proposed two methods for reducing communication overheads: using factorization and using adaptive encoding. Särkkä, Vehtari, and Lampinen [9] proposed a Rao-Blackwellized Monte Carlo data association method in a centralized setting. The authors propose partitioning the problem into many single target tracking problem and solving the data association problem by sequential importance sampling.

Sheng, Hu, and Ramanathan [10] propose approximating the estimates with the parameters of a low-dimensional Gaussian Mixture Model (GMM). Zuo, Mehrotra, Varshney, and Mohan [11] also use GMM approximation of particles to convey belief to fusion center in bandwidth-efficient manner.

Ihler, Fisher, and Willsky [12] have considered the problem of approximating the density estimates using lossy compression techniques. Vercauteren, Guo, and Wang [13] have addressed the problem of joint tracking and classification of targets in sensor networks. Teng, Snoussi, and Richard [14] have proposed a distributed state-estimation algorithm that allows implicit compression of exchanged statistics between leader nodes.

Leader-based tracking is essential in WSN due to high correlation among signals picked up by neighboring nodes and for energy conservation. However, selecting leader node(s) as the target(s) move requires computation and communication overheads. The problem gets more complicated for multiple and unknown number of targets, and in the presence of clutter and missed detections. Realizing the importance of efficient networking infrastructure, lot of work has been carried out in this area (reference [15] provides a review); however, almost all of the 'tracking leader election' is independent of such structures. Also, while it is important to reduce communication cost, which is order of magnitude higher compared to computation cost, the energy consumption due to computation cannot be ignored, more so given the typically large number of particles in tracking problems. To the best of our knowledge, no work has been reported that addresses computation cost.

III. Network Setting

As discussed by Sheng, Hu, and Ramanathan [10], the assumption of independent observations at individual sensor nodes is unrealistic, and hence the motivation to partition the network into cliques. The clique size proposed in [10] is enough to cater to the spatio-temporal bandwidth of signals. As the target moves, newer cliques are formed. The authors argue that it results in unpredictable and time-varying size of clusters. In such cases, the cost of network management is typically very high and tends to dominate the operations cost. The cost of creating newer clusters every time results in additional overheads [16]. Most importantly, when multiple targets move in a given region, it is non-trivial, even impossible in some cases, to form distinct cliques for individual targets. Hence it is ideally suited that network be partitioned efficiently, not dependent on target movements, and sensor tasking be done using a robust distributed mechanism.

N. Trivedi and N. Balakrishnan have earlier proposed *iGroup* [16], a lightweight distributed algorithm to partition the sensor network into hexagonal cells and also to decide on unique communication channels and time slots for interference-free inter- and intra-cell communication. In communication infrastructure, the number of neighbors being bounded by six allows pre-determination of communication channels and time slots in a way that avoids interference. Figures 1 and 2 indicating the organization, channel usage and time slot allocation have been reproduced from [16] for reference.

The numbers inside cells in Figure 1 indicate channel indices. Many mote systems allow selection of channel from a set of tunable frequencies up to 25 in number, and the channel can be changed at run time simply by using a system call by providing the index as parameter. In Figure 2, interpretation of data can be determined by the control messages. The cell diameter is chosen such that the neighboring cells can communicate using the available power levels, and that the desired sensing coverage can be obtained when only the cell leaders are awake.

This paper demonstrates how this infrastructure can be exploited for efficiency in tracking problems. Though discussed in this setting, 'Consensus Tracking' can be used with other networking structures with ease.

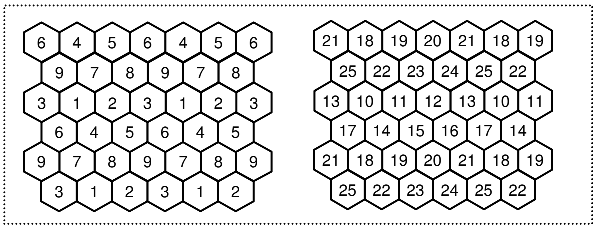


Figure 1. Intra-cell (left) and Inter-cell (right) channel usage patterns

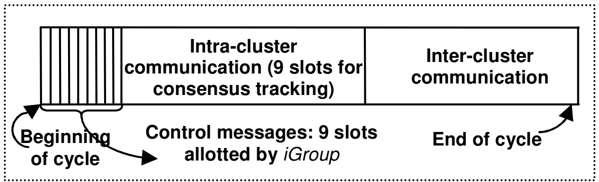

Intra-cluster communication (9 slots for consensus tracking)
Inter-cluster communication
Beginning of cycle
Control messages: 9 slots allotted by iGroup
End of cycle

Figure 2. Network Operation Cycle (modified for Consensus Tracking)

IV. Consensus Tracking

To exploit spatial diversity, multiple nodes observe a particular target; however, in a hierarchical infrastructure, it is the cell leader's responsibility to track target(s) using detection details from itself and the other cell members. Two neighboring cells may detect the same target(s) and hence it is required to decide which cells must track which targets.

The theme of Consensus Tracking is that the cell that is assumed to contain a given target must track it. Of course, the target position is never known in advance and hence a distributed mechanism is required by which all neighboring cells can uniquely agree on an assignment. This is achieved as follows.

All members belonging to the cell having intra-cell channel ID '*m*' broadcast their findings in slot '*m*' on channel '*m*' using transmission power enough to reach the neighboring cells. Random backoff is used to avoid collision. Having received the signals from own cell members and the neighbors', each cell generates 'energy plots' [5]. Figure 3 shows a sample energy plot. It is assumed that the sensor density is at least enough to capture the peaks and valleys of the energy field. The peaks in this plot indicate target positions. The cell to which the peak belongs is responsible for tracking that target. Since the same transmission is heard by all the neighbors for common targets and same algorithm used for generating energy plots, the peaks are common (error handling is discussed in the following subsection). The distributed algorithm for checking cell boundary [16] ensures that the cell assignment is also unique. Due to the properties of network infrastructure created by *iGroup*, neighbors more than one hop away need not participate in the coordination.

The assignment of targets to cells is followed by decision on whether to hand over the history. Specifically, if one cell was tracking a target that has just crossed its boundary, the current cell must hand over the time-series measurement data to the neighbor. The tracking process *per se* is out of scope of this paper; however, Section IV discusses computationally efficient methods towards distributed state-estimation process as well as towards preparing for handover.

A. *Effect of Message Errors*

If the presence of transmission or reception errors, the neighbors may end up with different peaks and sensor tasking may be ambiguous. However, the Bayesian tracking framework is robust enough to accommodate process and measurement noise and, as the further analysis and simulation results show, this fluctuation gets smoothened out in the tracking framework.

The framework for Consensus Tracking consists of an outer loop responsible for considering target births and deaths including clutter handling. Targets are identified by a grid-identification mechanism. When a target is detected for the first time, it is associated with a 'time-to-live' counter. If it was a false detection, this counter will eventually expire and the track will be deleted. Missed detections are handled in a similar manner by letting the tracks live for a 'time-to-live' counter. This mechanism helps combat occasional message errors as discussed below.

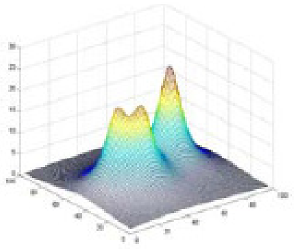


Figure 3. Sample energy plot [5]

Detection or transmission errors could lead to a target being detected in wrong position by all relevant neighboring cells. As a result, a wrong cell may begin tracking it. If the target was an existing one originally belonging to some other cell, the error would lead to missed detection there. Assuming that errors are spurious and non-identical over different time periods, the subsequent cycles will be able to recover from this error, where one cell assumes the phenomenon to be a false alarm and the other treats it as missed detection.

Reception errors could lead to generation of different peaks by neighboring cells and as a result, a target may be detected in two different positions by neighboring cells, potentially both of which may be wrong. Even this will be treated as false alarm and/or missed detections and subsequent cycles will allow recovery from this error as well.

A target moving on the cell boundary for some time could potentially cause fluctuating handovers. To avoid this, a cell hands over tracking to a neighbor only when the target is consistently in the other cell boundary for at least 3 cycles. The neighboring cell is informed not to initiate track during this period.

V. Multitiling

Computation steps in SMC techniques involve particle initialization, prediction, assigning importance weights based on measurement and sequential importance sampling to alleviate impoverishment. A large number of particles must be used if the tracking has to be accurate. This involves large number of computationally expensive floating point operations. Multitiling is a low-cost solution to this problem.

A. *Initialization*

In the absence of any prior knowledge, the initial measurement is best representation of the density for the belief state. The number of peaks is representative of the number of targets initially present in the system. Assuming typical sound source point target at location ζ with amplitude *a* and lossless, isotropic sound propagation model, the root-mean squared amplitude measurement z at location x is given by

$$z = \frac{a}{\| x - \zeta \|} + w \qquad \dots (3)$$

where *w* is measurement noise. The measurement function is discretized with desired resolution to generate particles.

B. *Prediction*

State transition is given by

$$x_{t+1}^i = g(\theta_{t+1}^i)x_t^i + w_t^i \qquad \dots (4)$$

where w is the process noise. The first prediction typically assumes a large variance for process noise for each state vector component. The variance gets reduced through the predict-measure-update cycle.

C. *Update*

Measurement is given by

$$z_t^i = h^i(\theta_t^i)x_t^i + v_t^i \qquad \dots (5)$$

where v is measurement noise. The distance between $z_t^i = h^i(\theta_t^i)x_t^i + v_t^i$ and $z'^i_{t|t-1} = h^i(\theta^i_{t|t-1})x'^i_{t|t-1}$ is the prediction error assuming $z'^i_{t|t-1}$ is what measurement would have been if state were $x'^i_{t|t-1}$ as predicted at time t-1.

Importance weights to be assigned to particles are inversely proportional to the 'distance' between predicted and actual measurements. In the cases where state vector involves non-spherically symmetric distributions, mere Euclidean distance is not a good metric. Distance measure that takes into account the variance of the variables is required. This can be achieved by scaling the variables by their 'variability'. Mahalanobis distance is one such measure that does not just take this variability into account but also caters to covariance between variables. Mahalanobis distance between two vectors $x = (x_1, x_2, ..., x_N)$ and $y = (y_1, y_2, ..., y_N)$ in $\Re^N$ is defined as

$$D = \sqrt{(x-y)'\Sigma^{-1}(x-y)} \qquad \dots (6)$$

where Σ is the covariance matrix of the distribution.

Mahalanobis distance must be calculated for each particle during every predict-update cycle for all the targets. The operation requires many floating-point multiplications. Given that the number of particles is typically very large, this is a highly computation-intensive step. In the following section the authors propose a low-cost approximation to this step. As the approximation progresses, side information is stored that helps in quick consolidation of state-estimate to be exchanged between cells either for belief handover or for distributed state estimate.

1) *Tile-based Weight Assignments*

The particles that are 'closer' to the measurement must be weighted higher. Points equidistant (in Mahalanobis sense) from the center form an ellipsoid around the center. A less computation-intensive approximation would be to consider cuboids enclosing the ellipsoid boundaries, because then the 'distance' could be calculated by only comparing the values with minimum and maximum values. This is the principle behind Multitiling. The region of interest in $\Re^N$ is divided into bigger cuboids enclosing smaller ones having common center point; this center point is the measured target state vector. Distance between the cuboids could be fixed for a fixed-interval comparison, or could increase with distance from the center for variable-intervals. Particles inside one cuboid share common weight.

Spatial data structures such as k-D tree are ideally suited for proximity-based searches even when state vector contains elements other than position [12]; however, the weight assignment process requires the entire particle set to be processed i.e., all the nodes in the tree must be visited. This results in undue space and time complexity compared to array-based representations. Nevertheless, the distance comparison concept in k-D tree is computationally efficient and is used in Multitiling for assigning weights. Multitiling treats concentric ellipsoids as 'bins', with particles in one bin considered to have equal weights. These bins approximated as cuboids allow simpler boundary comparisons.

a) *Statistically Independent State Variables*

When the covariance matrix is diagonal, i.e., the different state variables are statistically uncorrelated, the axes of the ellipsoids are parallel to the principle axes of $\Re^N$. Without loss of generalization, consider $\Re^2$. The area of ellipse is π**A***B* where *A* and *B* are the axes of the ellipse. The area of the enclosing rectangle is 4**A***B*. Scaling down the axes of the ellipse i.e., the bin sizes by 0.9 ensures that the rectangle covers desired area. Memberships get affected only for those particles on the boundary, which is an acceptable approximation considering the savings in computation. Figure 4(a) depicts this approximation called Multitiling-Z.

The test for a particle to lie within given cuboid boundaries involves simple order comparisons. Taking an example of 2-D Euclidean space, equidistant points form a circle and a low-cost approximation requires checking for (x < *center_x*+radius) and (y < *center_y*+radius). Extending this approximation to non-Euclidean distances means checking for (*ά* < *center_ά* + *ά_radius*) and (*έ* < *center_έ* + *έ_radius*), where *ά_radius* and *έ_radius* are the scaled radii of the ellipse corresponding to variances in *ά* and *έ* respectively.

The range for each variable in the state vector is divided into intervals that are multiples of σ_s (variance or spread) for that variable. In the present work, ranges are considered as multiples of 0.5σ_s for each variable.

b) *Correlated State Variables*

Some elements of the state vector may be correlated with each other, causing the concentric ellipsoids that depict 'scaled equal distances' to be inclined at a non-zero angle with respect to the principle axes in $\Re^N$. Mahalanobis distance calculation uses covariance for scaling. The inclined axes can be projected to the principle axes, effectively uncorrelating the elements; however, comparisons in this case will involve first projecting the particle vectors to the new coordinate system, involving costly operations. The following simplification by fitting ellipsoids to cuboids called Multitiling-N is proposed. $\Re^2$ is considered for illustration without loss of generality.

As a preprocessing step, a rotated rectangle is fitted to the ellipse (Figure 4(b)). It is required to compute the projection on each axis in $\Re^2$ only once; thereafter additions and subtractions can be used for finding the corner points of the rectangle. An acceptable quantization factor is decided, which is used for approximating the rectangle edges by stair cases. Even this operation uses only additions and subtractions. The resulting quantization effect is a good tradeoff towards the gain in speed and closeness of approximation. Figure 5 depicts the staircase approximation.

Having preprocessed the (quantized) edge points, Multitiling comparisons in each cycle amount to finding the staircase step in one direction followed by checking the boundary in the other direction. It can be easily seen that there are only two points on the horizontal axis for one step on the vertical axis. The number of comparisons during the update cycle is identical to that in the case of Multitiling-Z.

2) *Joint Tracking*

Multitarget tracking can be implemented as multiple independent particle sets or as multitarget particle systems, the latter being far more complex because it must deal with state sets rather than state vectors. The former case benefits directly from the approximations of the previous section. Multitiling concept applies equally well to the joint tracking case, though it can look a bit cumbersome.

It is convenient to order [1] $X^0_{k|k}, X^1_{k|k}, ..., X^{\upsilon}_{k|k}$ by increasing target number i.e., $w^0_{k|k}$ is the weight of the no-target particle $X^0_{k|k} = \phi$, the next υ_1 particles $X^0_{k|k}, X^1_{k|k}, ... X^{\upsilon_1}_{k|k}$ represent single-target samples $X^0_{k|k} = \{x^1_1\}, ..., X^{\upsilon_1}_{k|k} = \{x^{\upsilon_1}_1\}$ and, in general, the n-target states are represented by υ_n multitarget particles, up to some largest number ň of particles. Thus $\upsilon = 1 + \upsilon_1 + \upsilon_2 + ... + \upsilon_n$, where υ_0=1 is the number of zero-target particles.

Define $\theta(X)$ as $\theta(X)$ =1 if | X |=n and $\theta(X)$ =0 otherwise. Then, $f_{k|k}(n \mid Z^{(k)}) \cong \sum_{i=\upsilon_{n-1}+1}^{\upsilon_n} w^i_{k|k}$, where $f_{k|k}(n \mid Z^{(k)})$ is the cardinality distribution of $f_{k|k}(X \mid Z^{(k)})$ i.e., $f_{k|k}(n \mid Z^{(k)})$ is the probability that n targets are present in the scene [1].

Multitiling addresses joint tracking as follows. The number of peaks is counted and number of targets *n* estimated from there taking into account the probabilities of missed detection and false alarms. The sets having *n* targets are accordingly assigned weights. Probabilities of missed detection and false alarms indicate possible variance, and the sets having *n-l* to *n+l* targets are assigned reducing weights accordingly.

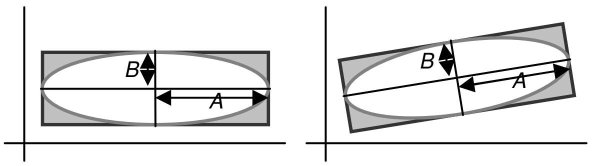


Figure 4. The distribution 'ellipse' for 2-D state vector when the two dimensions are (a) independent (Multitiling-Z) (b) having non-zero correlation (Multitiling-N)

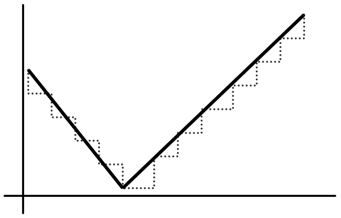


Figure 5. Quantizing rectangle edges for Multitiling-N

Mahalanobis distance must now be calculated for each particle in the set for permutations of locations. The weights assigned to the cardinality multiplied by the weights to the particle set by Multitiling are the net weights for the sets.

3) *Computational Complexity*

The computation complexity of Multitiling is order of magnitude less compared to direct method of distance calculation for the same number of state variables. Mahalanobis distance is calculated as $D = \sqrt{(x-y)'\Sigma^{-1}(x-y)}$. Even if we were to ignore the square root (only ordering is needed, not the absolute distance) and the inverse of Σ (assuming known covariance matrix, it is enough to calculate the inverse once), it requires O(N) floating-point subtractions and O(N^2) floating-point multiplications.

In contrast, Multitiling-Z requires O(N) floating-point comparisons only. Multitiling-N additionally requires preprocessing. Since the distribution covariance is known, the computation-intensive operations such as fitting the cuboids can be done offline and fed to the sensor nodes.

In the O(N) comparisons, the constant factor depends on the number and spread of bins (Б) and, in the case of Multitiling-N, the desired quantization factor (Ḱ). Assuming that Б<<ν (number of particles) and the number of steps due to Ḱ<<ν, the cost of binning and quantization does not dominate. This is huge saving considering the typical number of particles to be handled in each cycle of real-time operations. The saving is even more significant in the case of joint tracking, where distances for many permutations of target states must be computed.

4) *Resampling*

Since the particles have unequal weights, they must be replaced by new particles having equal weights while retaining the influence of weights. The weights assigned by Multitiling are analogous to those assigned by existing methods and hence sequential importance sampling for Multitiling can be performed using common methods. This paper uses multinomial resampling. Assuming ν particles and $\bar{w}_i \equiv w^i_{k+1|k+1}$ for all i=1…ν, the multinomial distribution $\mu(i_1, ... i_\nu) = \frac{\nu!}{i_1! ... i_\nu!} . \bar{w}_1^{i_1} ... \bar{w}_\nu^{i_\nu}$ is a probability distribution on all ν-tuples (i_1, …, i_ν) of nonnegative integers for which i_1 + …+ i_ν = ν. A random sample (e_1, …, e_ν)~μ(.) is drawn from this distribution. If e_i=0, then the particle $x^i_{k+1|k+1}$ is eliminated, else e_i identical copies are made of this particle. Since $e_1 + \ldots + e_\nu = \nu$, total number of particles stays as ν.

To avoid particle impoverishment, the copies of particles are randomly jittered.

5) Belief Handover

Belief propagation always requires more bits than raw data and hence is extremely expensive. Gaussian Mixture Model (GMM) approximation is an excellent way to save the transmission cost [10][11]. However, there usually is no prior knowledge of the number of components 'c' in the mixture. The μs and σs are also not known in advance. Reference [10] suggests selecting c as a function of number of targets and learning the GMM parameters using an iterating Expectation-Maximization (EM) algorithm. Reference [11] picks up c on a higher side subject to a maximum number, merges two components if the Kullback-Lieber distance between them is below a threshold, and applies EM algorithm.

De facto method for initialization of EM has been to assign random values to the parameters. However, EM is guaranteed only to converge to local optimum in the likelihood and hence it is best to initialize the model in the region of likelihood space where the local maxima are supposed to lie. Multitiling provides a crucial cue to c, μ, and σ, arguably the closest approximation for EM iterations, thereby saving precious iteration cost and also resulting in more accurate mixture parameters. The cue is provided as follows.

A mode is a local maximum and is represented by higher frequency of particles. In Multitiling, finding local maxima amounts to counting the number of particles in each bin. Multitiling divides the particles into bins having equal 'weighted' distance from the estimated state; however, selection of mode requires both range and bearing in the multidimensional space. First, the bins corresponding to local maxima are identified based on first-order differences; threshold is applied to avoid spurious small maxima. The number of maxima thus found is indicative of c. One random particle each from these bins is assumed to be the mean 'μ' for the mixture component. The variance 'σ' is estimated based on the first order difference.

This process does not take into account the possibility of multiple modes for a single target in a single bin e.g., equal probability of a target taking a left or a right turn, indicated by two modes on the opposite sides. To address this, Multitiling maintains another variable called 'quadrant' for each particle. Orthogonal hyperplanes in $\Re^N$ divide the cuboids into four 'quadrants'. No additional cost is needed to find this quarter through Multitiling process. Binning is done separately for the quadrants. Modes closer than this are considered as data fluctuations and are filtered out during thresholding.

VI. Simulation

An area of 100mx100m is considered with 400 nodes forming the *iGroup* infrastructure, and a target is moved through the area.

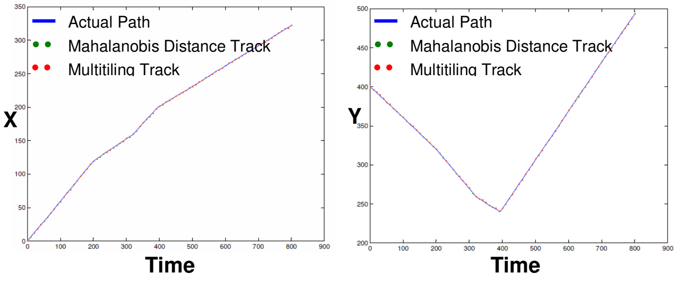


Figure 6. Tracking using Multitiling-Z

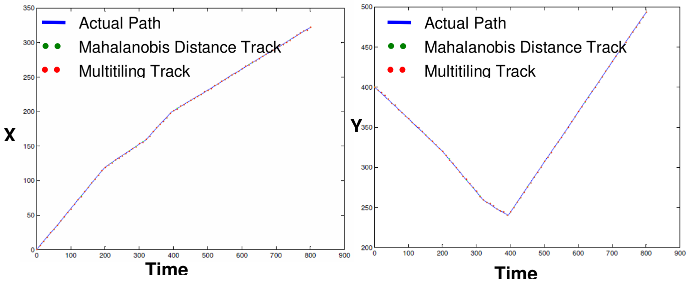


Figure 7. Tracking using Multitiling-Z: Zoomed-in version

A. Consensus Tracking

The metrics considered are multiple tasking (assigning one target to 2 or more cells), incorrect tasking and handover fluctuations (at least once). Results are averaged over 20 simulation runs. Parameters as given in Table-I are assumed, where P_d is probability of detection and λ is probability of false alarm. Results are shown in Table-II. Note that due to sensors erring in estimating their own location, a target in one cell could appear to be in another cell. This is not considered to be an error in target assignment.

TABLE I. Parameters for Consensus Tracking

Scenario	1	2	3	4	5	6
P_d	1	0.9	1	1	1	0.9
λ	0	0	0.001	0	0	0.001
Transmission error	0	0	0	0.01	0	0.01
Reception Error	0	0	0	0	0.01	0.01

TABLE II. Consensus Tracking: Results

Scenario	1	2	3	4	5	6
Multiple Tasking	0%	0%	0%	0%	<1%	<5%
Incorrect Tasking	0%	0%	0%	0%	<1%	<5%
Handover Fluctuations	0%	0%	0%	0%	0%	0%
Recovery after Error	NA	100%	100%	100%	100%	100%

B. Multitiling-Z

Figure 6 shows the actual (with process noise), Mahalanobis-estimated, and Multitiling-Z-estimated X and Y positions. Figure 7 shows a smaller sector zoomed in for clarity. Results were averaged over 20 simulation runs.

C. *Multitiling-N*

A hypothetical ellipse like the one in Figure 4 (b) is considered with variances and correlations as listed in Table-III. Monte Carlo simulations were used to generate data points with these statistical parameters; these points represent particles. State estimates were computed using Mahalanobis distance and Multitiling-N approximation with various quantization levels. The estimation errors in both cases are depicted on the scatter plot of Figure 8.

TABLE III. PARAMETERS CONSIDERED FOR MULTITILING-N

σ_x^2	σ_y^2	σ_{xy} (Covar)	$\rho_{x,y}$ (Correlation)
16	25	[10, -7, 0, 0.5]	[0.5, -0.35, 0, 0.025]
25	9	[-8, 13, 0, 0.9]	[-0.53, 0.87, 0, 0.06]
64	49	[50, -7, 0, 0.1]	[0.89, 0.125, 0, 0.0018]
49	49	[24, -12, 0, 0.8]	[0.49, -0.25, 0, 0.016]

VII. CONCLUSION AND FUTURE WORK

Using the strength of the Bayesian technique atop the foundation of a robust networking infrastructure helps build a robust distributed tracking framework. Use of Multitiling provides the additional cost-saving.

Results in Table-II will change for multi-target scenario, especially when targets move in close proximity. Handling target energy overlap is the subject of a related work and hence not discussed here; however, multi-target tracking can only build out of robust single target tracking.

Multitiling provides excellent approximation for Mahalanobis distance, and at a substantially reduced cost. Even in the case of Multitiling-N (Figure 8), the worst-case estimation error is less than 3 units (most of the worst-case numbers belong to the cases of large variances and coarse quantization), whereas most other errors are confined to ±0.5 units, similar to what Mahalanobis distance based estimates provided. The framework makes a low-cost tracking model.

Our ongoing work is on implementation of EM and using it with data generated by Multitiling to demonstrate increase in performance and reduction in iterations.

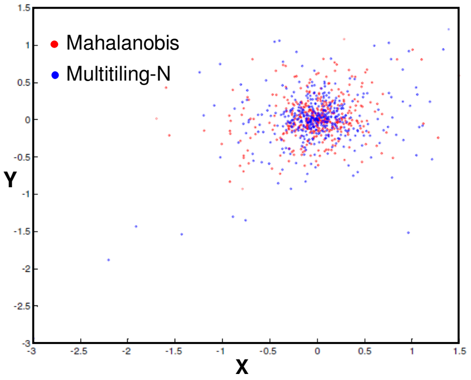


Figure 8. State Estimation Errors using Mahalanobis Distance and Multitiling-N

REFERENCES

[1] R. P. S. Mehlar, Statistical Multisource-multitarget Information Fusion. Artech House, Norwood, MA: 2007, ISBN 13: 978-1-59693-092-6

[2] K. P. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning", PhD Thesis University of California, Berkeley, Fall 2002

[3] N. Gordon, D. Salmond, and A. F. M. Smith, "Novel approach to nonlinear and non-Gaussian Bayesian state estimation", in Proc IEEE, vol. 140, pp. 107-113, 1993

[4] J. Liu, M. Chu, and J. E. Reich, "Multitarget Tracking in Distributed Sensor Networks", IEEE Signal Processing Magazine, Vol. 24, Issue 3, May 2007, pp 36-46, doi 10.1109/MSP.2007.361600

[5] M. E. Liggins, C.-Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos, "Distributed Fusion Architectures and Algorithms for Target Tracking", Proc. IEEE, vol. 85, Issue 1. pp. 95-107, Jan 1997, doi: 10.1109/JPROC.1997.554211.

[6] Q. Fang, F. Zhao, and L. Guibas, "Lightweight Sensing and Enumeration Protocols for Target Enumeration and Aggregation", In Proc. 4th ACM Intl Symp Mobile Ad Hoc Networking and Computing (MobiHoc'03), 2003, doi: 10.1145/778415.778436

[7] F. Zhao, J. Liu, J. Liu, L. Guibas, and J. Reich, "Collaborative Signal and Information Processing: An Information Directed Approach", Proc. IEEE, Aug 2003, Vol. 91, Issue 8, pp 1199-1209, doi: 10.1109/JPROC.2003.814921

[8] M. Coates, "Distributed Particle Filters for Sensor Networks", In Proc. 3rd Intl Symp Information Processing in Sensor Networks (ISPN'04), 2004, pp. , doi: 10.1145/984622.984637

[9] S. Särkkä, A. Vehtari, and J. Lampinen, "Rao-Blackwellized Monte Carlo Data Association for Multiple Target Tracking", In Proc. 7th Intl Conf. Information Fusion, July 2004, pp 583-590

[10] X. Scheng, Y.-H. Hu, and P. Ramanathan, "Distributed Particle Filter with GMM Approximation for Multiple Target Localization and Tracking in Wireless Sensor Networks", In Proc. 4th Intl Symp Information Processing in Sensor Networks, Apr 2005, pp 181-188, doi: 10.1109/IPSN.2005.1440923

[11] L. Zuo, K. Mehrotra, P. K. Varshney, and C. K. Mohan, "Bandwidth-efficient Target Tracking in Distributed Sensor Networks using Particle Filters", In Proc 9th Intl Conf Information Fusion, July 2006, pp 1-4, doi: 10.1109/ICIF.2006.301692

[12] A. T. Ihler, J. W. Fisher, and A. S. Willsky, "Particle Filtering under Communication Constraints", In Proc. 13th IEEE/SP Intl. Workshop Statistical Signal Processing, July 2005, pp. 89-94, doi: 10.1109/SSP.2005.1628570

[13] T. Vercauteren, D. Gou, and X. Wang, "Joint Multiple Target Tracking and Classification in Commaborative Sensor Networks", IEEE J. Selected Areas in Communications, Vol. 23, Issue 4, pp. 714-723, Apr 2005, doi: 10.1109/JSAC.2005.843540

[14] J. Teng, H. Snoussi, and C. Richard, "Binary Variational Filtering for Target Tracking in Sensor Networks", In Proc. 14th IEEE/SP Intl. Workshop Statistical Signal Processing, Aug 2007, doi: 10.1109/SSP.2007.4301346

[15] N. Trivedi, S. S. Iyengar, and N. Balakrishnan, "Ripples: Message-efficient, Coverage-aware Clustering in Wireless Sensor and Actor Networks", Intl J. Communication Networks and Distributed Systems", Vol. 2, Issue 1, Jan 2009, doi: 10.1504/IJCNDS.2009.021697

[16] N. Trivedi, S. S. Iyengar, and N. Balakrishnan, "Efficient Multiplexing for Multichannel Data Dissemination with Delay Guarantees in Wireless Sensor Networks", In Proc. 2nd Intl Conf. Sensor Technologies and Applications (SENSORCOMM-08), Aug 2008, pp 23-29, doi: 10.1109:/SENSORCOMM.2008.109

Live Data Acquisition for Situation Awareness in Traffic Management Systems Using Laser Sensors

Armin Veichtlbauer, Peter Dorfinger, Ulrich Schrittesser
Advanced Networking Center
Salzburg Research Forschungsgesellschaft m.b.H.
Salzburg, Austria
{*firstname.lastname*}*@salzburgresearch.at*

***Abstract*—Traffic management systems help to organise the more and more dense traffic in cities. However, the classic approach for such systems is the use of traffic models, which incorporate a long-term knowledge of traffic situations in the respective city. This approach is not adequate for reacting to changes in traffic, either long-term changes, which have not been trained to the model, or short-term changes due to unexpected events. In this paper, we describe how we collect real-time traffic data using a laser scanner, evaluate the precision of the traffic measurements and assess the circumstances under which the results are useable for a professional traffic management system.**

***Keywords*-Situation Awareness, Sensor Data Acquisition, Vehicle Counting, Traffic Management.**

I. Introduction

In modern cities traffic is becoming a topic of more and more controversial discussion: On the one hand the need for transportation of people and goods from place A to place B is still increasing [1], on the other hand this causes severe problems, especially in cities with dense population (as the most European cities are): congestion, pollution, noise, etc. [2]. Traffic management systems are used to optimise the traffic according to pre-defined metrics. This is done for instance by setting the time periods for green and red phases of traffic lights.

Yet in most cases, traffic management systems are based on static traffic models only [3], i.e., static data that provide statistical information about characteristic parameters like traffic density in an area, throughput of an intersection, rates for different directions, etc. These information are provided for several situations (e.g., rush hour, night traffic, holidays, etc.), and for each of these situations some optimisation calculations can be performed.

For many operating traffic management systems these parameters are captured manually, e.g., by a person counting cars passing a certain position [4]. This is an unfavourable situation: First, the correctness of the data cannot be validated. Second, in order to integrate long-term traffic changes, the counting has to be re-done in regular intervals, and the model has to be optimised again in order to adapt to the changes of the regarded traffic situations.

However the knowledge of the current distribution of these parameters does not compensate for a situation awareness including live traffic data. To be able to include unexpected events like accidents or short-time changes like the opening of a new shopping mall into the traffic management system, a real-time capable solution for sensing the current traffic situation is an indispensable precondition. A dynamic traffic management system is able to react to every change detected, i.e., instead of making offline optimisations based on measured traffic data, the traffic management system reacts in real-time to the traffic measurements, thus setting up a closed-loop control [5]. This is called a *dynamic traffic management* system [3].

Such a system consists of the following logical distinguished parts:

- The sensing part: Several traffic parameters of interest for the traffic management have to be measured, e.g., the number of vehicles passing per lane and per time unit, the ratio of vehicles driving to the outgoing directions of an intersection, the average time a vehicle needs to drive from reference point A to reference point B, the average pollution caused by vehicles, etc.
- The communication part: The sensed values have to be collected in timely manner (with timing constraints). Thus the underlying communication network, which is used for transmitting the measured values to the control part, has to be dependable [6].
- The control part: Dynamic traffic management systems use the sensed values under real-time conditions for the respective traffic management tasks (e.g., control traffic lights or dynamic speed limits) according to existing rules [3].
- The actuator part: To influence the traffic in order to optimise the relevant traffic parameters according to a defined strategy, the traffic has to be controlled by suitable actuators. Here, the traffic lights are the easiest way to influence traffic in cities; dynamic speed limit signs are used on highways; but also *weak actuators* can be taken into account [7], like signs with the number of free parking slots in city centres, etc.

During the project sTC-net [8] we worked on a prototypical solution of a dynamic traffic management system. The part of our group was the setup of a sensor system and the collection of sensor data, which are usable as input for a dynamic control system. This machine learning based control system was set up by a partner company.

In this paper, we describe the traffic sensing approach we used, based on a single row laser scanner system. First we give a brief overview of the state-of-the-art of scientific work in this area. In the following section we describe the architecture of the system we used to measure and collect data (hardware, software, communication infrastructure). The next section shows the traffic counting algorithm, which was implemented in Matlab [9], and how we validated the output of that algorithm. This is followed by an overview of the measurements we conducted with this system and the results of the measurements. After that, we present some conclusions that can be derived. Finally, we give an outlook at possible further research and development activities.

II. Related Work

There are several approaches for traffic counting and for sensing other traffic parameters like average speed etc.: Inductive loops; magnetic, piezo-electric and pneumatic sensors; microwave, infrared and ultrasonic radars; optical systems and floating car data (FCD).

For traffic counting, laser scanners are a popular alternative, especially overhead mounted at highways [10]. Zhao et.al. [11] propose to use horizontally mounted single row laser scanners to monitor the traffic at an intersection in a city. The applicability of such an approach to our scenario is limited, as a mounting at a height of less than one meter has two main disadvantages: First, as the lasers can easily be reached, they are potentially subject to vandalism. Second, pedestrians waiting at a position near the laser devices are “blocking” a large angle of the laser scanners.

Regarding traffic management systems, there are approaches to integrate live sensor data into dynamic traffic management systems [3]. Such systems could also exploit the sensor information to adopt their rule base, i.e., the underlying traffic model, by using machine learning methods (e.g., artificial neural networks [12], or genetic algorithms [13]).

For communication purposes, e.g., collecting the sensed data, wireless networks provide an easy to deploy solution [14].

III. System Architecture

Existing traffic counting solutions are using a large number of laser sensors. In typical installations [10], one laser sensor is installed above each lane in order to count the number of vehicles on this lane. In cities, it is often not possible to install a metal carrier above the lanes due to limited space. Consequently installing one laser per lane is not feasible. Besides that, for larger intersections, e.g., with 12 individual lanes (3 lanes per direction), installing 12 laser sensors is a cost factor that operators are not willing to pay in many cases.

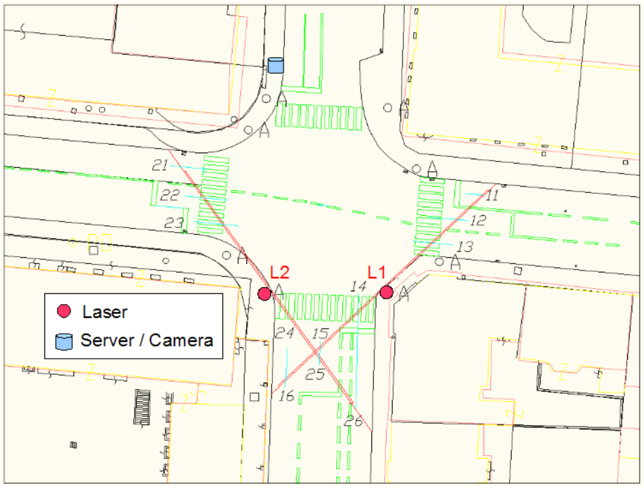


Figure 1: Intersection Salzburg Lehen

Our approach had the goal to sense the traffic on an intersection using a minimum number of laser devices and to setup an installation with minimal costs, being accurate enough to be used as input for a traffic management system. According to the requirements of the traffic management system of our partner a target value of 90% [8] for precision and recall [15] is sufficient. Here, precision is the ratio of correctly counted vehicles compared to all counts; recall is the ratio of correctly counted vehicles to all vehicles.

Figure 1 shows our installation at the intersection Ignaz-Harrer-Str./Rudolf-Biebl-Str. in Salzburg Lehen. Two lasers have been installed to count the vehicles to and from south and partially vehicles from west (Lane 23) and to east (Lane 13). For sensing the complete traffic on all lanes at the chosen intersection, four lasers would be required. The lanes are numbered according to the following scheme: The first digit is the number of the laser, and the second digit is a running index. Lanes may be scanned by two lasers, having two identification numbers then, e.g., Lane 24 = Lane 16.

The lasers were mounted on existing traffic light masts in order to minimize the effort and costs for mounting. As a consequence the position of the masts is not optimal for the used lasers. Thus we expected to get a good estimation under which circumstances (e.g., light, weather conditions) the measurements fulfil the required precision and recall.

For validating the conducted measurements a camera was installed. The camera’s resolution does not allow for identifying the plates, but is sufficient for counting.

Figure 2 visualizes the measurement architecture at the chosen site. It consists of two lasers, three router boards and one server:

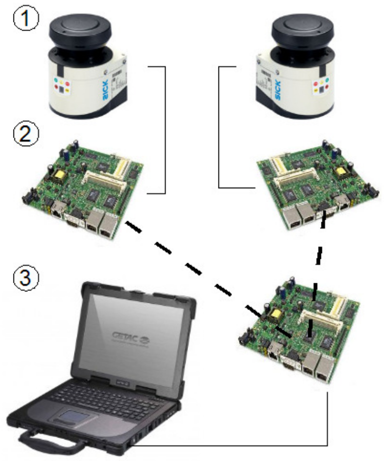


Figure 2: Measurement architecture

1) Laser: The LMS111 [16] is a single row type laser scanner from the company Sick with a high profiling rate of 50 Hz and wide viewing angle of 270 degrees. The LMS111 measures the distance to the reflecting object for each laser beam. On black background, this laser scanner has a maximum range of 18 m. The stepsize between two laser beams is 0.5 degrees; thus a single scan consists of 540 values. The last scan is stored locally and can be transmitted to the server upon request.
2) Routerboard: The Mikrotik Router Board 532A [17] is a fully integrated communication platform for high-speed, Ethernet and wireless networks with the Linux based Mikrotik Router Operating System. It offers the use of the R52 wireless module, which operates in 2.4 GHz and 5 GHz ranges.
3) Server: The server is a robust IP54 laptop with dual core and Win XP operating system. It is placed inside the intersection's main control cabinet. An application requests the laser measurement data in regular intervals and relays the collected data to a MySQL database on the laptop.

The lasers and the server are connected to the Routerboards via Ethernet cables. In order not to influence any other radio frequencies, the communication between the Routerboards is done in the 5 GHz frequency range. The embedded boards are installed in a splash-proof enclosure mounted in the proximity of the sensors on the respective traffic light masts.

IV. Algorithm

Our system requires knowledge about the individual geometry at the intersection. The mounting heights of the lasers as well as the horizontal distance from the laser to the borders of the individual lanes have to be known. A further step needed in initialization is to measure the ground, as the shape of the ground is not necessarily even. Unevenness or inclination of the ground influence the measurements and have to be identified in advance.

This is done best by simply sensing the empty intersection: The laser measures the distances to the road surface itself without any cars present. For places further away than 18 m from the sensor, we are not able to detect the road surface any more. We are using linear extrapolation of the detected surface then. This extrapolation only works for almost horizontal intersections.

Our newly developed vehicle counting algorithm itself consists of three major steps:

1) Pre-process the data for each lane
2) Count vehicles for each lane
3) Combine information from different lanes

Pre-processing the data sets is done as follows: Values, which are further away than the current lane of interest, are set to ground. Values, which are closer than the current lane of interest are replaced by "NaN" (the Matlab value "Not a Number") [9], as it is uncertain whether there is a vehicle behind this obstacle or not.

For the vehicle counting algorithm, the pre-processed data is used for each lane of interest. For each measurement value (which is representing the distance from the laser scanner to the reflection point) the height of the corresponding obstacle is derived, using a trigonometric calculation (cosine function) with the current angle of the laser beam and the height of the laser sensor as inputs.

Figure 3 shows 200 consecutive scans on a specific lane of interest. The color scheme changes from dark red for maximum distance to dark blue for minimum distance. White fields indicate that no value has been retrieved (NaN). The figure shows three cars passing the lane, where especially for Car 1 and Car 3 only a small number of measurement values have been retrieved. For the measurements on this lane an angle of 22 degrees is of interest; thus we are concentrating on the respective 45 scan points.

Due to the plain surface of cars and the acute measurement angle of the objects, many measurements do not produce correct values (NaN). As a consequence the algorithm has to detect a car even if it is represented by NaNs and not only by respective measurement values (i.e., smaller distances between laser and reflection point than for the ground). This can be observed for instance for Car 2.

Figure 4 shows same plot as Figure 3, but after determination of the height (above ground) of each individual scan point.

To deal with the NaN values a linear interpolation is used. The results of such an interpolation for the above example are shown in Figure 5. These interpolated values are used for classification.

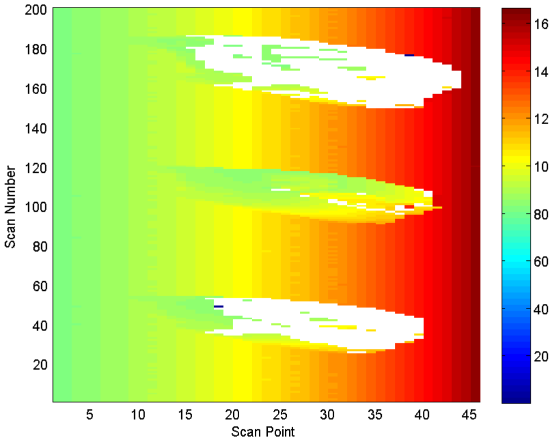


Figure 3: Representation of three cars (distance)

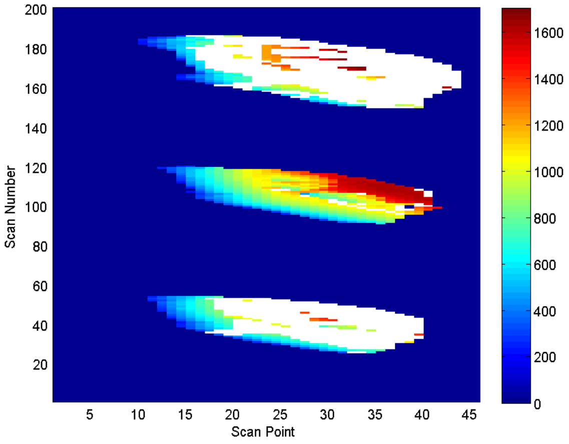


Figure 4: Representation of three cars (height)

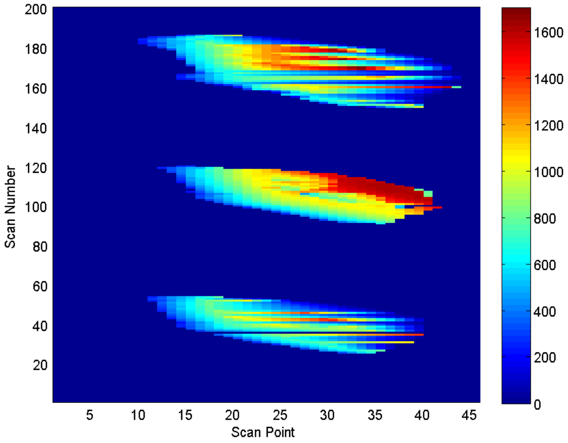


Figure 5: Representation of three cars (interpolated)

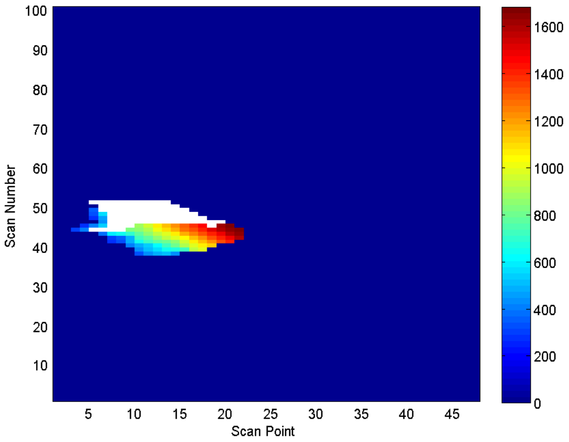


Figure 6: Representation of a pedestrian

In comparison to a car, a pedestrian leads to a smaller representation, as indicated in Figure 6.

A scan is classified in one of three categories:

- "Car Scan": This scan provides an indication for a vehicle
- "Clear Scan": The measurement reflects that there is no vehicle present on the lane
- "Unsure Scan": For this scan it cannot be determined whether a vehicle is present or not

A scan is classified as Car Scan, if two characteristics are met: First there has to be at least one block (of a pre-defined length) of (consecutive) values greater than zero. Second the average height within this block has to exceed the *MinHeight* threshold.

For classification as Clear Scan two other characteristics have to be met: First the average height has to be below the *NoiseHeight* threshold or the average is based on less than 5 values (the rest is interpolated). Second the NaN fraction (number of NaN values divided by the number of scan points for this lane) has to be below *MaxNaNtoClear*.

An Unsure Scan is one that falls into none of the above categories.

In our prototypical implementation the *MinHeight* threshold is set to 250 mm and the *NoiseHeight* threshold is set 100 mm. The value of *MaxNaNtoClear* depends on the lane and is configured between 0.2 and 0.9.

To count a detection as car, three Car Scans without a Clear Scan in between have to be present. Each Car Scan increases the *CarIndicationCount* whereas a Clear Scan resets this variable to 0. An Unsure Scan does not change the *CarIndicationCount*.

For traffic management systems it is important to consider the fraction of large vehicles like trucks or buses compared to the total number of vehicles. Thus our algorithm adds a further category for the scans, which is based on the maximum height value within one lane: A Truck Scan.

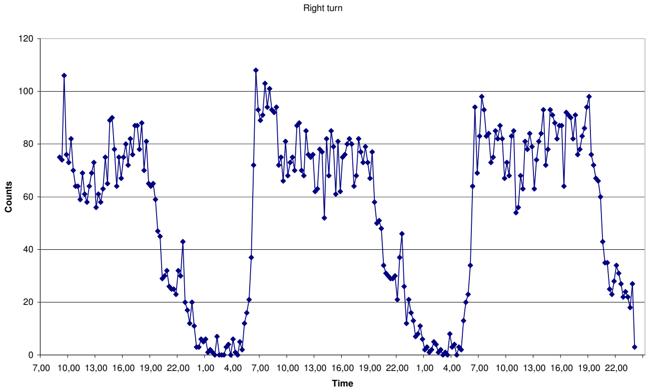


Figure 7: Right-Turn Lane 23 to 24

MEAS.	LANE 23		LANE 24		RIGHT TURN	
	PREC.	RECALL	PREC.	RECALL	PREC.	RECALL
M1	99.1	100	98.8	100	100	100
M2	100	100	100	100	100	99.1
M3	100	100	97.3	98.6	100	100
M4	97.7	100	97.0	98.8	100	99.2

Table I: Results for different conditions

	LANE 25A	LANE 15B	LANE 14	LANE 13	RIGHT TURN
PREC.	98.3	98.9	100	91.1	100
RECALL	96.7	98.9	100	99.5	98.5

Table II: Results for different lanes

This is a subcategory of a Car Scan, where additionally the maximum height exceeds 2750 mm. If we count five scans as Truck Scan (using the variable *TruckIndicationCount*), without a Clear Scan in between, the vehicle is classified as truck or bus; otherwise it is assumed to be a car.

After the counts of the separate lanes have been finished, the measurements can be combined in order to improve the rate of correctly measured turns. Lane 23 for instance is solely intended for right turns only; nevertheless it occurs that cars on this lane go straight ahead. To correct the counting for such (misbehaving) vehicles, we combine counts on Lane 23 and Lane 24: If within Z seconds after a vehicle was counted on Lane 23 there is no count on Lane 24, this item is removed from the number of right turning vehicles, as it probably was (illicitly) moving straight ahead. The parameter Z has to be adjusted according to the intersection's geometry.

Furthermore the validation process has shown that trucks and buses, which perform a right turn from Lane 23 to Lane 24 trend to sheer out to Lane 25, thus making a passing of two trucks/buses at the same time impossible. Thus whenever a truck/bus is counted on Lane 24, a truck/bus count from Lane 25 within a given time range is removed.

An important design goal was that the ability of the algorithm to be able to operate live, with a delay small enough to enable the traffic control in time. Thus we decided that the counting algorithm should basically operate on single scans. As mentioned, it only uses two status variables (*CarIndicationCount* and *TruckIndicationCount*) to proceed information between different scans. In practice, for traffic management systems a typically used counting granularity is 15 min.

V. MEASUREMENTS

We have performed a number of measurements on the intersection shown in Figure 1. The measurements were performed under different environmental influences: We have taken into account different weather conditions, day and night, as well as high and low load situations.

We performed 4 different measurements with a length of about 20 min each, and analysed the classification results compared with a manual counting based on the camera movie. Measurement M1 was performed at a high load situation in the morning with good weather conditions, M2 at a high load situation during snowfall, M3 at a low load situation in the night and M4 at a high load situation with sun in the afternoon. Furthermore we conducted a long-term measurement for 3 consecutive days. For the latter, Figure 7 shows the number of counted right turns.

As evaluation metrics, we use precision and recall [15]. Table I shows precision and recall for the counts on Lane 23 and Lane 24 as well as for the number of right turns. Precision and recall are close to 100% and comply with the requirement to exceed 90%.

We further evaluated the ability to count traffic on further lanes including interior lanes (e.g., Lane 15/25) where it is likely that an object on an exterior lane influences the measurement. Table II shows the results of these lanes based on Measurement M4.

These results show that also traffic on interior lanes can be counted with the needed precision and recall. Yet the values are lower due to influences of exterior lanes. Lane 15 resp. 25 is a split lane where the left part is used for left turns, and the right part to go straight ahead. Thus we split this lane also for our counting algorithm. We used the respective nearer lasers: Laser 2 for the left part (Lane 25A), Laser 1 for the right part (Lane 15B).

We observed a lower precision (yet still above 90%) for Lane 13: This is due to the mounting position of the laser; the laser beam crosses the cross-walk in the region of interest. It may occur that pedestrians or groups of pedestrians are counted as cars. This problem increases with the distance between the laser and the area of interest, as the angle gets more acute and thus laser beams reach the side of the car/pedestrian and not the roof of a car. However, these wrong counts for the pedestrians can be easily removed by post processing, e.g., by mapping incoming and outgoing traffic together (like it was performed for Lanes 23 and 24) or by simply taking into account the current phase of the traffic lights.

For interior lanes, which are further away from the laser devices, the counting algorithm does not obtain the required precision and recall of 90%; thus for counting Lane 11, 12, or 22 further lasers would have to be installed. The reason for these limits of the applicability of the laser devices is, that (besides the limited distance the laser is able to measure) the angle between laser beam and moving object gets too acute to obtain enough correct measurement values.

VI. Conclusion

We have shown the ability and the potential of the examined technology to work as a traffic sensor, measuring the number of two different categories of vehicles (cars vs. buses or trucks) passing a control point in a defined measurement period. The required measurement precision and recall (90%) have been reached, yet both values turned out to be very much dependent on the angle between laser and street surface and on the distance between device and the reflecting point.

VII. Further Work

An automatic adaptation to different geometries of intersections would help to reduce the effort of installing sensor systems at new intersections. This has potential to be researched in follow-up projects.

Another unsolved question is the integration of sensors and actuators from third-party providers by generating a generic interface in order to enhance the situation awareness of the traffic management system and to enable alternative control mechanisms to red lights and speed limits. For instance the integration of FCD could help to identify regions of high traffic density and slow speed. If drivers get informed of the current traffic situation this could enable them to avoid these regions for the time being.

Acknowledgment

The work described in this paper was conducted during the project ”Subsidiary Traffic Control Network” (sTC-net) funded by the Austrian Federal Ministry for Transport, Innovation and Technology.

References

[1] The Austrian Federal Ministry for Traffic, Innivation and Technology (BMVIT), “Klug investieren, verantwortungsvoll sparen - Ausbauplan Bundesverkehrsinfrastruktur 2011–2016,” 2009.

[2] M. Finkelstein, M. Jerrett, and M. Sears, “Traffic Air Pollution and Mortality Rate Advancement Periods,” in *American Journal of Epidemiology, pp. 173-177*, 2004.

[3] Q. Yang, “A Simulation Laboratory for Evaluation of Dynamic Traffic Management Systems,” Doctoral Thesis, Massachusetts Institute of Technology, June 1997.

[4] Ad Hoc News. (2010, April) Verkehrszaehlung 2010 hat begonnen. Accessed: 2011-03-30. [Online]. Available: http://www.ad-hoc-news.de/verkehrszaehlung-2010-hat-begonnen--/de/News/21208522

[5] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, “Review of road traffic control strategies,” in *Proceedings of the IEEE., 91 (12), pp. 2043-2067*, 2003.

[6] G. Panholzer, A. Veichtlbauer, P. Dorfinger, and U. Schrittesser, “Simulation of a Robust Communication Protocol for Sensor Data Acquisition,” in *Proceedings of the Sixth International Conference on Wireless and Mobile Communications (ICWMC 2010), pp. 145-150*, September 2010.

[7] M. Vasirani and S. Ossowski, “Market-based coordination for intersection control,” in *Proceedings of the 2009 ACM symposium on Applied Computing, pp. 747-751*. ACM Digital Library, 2009.

[8] Salzburg Research Forschungsg.m.b.H. (2011) sTC-net – Intelligente Sensoren im Verkehrsmanagement. Accessed: 2011-03-23. [Online]. Available: http://www.salzburgresearch.at/projekt/stc-net/

[9] The MathWorks, Inc. (2011) MATLAB - The Language Of Technical Computing. Accessed: 2011-03-23. [Online]. Available: http://www.mathworks.com/products/matlab/

[10] Road Traffic Technology. (2011) AutoSense Truck Body Classifier. Accessed: 2011-03-31. [Online]. Available: http://www.roadtraffic-technology.com/contractors/detection/osi/osi3.html

[11] H. Zhao, J. Cui, H. Zha, K. Katabira, X. Shao, and R. Shibasaki, “Monitoring an intersection using a network of laser scanners,” in *Proceedings of IEEE Int. Conf. on Intelligent Transportation Systems (ITSC 08), pp. 428-433*, 2008.

[12] K. Dresner and P. Stone, “Multiagent Traffic Management: An Improved Intersection Control Mechanism,” in *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems 2005 (AAMAS 2005), pp.471-477*. ACM Digital Library, July 2005.

[13] K. Fikse, “Accelerating the Search for Optimal Dynamic Traffic Management,” Master Thesis, University of Twente, January 2011.

[14] D. Curiac and C. Volosencu, “Urban Traffic Control System Architecture Based on Wireless Sensor-Actuator Networks,” in *Proceedings of the 2nd International Conference on Manufacturing Engineering, Quality and Production Systems, pp. 259-263*, September 2010.

[15] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, 1st ed. Cambridge University Press, July 2008.

[16] SICK AG. (2011) LMS100 and LMS111. Accessed: 2011-03-29. [Online]. Available: http://www.sick.com/us/en-us/home/products/product_news/laser_measurement_systems/Pages/lms100.aspx

[17] MikroTik. (2011) Routerboard. Accessed: 2011-04-01. [Online]. Available: http://www.routerboard.com/rb500.html

A Wireless Electronic Nose for Emergency Indoor Monitoring

Vincenzo Di Lecce, Rita Daria
DIASS – Politecnico di Bari
Taranto, Italy
{v.dilecce, r.dario}@aeflab.net

Jessica Uva
SINCON s.r.l.
Taranto, Italy
jessica.uva@sincon.it

***Abstract*—The non-compliance with the ambient air quality standards in hospitals as well as in all working settings can cause possible adverse effects on health of people. It is extremely important to monitor public exposure of air pollutant contaminants in order to prevent air quality related emergencies such as heavy smoke from fires or toxic releases.**

In this paper a wireless device for monitoring air quality is presented. Each node, based on commercial gas sensor arrays and a ZigBee wireless network interface, is local powered by plugging it into a generic light socket. No wiring is required for data transmission. Air data measured by nodes are delivered via ZigBee network to a database station where different technical analyses can be used to obtain pollutant dispersion information. This system can be used for monitoring and classifying Hazardous Air Pollutants in various places: laboratories, offices and operating rooms. Thus, the wireless electronic nose approach aims at strengthening its potential for automatic environment monitoring with regard to many industrial applications too.

Keywords: e-nose, environment, emergency

I. Introduction

The implementation and development of electronic noses are linked to different areas of application today. In food field artificial olfactory systems have an important role for preventing food fraud. Similarly, in industrial areas (e.g., industry of perfumes and cosmetics) there are many potential applications, such as the dangerous chemical agent detection. In the healthcare field, extensive researches, based on a thorough examination and appropriate investigations of relevant data as a result of medical diagnosis in this regard, were conducted. The most promising results have been obtained with the detection of lung cancer through breath. The principle is based on clinical chemistry. In fact, the chemical composition of a person's breath changes when they develop this disease. Furthermore, numerous scientific studies have clearly demonstrated the influence that the air has on human health. The great importance of the continuous air monitoring in residential areas or in workplaces is really self evident. Particular attention is given to the risk factors for specific population groups (children, elderly, sick, etc). This paper aims to define a low-cost, real time, self configurable sensor network for the continuous evaluation of indoor/outdoor air quality.

A classical limit of monitoring is the data network wiring. Currently, fully wired buildings are going to become the norm, but this feature is typically reserved to working areas and not to service and technical areas (i.e. storage room, basement and wire way).

Particular consideration, by the same authors, has been given to air quality in hospital environments. In [1] we report an application of monitoring system within laboratories, operating rooms, and executive offices. However this technology can be applied to several industrial and domestic fields.

From the environmental perspective, the hospital is characterized by a mixture of gases and chemical products (many of which are volatile organic compounds) used in therapy and generic activity (cleaning, maintenance, potting etc).

The monitoring stations, for checking the presence of hazardous substances to human health, are only located in operating rooms and in specific laboratories (that use antiblastic and another dangerous compounds). Therefore, these devices cannot provide the control of the whole hospital, but only of some spaces.

In addition to human risks, the presence of oxygen, butane and other technical gases highlights the risk of explosion.

Significant examples of rules to be observed are offered by the Italian Public Health Office: Circular letter n. 5 of the year 1989, that establishes the maximum concentration levels of substances used for anesthesia gases in operating rooms (*Guidelines for environmental and hygienic procedures in operation theatre)* and the DECREE on April 9, 2008, n. 81 for the protection of health and safety at work [9].

It is clear that the proposed technologies can be used in fire, flooding and contamination detection [2-3].

With the continuous development of wireless sensor network (WSN), the usability of sensors is becoming increasingly prevalent in our environments. Therefore, combining an electronic nose technology with WSN to realize wireless detection system can be seen as a very important step in the right direction of human safety research. According to this trend, a wireless electronic nose has been developed; the device has been based on IEEE 802.15.4 or the so-called ZigBee and tested for environment monitoring in various rooms and areas.

ZigBee technology is a short-range (10-300 feet), low power, low-speed, low-cost wireless communication technology, mainly for wireless sensor networks, automatic control and remote control areas. It is, generally, considered as a good wireless communication protocol, because it fully meets the requirements of WSN application and owns such property as higher reliability, self-organization network, self-cure capacity and large network volume [2].

In this paper after a methodological approach, describing the sensors and the circuits of the electronic nose, a case study with the related results is discussed.

II. METHODOLOGY

The design of the wireless electronic nose node (e-nose) can be divided into three parts including sensor and electronic hardware, power supply and ZigBee network interface.

A. Electronic Hardware and Sensor

The e-nose is composed of an acquisition module comprising the sensors and the signal conditioning circuitry, and an interface to the ZigBee transponder, also a different protocol transponder or a data-logger that records the data in an SD memory can be used [4]. For specific applications in medical and hospital environments, the following sensors are provided: LM335 sensors for detecting temperature, HIH-4000-001 for relative humidity, TGS2620 for CH_4 and TGS2602 to monitor CO, MQ811 for CO_2, and MQ137 for NO_x, and MQ131 for O_3 and MQ136 for SO_x.

The criterion for the choice of these sensors was based on the need for covering the greatest range of pollutants to be observed in hospital environment [Table I].

These sensors are semiconductor gas sensors and they are widely used for detecting inflammable gases and certain toxic gases in air. The adsorption or reaction of a gas on the surface of the semiconducting material induces a change in the density of the conducting electrons in the polycrystalline sensor element. This chemical reaction can be described by four steps as follows:

- Pre-adsorption of oxygen on semiconducting material surface;
- Adsorption of specific gas;
- Reaction between oxygen and adsorbed gas;
- Desorption of reacted gas on surface.

The sensing element gas sensor is a tin dioxide (SnO_2) semiconductor, which has low conductivity in clean air. In the presence of a detectable gas, the sensor resistance decreases depending on the gas concentration in the air. An electrical circuit can convert the change in resistance to an output signal operating the preliminary correction as function of temperature and humidity.

B. Power supply

A very simple and low cost constant voltage power supply has been used. Main feature is 5V DC, 1 A (high current for gas sensor header supply). The circuit is designed around ON's NCP1014 integrated controller with internal mosfet in a discontinuous mode flyback topology. A low drop serial regulator IC has been used to reduce 5 Volts outputted to 3.3 volts for ZigBee transponder supply. Due to low current used by the transponder, the dissipated thermal power is about 170 mW. In order to evaluate the features of powerline communications an OFDM-based modem is also present on the board [Figure 1-2].

C. ZigBee Modules

ZigBee technology is based on the IEEE 802.15.4 standard, which also represents the personal area wireless network (PAN) [4]. ZigBee is characterized by its low cost, low power consumption and miniaturization.

The ZigBee stack architecture defines two layers, namely, the physical layer and the medium access control sub-layer. The ZigBee alliance provides the network layer and the framework in the application layer. In the test-bed a commercial ZigBee module has been used (XTR-ZB1-xHE from Aurel). From specifications, the line of sight distance of this module can extend up to 1000 feet (open air) with power consumption of 350 mW. ASCII strings commands are used to configure the module (PAN ID, channel scan function, time to join the network, destination address, hopping parameters, data baud rate, sleep mode function

TABLE I. SENSORS USED IN THE ZIGBEE E-NOSE NODE PROTOTYPE.

Gas	Sources	Typical concentrations	Sensor
CO_2	Decomposition of organic matter; combustion	380 ppm throughout the troposphere	MQ811
CO	Decomposition of organic matter; combustion of fossil fuels	0.05 ppm unpolluted air, 10-50 ppm in the urban traffic areas	TGS2602
CH_4	Decomposition of organic matter; natural gas emission	1-2 ppm throughout the troposphere	TGS2620
NOx	Electric shock; internal combustion engines, combustion of organic matter	0,01 ppm ppm unpolluted air; 0,5 ppm in air pollution	MQ137
O_3	Electric shock; photochemical smog	Up to 0.01 ppm in the air unpolluted, 0.5 ppm in photochemical smog	MQ131
SO_2	Volcanic gases, forest fires, fossil fuels, roasting ore	Up to 0.01 ppm in the air unpolluted, 0.5 ppm in photochemical smog	MQ136
Temperature		-10 °C to + 40° C	LM335
Humidity		0-100% relative	HIH-4000

ISBN: 978-1-61208-144-1

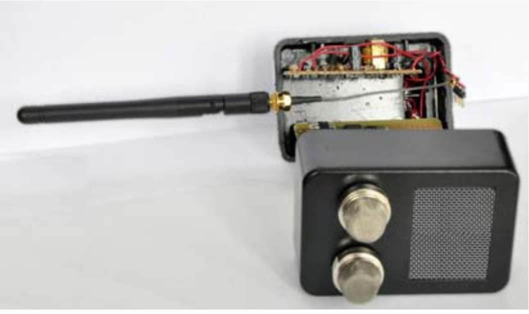


Figure 1. Wireless electronic node for environmental emergency application (e-nose). Here, it is possible to note the power supply modules with OFDM modem, the ZigBee transponder antenna and the gas sensors.

etc). This ZigBee module uses a frequency band at 2.4 GHz and 128 bits cryptography. We have employed a stylus external antenna in order to work also in reinforced concrete buildings. The Figure 3 depicts a ZigBee module (on the right).

In general, a ZigBee network can be implemented using one of topologies, as depicted in left side of the Figure 3. The used modules adopt the mesh topology allowing nodes to have as many paths as possible for communication. As a result, the mesh topology is very reliable and the extension of the network size can be done with a simple reconfiguration (up to 6 links in the used Aurel firmware). However, this network topology requires each node to be activated for most of the time, thus consuming more energy than other topologies. The base node is realized using the same module, with specific network interface (in our case USB) collecting data from all nodes in the network.

A previously-tested database, MySQL base was used for storing data. Before storing, two relevant operations are made on the row-data. The first one is needed to correct the sensors output for temperature and humidity dependence [5]. The second one is needed to reduce the ambiguity of this class of sensors.

It is well known that low-cost oxide-based resistive sensors are sensitive to a wide spectrum of VOCs, however their selectivity is generally low [6] thus providing an ambiguous response in terms of individual components of the gas mixture. Several attempts have been made to overcome this problem. With reference to tin oxide chemical sensors, two typical measurement strategies are employed [7]: multi-sensor arrays or dynamic measurements based on a single sensor [8].

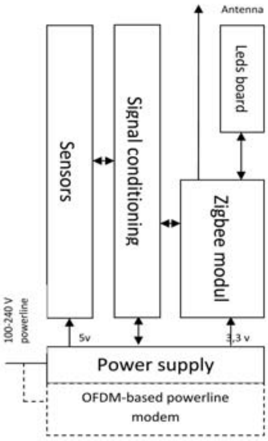


Figure 2. Block diagram of the electronic node. Dot line highlights an additional feature under test.

In this paper, according to previous works of the same authors, an IF THEN inference system is used [10].
The experimental results seem to be in good agreement with what has been previously observed experimentally for these systems and show the effectiveness of the proposed method consistent with those in literature

III. APPLICATION AND CASE STUDY

In hospitals both medical gas cylinders and compressed gases are used. The most commonly used gases are: oxygen, air, carbon dioxide, propane, acetylene and anesthetic gases (nitrous oxide and halogenated agents). They are generally available in vial forms. These substances, with their distribution systems are the subject of close attention for a

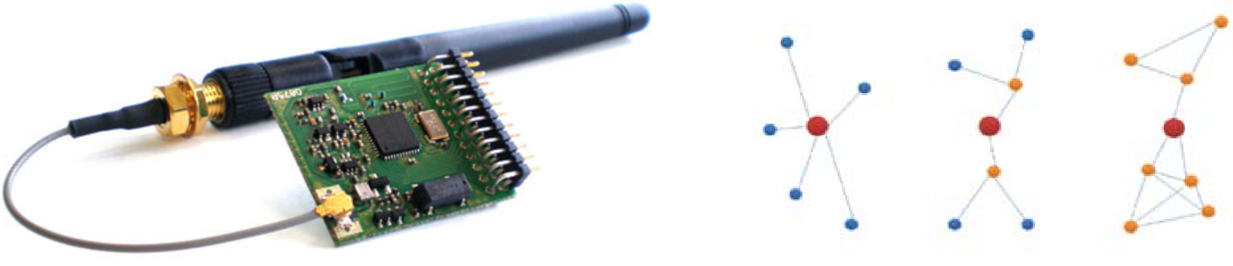


Figure 3. On the left: high power Aurel XTR-ZB1-xHE module, based on IEEE 802.15.4 standard and ZigBee protocol [15]; On the right: topology of ZigBee wireless networks: star, tree, and mesh.

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

proper assessment and management of potential chemical hazards in a hospital [12]. We can suppose that in a hospital there could be:

✓ Operating rooms and medical center
✓ Hospital rooms
✓ Offices
✓ Laboratories and kitchen
✓ Hall and open public areas
✓ Unattended rooms.

Here, several logistic and health activities are performed and these require the adoption of specific security measures. In particular, the operating rooms [11] are equipped with ventilation and air change systems and fixed monitoring stations. In fact, here there are many pollution sources due to the several activities performed by the staff.

Another important issue regards air quality of hospital rooms and medical center, equipped with air conditioning systems. Therefore, in these areas, it is important to adhere to air quality safety features by equipping the rooms with humidification and dehumidification system, filtration, and air flow regulation devices. In fact, in these rooms microclimatic conditions must be respected according to Italian standards, (referring to D. P.R. 01/14/1997). In these areas, however, air quality can be altered (e.g., a low number of air changes per hour, cigarette smoke, etc). This situation cannot be detected, because in these places there are not any monitoring devices.

The same thing could occur within the laboratories. They are a potential source of chemical hazard. Here, the solvents and detergents have to be used, in accordance with appropriate safety procedures (under the hood, with gloves and masks). However, they could cause accidental releases of hazardous substances thus bringing about dangerous situations such as poisoning or fire.

The kitchens are hazardous too for the presence of propane gas. Specific risk areas may also be halls (open public room) and administrative offices. For adequate air exchange, these places must be equipped with efficient air conditioning systems and active and passive ventilation systems providing fresh air circulation. A default of air exchange may be a source of microbiological risk, causing the inhalation of microbial aerosol or deposition of contaminated particles. Particular attention should be given also to the unattended areas, which represent approximately 20-30% of all hospital. The lack of staff to check regularly or occasionally spills, changes in humidity or other similar risks is a condition that deserves attention. They are typically intended for ancillary services to health care:

✓ Technical areas
✓ Paper archives
✓ Computer rooms
✓ Conducted for the gas adduction gas and other products within the hospital
✓ Power and data distribution lines.

In normal conditions there are not gas leaks and therefore the direction does not perform a periodic inspection. In fact, usually, technical staff is only in charge of annually maintenance operation, with the exception of occasional conditions, such as emergency or concurrent events (restructuring, maintenance) [14].

The fire detection system, as required by the regulations for public and private healthcare settings, is the only security measure that is effective in these places.

In addition, many of these spaces are also used to store laboratory materials, disused electrical devices or waste. This stuff stationed here for a long time.

In fact, the material can show the signs of wear over the times. This could cause small leaks of various kinds of substances, sometimes with harmful consequences. It should also be recalled that seasonal temperature changes i.e., the climatic variations occurring from warm to cold seasons in these storage rooms involve a high risk of incorrect maintenance of materials, for instance the degradation of papers or radiological materials. In many cases, these risk situations are very difficult to cope with if there is no outside automatic control.

In these conditions, the environmental relief appears to be very complex. It will be measured only by the subjective perception of people entering into the room (smell of damp, smelly material, acrid smell of gas). The devices that can detect the problem are extremely expensive and complex (e.g., hygrometer, gas detector, samples of chemicals).

These phases of control, however, are not very useful in

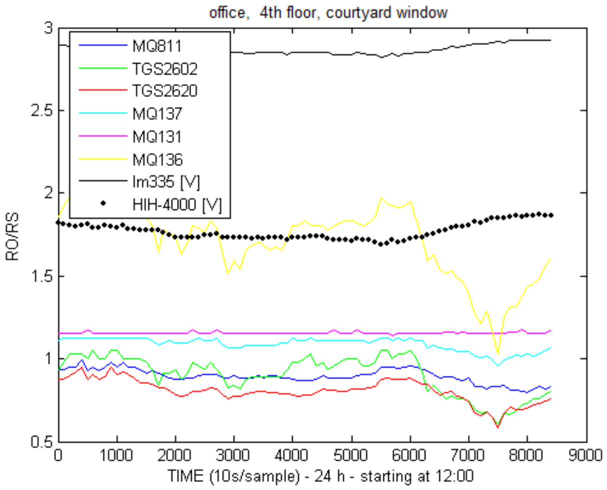


Figure 4. Row-data of the acquisition realized in an office on the 4th floor of the hospital with a courtyard window.

emergency situations that may arise from sudden emission of toxic gases, when we have not time to sample the surrounding air and analyze it with sophisticated instruments that provide detailed concentrations of pollutants, but require extremely long time to report a hazard [13].

According to the aim of this paper, the wireless e-nose was tested in three operating rooms, in the hall and an office of a public hospital. The first test has been realized in an office on the 4th floor with a courtyard window. The office belongs to a Head Physician. Row-data (electric output of the sensors as ratio Ro/Rs) from the sensor, except for direct voltage output sensors (LM335 and HIH-4000-001) are shown in Figure 4. The same data after humidity/temperature correction and disambiguation are shown in Figure 5 according to [10].

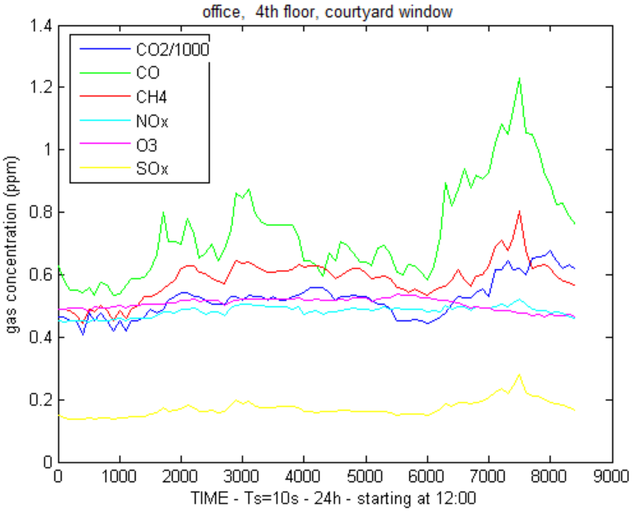


Figure 5. Gas concentration (ppm) of the acquisition realized within a 4th floor office of the hospital with a courtyards window.

The second test has been realized in the hall of the Hospital with the access to Emergency Room. The hall is open and exposed to an urban traffic-congested street, with private cars, ambulances and bus. There are at least 30 persons including patients, relatives, staff and waiting persons. Gas concentrations in ppm are reported in Figure 6.

The last test has been made in a surgical room. As already reported in [4] all operating rooms are equipped with continuous monitoring stations and they have air conditioning and ventilation systems with HEPA filters. The volume of each room is 25 m^2. The presented data are related to a plastic surgery facility (specifically in relation to mastectomy operation). The smell of alcohol was perceived by the operators, probably due to use of Betadine (alcohol disinfectant), Sevoflurane (vaporized) and Propoform. During the surgery operation, the door of the operating room is often open or ajar and there is continuous passing-by of personnel (doctors, nurses and auxiliary ones). An electrosurgical device was used. Gas concentrations of the monitoring performed in operating rooms on the 1th floor in 1 hour of acquisition, are shown in Figure 7 and, after 25 minutes, a probably sensor saturation can be seen, synchronized to use electrosurgical devices and a strong smell perceived by the operators.

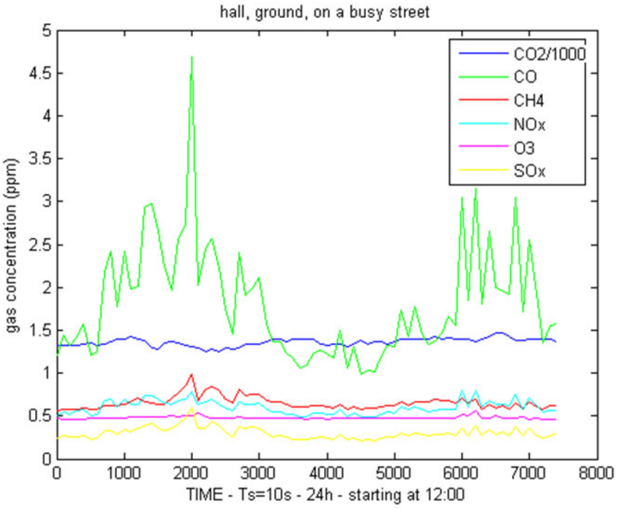


Figure 6. Gas concentrations of the monitoring carried out in a hall hospital on the ground floor exposed to an urban traffic-congested street (private cars, ambulances and bus). There are at least 30 persons including patients, relatives, staff and waiting persons.

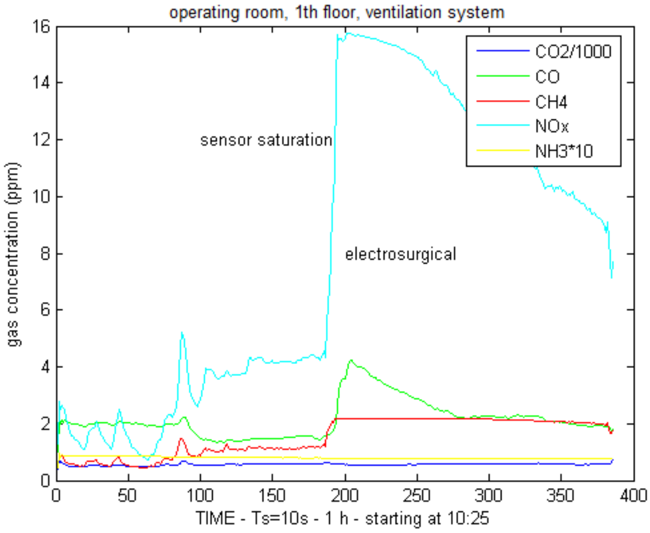


Figure 7. Gas concentrations of the monitoring carried out in operating room on the 1th floor in 1 hour of acquisition, where, about after 25 minutes, the electrosurgical device was used during a mastectomy operation.

IV. Results and Conclusion

In this study, six gas sensors were used to test air quality inside different rooms and areas of a public hospital. It should be noted that after a warming up period (some minutes for data sheet, about 1 hour in our tests) the sensors are able to measure the presence of pollutants in the air.

Typical responses of gas sensing measurement are shown. None of the data are reposted by the existing sensor. Some of these data show the exceeding of the limits imposed by a directive of Italian Public Health policy.

But low-cost solid state sensors are generally sensitive to a number of different gases and are characterized by a high response to the temperature and the humidity. In this work, the problem of discrimination abilities of these sensors has been solved. The basic idea grounds on the hypothesis that, if the same gas is actually measured by two or more sensors, then their estimated concentrations will be similar, with accuracy related to the number of concordant sensors.

The sensors are connected to the hub using the ZigBee protocol. Main features of this choice refer to the high connectivity ability. In particular, the adopted mesh topology is very reliable and extension of the network size can be done without reconfiguration.

Joining these two technologies a very cheap gas sensor network arrangement has been realized, tested and proposed in this paper.

The same technology, tested for a hospital application, can be used in any class of application. Farms, workshops, offices, public area can suffer from air contamination. The classical solutions, based on air sample or chemical tests are very slow and expensive. Moreover, the proposed solution does not require wiring and fixed installation, it is possible to reconfigure dynamically the network configuration. As described, a kit with a mixture of different selectivity e-noses can be considered as a smart and cost effective solution to emergency air monitoring.

Therefore, the experimental results, albeit grounding on a set of catalytic sensors, seem to support the idea that smart compositions of low-cost sensors are able to manifest surprising discrimination abilities.

The achieved goals are in line with our expectations and show that overall our system is able to perform an analysis, efficient enough to be of practical use. Future works will focus on studying further improvements of the system for getting higher accuracy level measurements.

Aknowledgments

We are grateful to the reviewers for their helpful suggestions and for their proposed corrections to improve the paper. The authors would like to thank Mr C. Martines, V. Guastamacchia e G. Mola for having contributed to the production of the results obtained by the acquisitions performed in the hospital.

References

[1] V. Di Lecce, A. Amato, R. Dario, C. Martines, Air Quality Control for Health Care Centres. The Application of an Intelligent Distributed System. 2009 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems. EESMS 2009 Crema (Italy) 25/09/2009, ISBN: 978-1-4244-4848-7 . Pag 27-30.

[2] T. Defeng, L. Shixing, X. Wujun, Z. Yongming “A Fire Monitoring System in ZigBee Wireless Network” CYBERC 2010 - The 2nd International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery CyberC 2010. October 10-12, 2010. Huangshan/China.

[3] M. Xiangyin, X. Shide, X. Ying, H. Huiping “ZigBee Based Wireless Networked Smart Transducer and Its Application in Supervision and Control System for Natural Gas Gate Station”. July 25-28, 2009 Nanning, China.

[4] R. Dario, A. Mundo, A. Leaci, V. Di Lecce, J. Uva, G. Bianco “Medical Environment Spot Exposure: a Solid State Personal Dosimeter” 2010 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems, pag. 19-25. Taranto, Italy, 9 September 2010.

[5] http://www.figarosensor.com/products/common%281-104%29.pdf

[6] P. Moseley and B. Tofield “Solid-State Gas Sensors” Bristol, U.K. Adam Hilger, 1987.

[7] X. Huang, F. Meng, Z. Pi, W. Xu and J. Liu “Gas sensing behavior of a single tin dioxide sensor under dynamic temperature modulation”, Sensors and Actuators B: Chemical, Volume 99, Issues 2-3, 1 May 2004, pp. 444-450.

[8] A. P. Lee and B. J. Reedy, “Temperature modulation in semiconductor gas sensing”, Sensors and Actuators B: Chemical, Volume 60, Issue 1,2 November 1999, pp. 35-42.

[9] http://www.ego-gw.it/public/about/VIR-0558A-09.pdf

[10] V. Di Lecce, M. Calabrese, R. Dario “Computational-based Volatile Organic Compounds discrimination: an experimental low-cost setup” Proc. of the 2010 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications Taranto (CIMSA2010), Italy, 6-8 September 2010, pp. 54-59, ISBN: 978-1-4244-7229-1.

[11] S. Mehta, W. J. Cole, J. Chaw and K. Lewin, “Operating room air pollution: Influence of anaesthetic circuit, vapour concentration, gas flow and ventilation”, Canadian Journal of Anesthesia, Volume 22, Number 3, pp. 265-274.

[12] M. Leung, A. H.S. Chan, “Control and management of hospital indoor air quality”, Med Sci Monit, 2006; Volume 12, Number, 3, pp.17-23.

[13] J. E. Hall, K. A. Henderson, T. A. Oldham, S. Pugh and M. Harmer, “Environmental monitoring during gaseous induction with sevoflurane”, British Journal Anaesthesia Volume 79, Issue 3, pp. 342-345, 1997.

[14] D. Norback, M. Ingegerd, J. Widstrom, “Indoor air quality and personal factors related to the sick building syndrome”, Scand Journal Work Environmental Health, Volume 16, pp.121-128.

[15] http://www.aurel-zigbee.com/download/documents/italiano/manuale-XTR-ZB1-xHE.pdf

Design of Noise Measurement Sensor Network: Networking and Communication Part

Ilkka Kivelä, Chao Gao, Jari Luomala, Ismo Hakala
University Of Jyväskylä
Kokkola University Consortium Chydenius
P.O.Box 567, FI-67701, Kokkola, Finland
{*ilkka.kivela, chao.gao, jari.luomala, ismo.hakala,* }*@chydenius.fi*

***Abstract*—In this paper we report the design and implementation of the networking and communication part of a WSN application for measuring industrial and residential acoustic noise. The network is formed in tree topology and a global synchronization is achieved. A link-state routing is tightly binded with the synchronization so that the network overhead is greatly reduced. Transmission scheduling is implemented in the network due to the fact that noise measuring is time-correlated, resource-consuming, and uninterruptible. The application is built on the *CiNet* cross-layer protocol stack. In our testbed, two IEEE 802.15.4 platforms (Chipcon CC2420 and Jennic JN5148) worked seamlessly. The uniqueness of this application is that it combines routing, global synchronization, and scheduling under a single framework. The network has been already deployed in the residential area of Kokkola city on the western coast of Finland.**

***Keywords*-synchronized network; routing; scheduling; noise measurement; environment monitoring**

I. Introduction

Wireless Sensor Network (WSN) design is usually application oriented [1]. Different applications have different requirements and objectives in protocol design. Though there are plenty of proposals published for WSN, yet a specific application needs a specific solution which is usually an optimization and trade-off between available proposals.

In many countries, environmental acoustic noise is regarded as a critical metric for working and living comfort. Recent studies have shown that if people are exposed to environmental noise levels that are too high, this increases the risks for hearing problems but it also contributes to ischemic heart diseases, hypertension and sleep disorders [2][3]. The European Commission also states that environmental noise negatively affects productivity and that it is one of the major environmental problems in Europe [4].

The traditional way of conducting noise measurements is cumbersome: A technician has to carry a Sound Level Meter (SLM) to a measuring location, set the meter up for a necessary measurement which usually takes several hours and repeat this procedure for all the measuring points. The disadvantages of this method are obvious: 1) a commercial SLM is expensive, making large-scale measurements very costly, 2) point-by-point measurements make the results incoherent timewise, 3) due to the lack of a communication facility, the measured result will not be available in real-time, and 4) SLM needs full attention, which is increasing the work load.

There is a requirement for measuring acoustic noise in both industrial and residential areas in the Ostrobothnia area of Western Finland. In Kokkola city area, officials need a flexible and inexpensive method to monitor environmental noise. Noise sources include loading cargo on a train or a rock concert, among others. Thus the measuring system should be able to cover a large area, such as a university campus, an industrial park, or a residential block. System should be able measure over weekend and store continuous noise levels (1s samples). Real-time data is needed when monitoring rock concert so that officials could react on time. This local need and disadvantages associated with the traditional method gave us the motivation for the design and implementation of a wireless noise sensor network. In such a WSN, a set of wireless sensor nodes are scattered within a concerned area. Each node measures the acoustic noise level at its location, and the measured data is collected by a sink node. Compared with the traditional method, the designed system has the following significant advantages: 1) cost reduction in both sensing devices and workload; 2) real-time, multi-point, coherent measurement; and 3) minimal attention required. The uniqueness of this application is that it combines routing, global synchronization, and scheduling under a single framework. The rest of the paper is arranged as follows: Section II outlines the related work concerning environmental noise monitoring. In section III we illustrate the details of protocol design, including platform selection, protocol stack architecture, timing and synchronization, link-state routing etc.; in section IV the results and corresponding analysis are given; Section V summarizes the design and provides some prospects for future work with this application.

II. Related Work

Transmission of noise sensor measurements in real-time also sets special demands for network protocols. As far as

ISBN: 978-1-61208-144-1

we know, there are only few reports in literature resembling our approach to the problem. This is because wireless noise measurement for WSN is quite special. Thus we review reports and commercial tools related to wireless noise measurement.

The work carried out in [5] is probably the most related one to our project. In that project the Tmote Sky platform was used, and sensors were deployed to measure road traffic noise. From the measured data it is possible to count the number and type of vehicles. The authors assert that large-scale noise measurement using a WSN solution is possible. However, the accuracy of their measurements is not mentioned, and the calibration of nodes was left as an open issue in their work. The sampling rate is set at 8kHz due to the CPU/ADC limit, which does not cover the proper acoustic frequency range.

In [6] the authors stated that wireless sensor network is feasible for the use in environmental noise monitoring. They also evaluated three hardware platforms and two data collection protocols. By using their own custom noise level sensor, demanding noise level calculations could be delegated to dedicated hardware. Their protocol comparison results showed that CTP (Collection Tree Protocol) with LPL (Low Power Listening) provides better performance in terms of energy efficiency compared to CTP and DMAC protocols.

There is a Bluetooth-based solution for noise measurement available in market [7]. In this solution, a Bluetooth piconet which supports a maximum of 5 noise sensor nodes can be deployed. It does not support multi-hop communication, and therefore the application's scale is quite limited.

In SoundEar Pro [8], 10 independent noise level meters can send data wirelessly to the PC within maximum range of 70m. There was no mention about the technology employed in this.

APL Systems Aures [9] is a wireless noise measurement network, which can measure noise levels at multiple points at the same time. The technology behind the product is hidden, but it is told that system works in single a cluster, containing a single network controller and Aures devices.

Compared to reviewed solutions, our system has significant advantages. With multi-hop feature, our wireless noise measurement network is able to cover a large area. Low-cost design makes system much cheaper, decreasing the costs of installation. Because the system uses battery powered sensor nodes with dynamic data routing in network, it is also more flexible than any of the reviewed systems. By choosing suitable duty cycles, the lifetime of the designed battery-powered measurement network can be extended to months.

III. Protocol Design

We only describe the networking and communication function design in this document. The sensing function, hardware design and energy consumption results can be found in [10].

A. Design Objectives

The network is able to support multi-hop tree topology so that a large area can be covered. For example, monitoring environmental noise of surrounding areas of open air rock concert, the network should cover several hundred square meters. In a network size this means that it has 2-3 clusters which each includes 3 to 6 noise sensors. Throughput should be high enough so that the loss of data does not affect the precision of long-term statistics. A global synchronization is necessary in order to produce timely correlated noise data.

The sensors are able to measure the acoustic noise continuously for a long enough period, usually for a whole weekend, and the cost of sensor nodes must be minimized.

The most challenging feature of this application is the collection of large amount of noise data (72 bytes every 5 seconds for every sensor node in network) from the whole network, and delivering that to the sink node in a very short communication window. When a sensor node is measuring the noise, it has to turn off the radio transceiver due to that 1) noise ADC (Analog-to-Digital Conversion) sampling can not be interrupted, 2) radio activities are the source of significant interference to the sampling circuits. This leaves only a short time for all the sensor nodes to send and receive.

Our first design was to let the sensor nodes near the SINK relay data for the remote sensors. Soon we found that it had a severe scaling problem when the number of sensors grows. In order to alleviate the bursty traffic, relay nodes are introduced for the remote sensors. Thus our target WSN contains a SINK node, a set of relay nodes, and a set of noise sensor nodes. The network topology is a multi-hop tree with SINK as the root of the tree. The sensor nodes can be deployed at any arbitrary location in the area concerned. An example application of such a network is shown in Figure 1. The relay nodes take care of relaying sensor packets to the sink.

B. Platform Selection

We choose IEEE 802.15.4 standard [11] compatible modules as our design platform for the following reasons:

1) We had already integrated (microcontroller + RF) modules, with the cross-layer architecture CiNet[12] implemented, which helped reducing the node size and the power consumption as well as the load on software development.
2) IEEE 802.15.4 is the *de facto* standard for WSN. This guarantees the portability and continuity of the project.
3) A sophisticated CSMA/CA MAC protocol eases the design of upper layer protocols.
4) IEEE 802.15.4 offers radio link statistics in terms of Receive Signal Strength Indication (RSSI) or Link

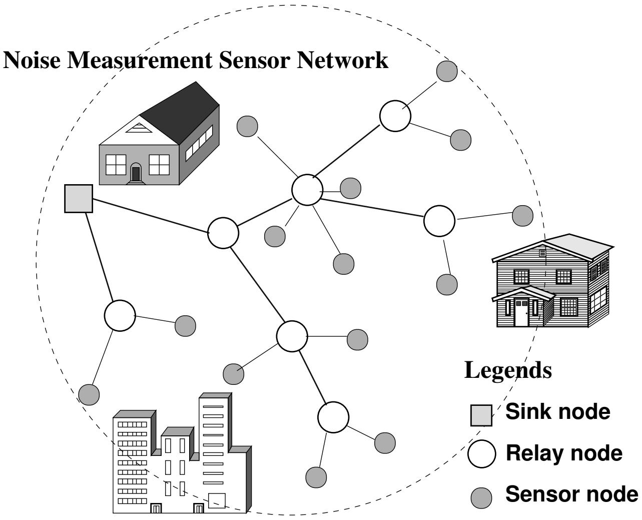


Figure 1. Noise measurement WSN example

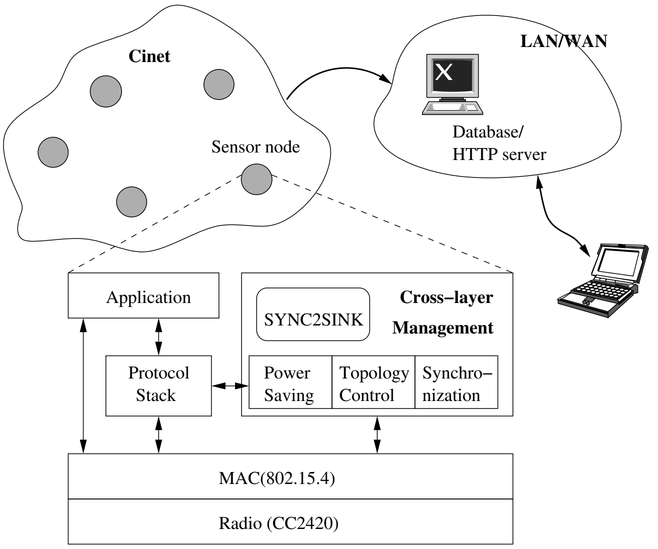


Figure 2. System and node architecture

Quality Indication (LQI). This helps to implement a high throughput link-state routing protocol.

In practice, two platforms are included in our development: one is a microcontroller-RF separated solution (ATmega128+CC2420) and the second is an integrated solution (Jennic JN5148). They work seamlessly in our testbed network.

C. Networking and Synchronization

We developed an integrated synchronization and routing protocol denoted as SYNC2SINK[13] to achieve our objectives. The SYNC2SINK protocol enables the nodes to establish and maintain a route to the sink using the information contained in synchronization frames. SYNC2SINK is built on the CiNet protocol stack[12], which is a cross-layer architecture in which time, radio link state, battery and topology information are shared by sensing, packet transmission and reception, and route table maintenance. The architecture of CiNet can be seen in Figure 2

In the SYNC2SINK protocol a node works periodically in 4 phases: synchronization phase, sensing phase, data communication phase, and optional sleep phase for energy saving. In this particular application the noise measurement has to be done continuously so that the sleep phase is removed. Thus the target network works periodically for synchronization, sensing, and data communications, denoted as $T_{syn}, T_{sen},$ and T_{com} respectively. However, note that during T_{sen} the radio part of the sensor nodes must be turned off due to the uninterruptability of noise sampling and interference reduction. Because the sensed data must be time-stamped, the SINK is synchronized by a server. The sensing phase is started synchronously right after the synchronization phase. The timeline activity of the network is shown in Figure 3.

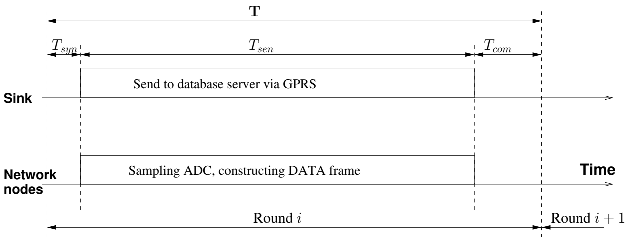

T
T_{syn}
T_{sen}
T_{com}
Sink
Send to database server via GPRS
Network nodes
Sampling ADC, constructing DATA frame
Time
Round i
Round $i+1$

Figure 3. Timeline of the network operations

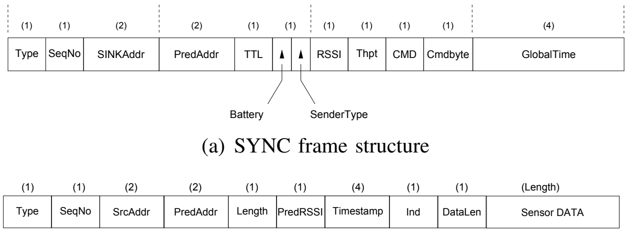
The SINK broadcasts SYNC frames periodically, with the period denoted as **T**. The frame size is 128 bytes in IEEE 802.15.4 networks. A current data frame can fit maximum of 4.5 second of noise data. Due to that, a 5 seconds sync period is chosen of which sensing phase is 4.5 seconds and 0.5 seconds is transmission phase. The broadcasting of SYNC has four functions: 1) let the whole network become synchronized, 2) let the relay nodes establish a route back to SINK, and 3) let the sensor nodes select the best relay node. The SYNC frame structure is shown in Figure 4(a) it has a length of 16 bytes, and function of its fields is given in Table I. More detailed function of the fields is illustrated in the following section.

Correspondingly, in each synchronization period a sensor node sends a DATA frame back to the SINK. The length of the DATA frame is variable and the structure of DATA frame can be seen in Figure 4(b), in which the *SeqNo* field and *GlobalTime* is copied from the latest SYNC, *SrcAddr* and *PredAddr* are the addresses of the sending node and its predecessor, respectively. This address couple is used by the SINK to figure out the network topology.

A relay/sensor node must be synchronized by the a SYNC frame, as it is not possible to relay or deliver any data if no

Table I
SYNC FRAME STRUCTURE

Field(octet)	Symbol	Description
Type (1)	—	Indicate that this is a SYNC frame
SeqNo (1)	s_i	Sequence number that increments every cycle
SINKAddr (2)	A_{sink}	Network address of the originating sink node (short IEEE 802.15.4 MAC address)
PredAddr (2)	A_{pred}	the predecessor of the sending node
MaxTTL (1/2)	TTL^*	The upper nibble is the initial value of TTL and kept constant during the broadcasting;
TTL (1/2)	TTL	the lower nibble is set by SINK as TTL^* and decremented by each node, so that the nodes can calculate the hop count to the SINK.
Bat (1/2)	B	The upper nibble, indicating the battery residual of the sending node.
ST(1/2)	—	Indicate the sending node type.
RSSI (1)	Q_{rssi}	Indicate the link quality
Thpt (1)	Ψ	Downlink throughput, basically a statistics of SYNC frame reception rate, can be used to estimate the uplink performance.
Cmd (1)	—	Network command given out by SINK.
CmdData (1)	—	Network command parameter
GlobalTime (4)	T_g	Global time in seconds



(a) SYNC frame structure

(b) DATA frame structure

Figure 4. SYNC and DATA frame structure

route is established. The relay/sensor node must follow a strategy where:

1) When powered up, a node sets the radio module to receive the mode and starts waiting for a SYNC frame for a *SYNC-hunting* time t_{sh} when $t_{sh} \geq \mathbf{T}$.
2) If the SYNC frame is received within t_{sh} time, the node takes the information from the SYNC to the route entry which contains 1) SINKADDR, 2) seqno, 3) predecessor addr, 4) Hop-count to SINK, 5) Link Quality (RSSI); then rebroadcasts the SYNC after the decrementing TTL field. The node is then running in *Synchronized mode*.
3) If no SYNC frame has been received within t_{sh}, the node turns off the radio and sleeps for a random backoff time t_{bk}. When t_{bk} expires, the node goes back to Step 1. Such an operation will prevent the node from spending unnecessary energy in *SYNC-hunting mode*.
4) If the node is in *synchronized mode* and the next m consecutive SYNC frames are missed, the node will turn back to *sync-hunting mode*.

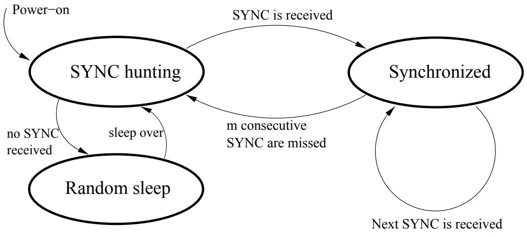


Figure 5. The state-transition diagram of SYNC2SINK

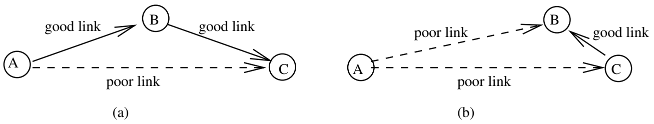


Figure 6. Problem of simple first-SYNC-routing

The state-transition diagram of the nodes is shown in Figure 5.

D. Link-State Routing

In our first design, the relay/sensor nodes establish the route path to the SINK by the taking parameters of the first received SYNC frame (i.e., a greedy algorithm). We found that the DATA frame Packet Receive Ratio (PRR) to SINK was very poor. This is due to the fact that radio links established by the strategy may be poor, because the first arrival SYNC usually comes from a node over the maximum distance. As shown in Figure 6(a), a better choice might be to have a two-hop route from node C to node A, using node B in the middle as a relay, even though C hears SYNC from the A node first.

In order to overcome this problem, we adopt a link-state routing protocol (Power-Aware Routing - PAR[14]). We utilize RSSI, which is a default IEEE802.15.4 MAC layer information. Upon the reception of a SYNC frame, the receiving node can choose a more favourable predecessor as a relay to SINK. According to [15], [16] and a number of related works, an RSSI>-75dBm or equivalently LQI>90 indicates a PRR≥90% over a single link.

The logic of PAR can be depicted as: if a node has received a SYNC frame with new $SeqNo$, it takes the sender of this SYNC as predecessor and mark the RSSI in the route entry; If a node has received a SYNC frame with the same $SeqNo$, it compares the RSSI with the previous one in the route entry. If 1) the new SYNC RSSI is within the range between a lower limit Q_L and a higher limit Q_H, 2) the old one is less than Q_L, 3) the hop count of the new one is no more than the old one plus 2, and 4) the sending node's predecessor is not the receiving node; the receiving node will replace the route entry by the new SYNC's information. If all the SYNC frames that come to this node have RSSI lower than the threshold, the node will simply use the first

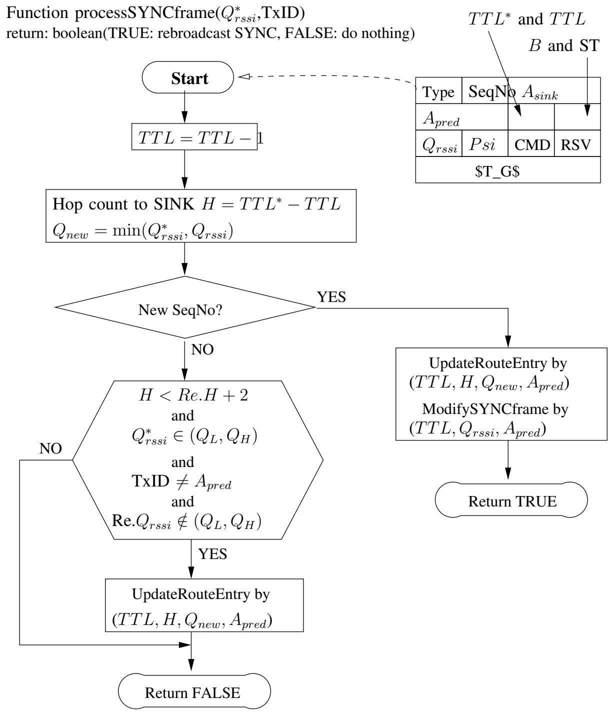

Function processSYNCframe(Q^*_{rssi},TxID)
return: boolean(TRUE: rebroadcast SYNC, FALSE: do nothing)
TTL^* and TTL
B and ST
Start
Type
SeqNo
A_{sink}
A_{pred}
Q_{rssi}
Psi
CMD
RSV
T_G
$TTL = TTL - 1$
Hop count to SINK $H = TTL^* - TTL$
$Q_{new} = \min(Q^*_{rssi}, Q_{rssi})$
New SeqNo?
YES
NO
UpdateRouteEntry by $(TTL, H, Q_{new}, A_{pred})$
ModifySYNCframe by $(TTL, Q_{rssi}, A_{pred})$
Return TRUE
$H < Re.H + 2$
and
$Q^*_{rssi} \in (Q_L, Q_H)$
and
TxID $\neq A_{pred}$
and
Re.$Q_{rssi} \notin (Q_L, Q_H)$
NO
YES
UpdateRouteEntry by $(TTL, H, Q_{new}, A_{pred})$
Return FALSE

Figure 7. Flowchart of SYNC frame processing ($Re.x$ represent the parameter stored in Route Entry)

one. Here we set a higher limit Q_H in order to prevent too short hops. As an example illustrated in Figure 6(b), both B and C have a poor link to A; however, when B receives a rebroadcast SYNC frame from C with a good RSSI, it shall not replace A by C as its predecessor.

The flowchart about the processing of a SYNC frame is given in Figure 7. The result of adopting the link-state routing can be seen in Section IV.

IV. Results and Analysis

Before setting up a testbed network, important parameters such as T_{com}, T_{syn} must be determined. These parameters are tightly bounded by the hardware properties such as clock oscillator precision, physical and MAC layer features of IEEE 802.15.4 such as CSMA/CA slot time and contention window. As mentioned in Section III, noise sampling must be as continuous as possible —therefore both T_{syn} and T_{com} shall be kept small.

A. SYNC Phase Time T_{syn}

A broadcast frame in IEEE 802.15.4 MAC does not require acknowledgement. According to the analysis in [17], the upper-limit time of broadcasting a SYNC takes

$$\begin{aligned} t_{bc} =& \text{randombackoff}([0, 2^3 - 1] \times 320\mu s) + \\ & \text{dataframeduration}(352\mu s) + \\ & \text{turnroundtime}(192\mu s). \end{aligned}$$

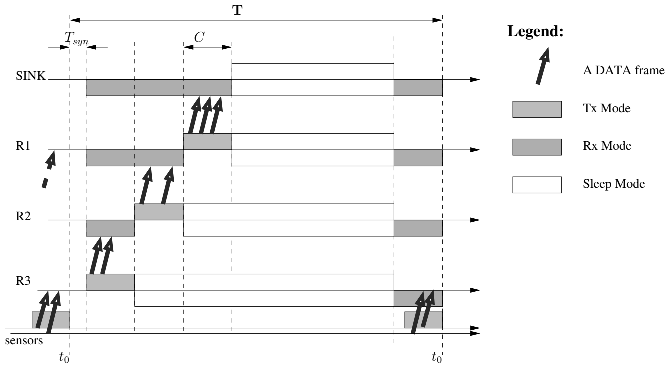


Figure 8. Inter-layer scheduling

This gives that the maximum $t_{bc} = 2784\mu$s. Thus the phase time for synchronization is

$$T_{syn} = \text{TTL}^* \times t_{bc} \times D \tag{1}$$

where D is the node density in terms of the node number that can hear each other in a given area.

This analysis does not consider the processing delay in each rebroadcasting node, which is actually a variable due to the task scheduling features of the running operating system. However, the processing of a broadcast message should be set as the highest priority task because this type of messages usually contains network management/control information, thus resulting in very short delay comparing to T_{syn}.

B. Transmission Scheduling

At the end of every cycle, each sensor node generates a DATA frame and sends it out. If we try to deliver all the data frames to the SINK within a small time slot, it will create a burst of radio traffic and PRR can not be optimistic. In order to mitigate the problem, we set up both inter-layer and intra-layer transmission schedules. Here "layer" means the hop count to the SINK.

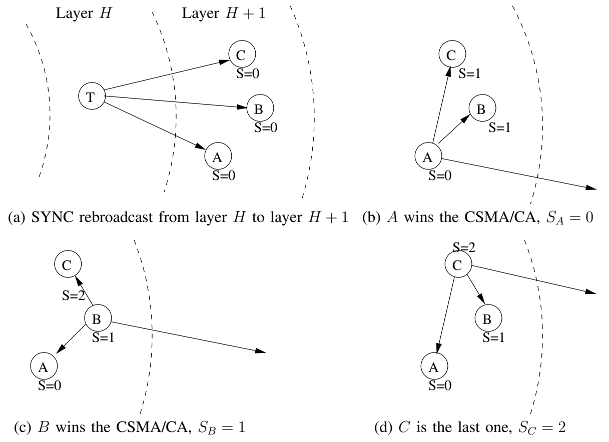
Inter-layer scheduling is based on hop count to the SINK, denoted as H. Each relay node schedules the radio transceiver into the transmitting mode by

$$t_{tx}(H) = C \cdot (TTL^* - H) + T_0 = C \cdot TTL + T_0 \tag{2}$$

where C is the scaling constant, and T_0 is the cycle starting time. Figure 8 shows an example. By this scheduling the remote nodes start the forwarding of DATA frames earlier than those closer to the SINK, and data frames are accumulated to the SINK in $C \times TTL^*$ time.

Intra-layer scheduling is done by the order of rebroadcasting SYNC frame. Since a SYNC frame can be heard by all the nodes in radio range, each node marks the sequence of its own broadcasting, and schedules the transmission by

$$t_c(S) = t_{tx}(H) + (S \mod D') \cdot C' \tag{3}$$



(a) SYNC rebroadcast from layer H to layer $H+1$ (b) A wins the CSMA/CA, $S_A = 0$

(c) B wins the CSMA/CA, $S_B = 1$ (d) C is the last one, $S_C = 2$

Figure 9. Intra-layer scheduling among nodes A, B, and C

where $t_{tx}(H)$ is obtained from (2), C' is a scaling factor, D' is predefined node density, and S is the sequential number of rebroadcasting SYNC frame in the same layer. An example of determining S is illustrated in Figure 9. A modulus operation bewteen S and D mitigates the scaling problem. However, if the node density is too high in some areas, multiple nodes will get the same t_c and in this case they will rely on IEEE802.15.4 CSMA/CA to avoid collisions.

C. Server-Sink Synchronization

The SINK acts simply as a passthrough gateway between the WSN and a webserver which manages the sensor data in the SQL database. Therefore the measurement must follow the server time so that sensor data can be correctly stored and displayed. The connection between the server and the sink is either a direct RS-232 link or a GPRS channel. In order to synchronize with the server time, a fine adjustment is implemented between the SINK and the server as illustrated in the follow.

Right after a SYNC frame has been sent out, the SINK immediately sends a polling message to the server. After the reception of the polling message, the server records the time when the message arrives, and compares it with the time of the previous arrival, denoted as $\delta_i = t_{i+1} - t_i - \mathbf{T}$, and immeadiately sends δ_i in a reply message, as shown in Figure 10. The SINK adjusts its next SYNC broadcasting time by $T_{i+1} = T_i + \mathbf{T} + \delta_i$. This adjustment will force the SYNC broadcasting to follow the server time.

This simple time adjustment worked fine in our testbed. We tested the algorithm for nearly 1 day and the synchronization was maintained well (with mean δ only 0.0767ms).

D. Test Settings

We designed 5 scenarios to test different aspects of our communication protocol. In the first scenario, we examined our synchronization performance. In the second scenario, we tested the link-state routing performance by varying the LQI threshold, to obtain an optimal LQI threshold. In the third scenario, single-hop capacity was tested. The last two scenarios were designed to verify dynamic routing and multi-hop communication performance, respectively. More detailed protocol parameter settings are given in Table II for all the scenarios.

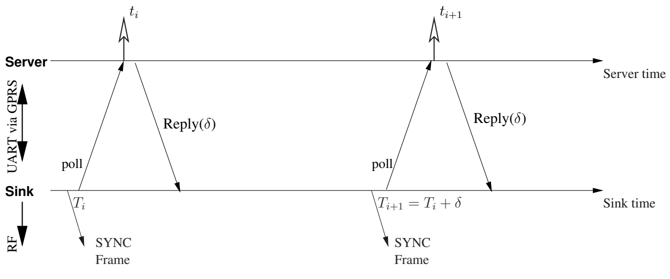


Figure 10. Time Adjustment between Server PC and the SINK

Table II
TESTBED NETWORK SETTINGS

Cycle time $\mathbf{T}$	5 sec.
Data frames k per cycle	1
Data frame length	80 bytes
T_{com}, T_{sen}	1/4 sec.
Node time precision	0.001 sec.
Inter-layer Delay Constant C	0.15 sec.
Run time of each senario	7200 sec. or over night
Q_H (LQI)	150
Q_L (LQI)	varying

We first put five noise sensor nodes in a coffee room to exam the synchronization performance. Figure 11 shows the time-line of the measurement results, and it can be seen that they are time-correlated.

Then we set up a test network which has nine relay nodes

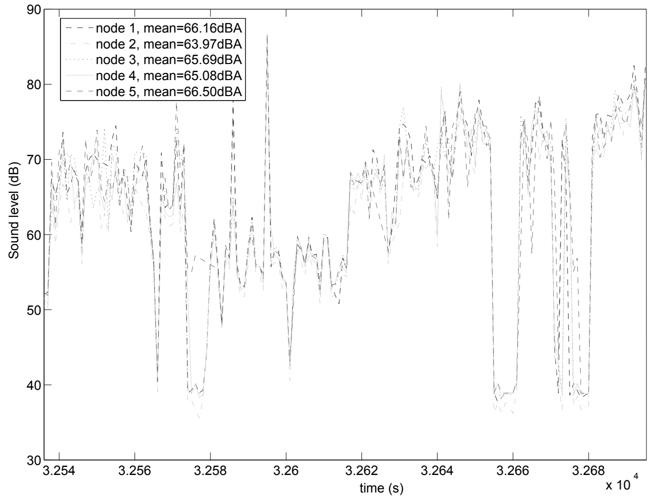


Figure 11. Measurement result of 5 nodes in a coffee room

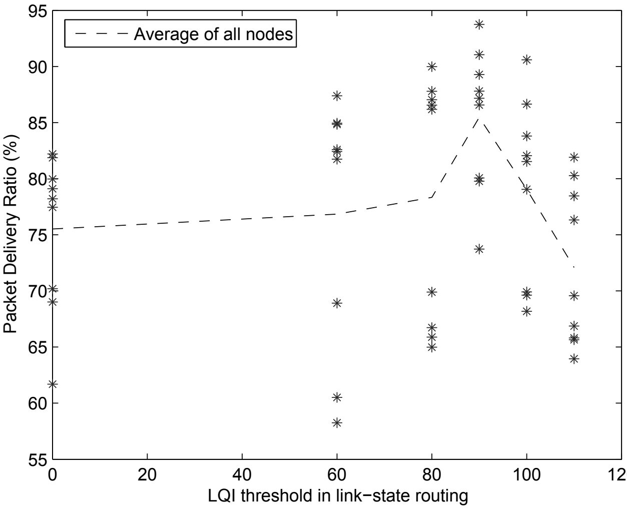


Figure 12. Packet receive ratio at different LQI settings in Link-state routing

and one SINK at 4th floor of our laboratory building. In each cycle, a relay node produces one data frame and sends it back to the SINK.

Figure 12 shows the link-state routing performance with different LQI thresholds. Note that LQI_{th}=0 (or RSSI=-128dBm) indicates a non-Link-State routing protocol. It can be seen that at LQI_{th}=90 (or RSSI=-75dBm) the PRR hits the maximum value in the multi-hop scenario, which improves PRR by 10% compared with that of LQI_{th}=0. A low LQI threshold results in that the network has smaller number of hops from the sink to the most remote nodes, but the radio link goodput is poor due to the long distance of the hops. On the other hand, a high LQI threshold gives a good radio link performance, but a remote data packet has to be relayed by more nodes back to the SINK, and it also aggravates the hidden node problem [18].

The third test was for single-hop capacity. We set up a number of sensor nodes as the first layer from a SINK. It can be seen that using intra-layer scheduling, one SINK can serve up to 14 sensor nodes with an average PRR greater than 96%. Table III shows the results of setting 12, 13, and 14 nodes respectively.

Table III
SINGLE-HOP PACKET DELIVERY PERFORMANCE

Node No.	PRR_{min}	PRR_{max}	Mean
12	0.9870	0.9978	0.9916
13	0.9739	0.9942	0.9868
14	0.9301	0.9892	0.9659

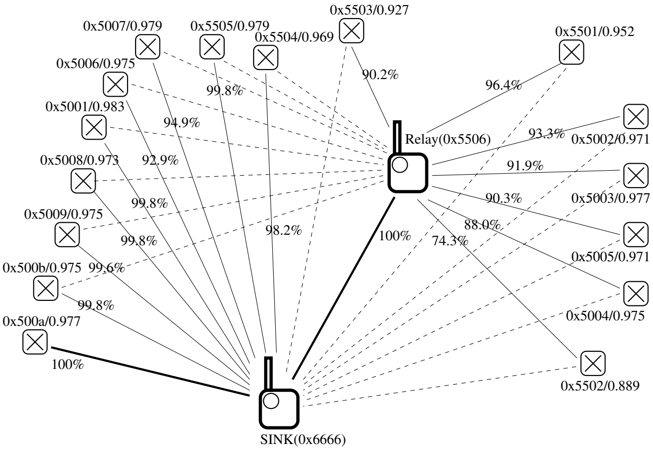


Figure 13. The 2-hop scenario setup and results. A solid line means that the link is used primarily. A dotted link means the link is occasionally used. The number associated to node is the node address.

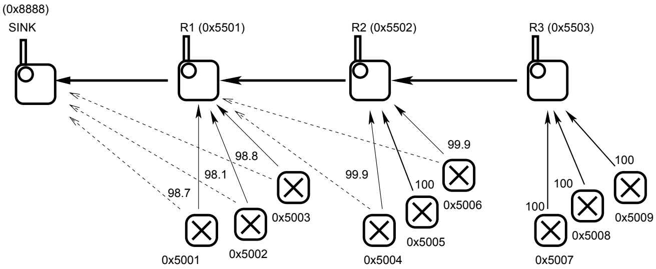


Figure 14. The 4-hop scenario setup and link usages. A solid line means that link is used primarily (usage in %). A dotted line means the link is occasionally used. The number associated to node is the node address.

Our next scenario was a two-hop setup with 1 relay and 16 sensors. This setup was meant to exam the dynamic routing performance. The result given in Figure 13 shows how well the dynamic link-state routing performs. The second column of Table IV shows the PRR results. A node mostly sends data to its closest SINK/relay, and in case of link fluctuation, almost all of them have tried the alternative route.

The last scenario was a 4-hop setup with 3 relays and 9 sensors. This setup was meant to exam the scalablity of routing protocol. The setup scenario is shown in Figure 14. PRR results are in the third column of Table IV. Results are good and it shows that scheduling works.

V. CONCLUSION

In this paper we report the design and implementation of a protocol suite for a WSN application which measures the instantaneous environmental acoustic noise in a given area. The protocol features synchronization, link-state routing, and can be deployed/retrieved quickly. A good packet delivery ratio is achieved by carefully adjusting the timing and link-state parameters. The most significant unique feature of our design is the support of multi-hop communications, which makes large-scale noise measurement possible. This feature has not been seen in any peer solution.

Table IV
2-HOP AND 4-HOP LINK-STATE PERFORMANCE

Node addr	PRR 2-hop	PRR 4-hop
5506	0.957	-
5501	-	0.984
5502	-	0.969
5503	-	0.957
5001	0.983	0.978
5002	0.971	0.975
5003	0.977	0.977
5004	0.975	0.966
5005	0.971	0.967
5006	0.975	0.965
5007	0.979	0.947
5008	0.973	0.956
5009	0.975	0.956
500a	0.977	-
500b	0.975	-
5501	0.952	-
5502	0.889	-
5503	0.927	-
5504	0.969	-
5505	0.979	-

REFERENCES

[1] B. Raman and K. Chebrolu, “Censor networks: a critique of ”sensor networks” from a systems perspective,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 75–78, 2008.

[2] European Commission. (1996) Green paper on future noise policy. com (96) 540 final, november 1996. Official Journal of the European Communities. [Online]. Available: http://ec.europa.eu/environment/noise/pdf/com_96_540.pdf [Accessed: 2011-05-30]

[3] WHO. Occupational and community noise. World Health Organization. [Online]. Available: http://www.who.int/peh/Occupational_health/OCHweb/OSHpages/OSHDocuments/Factsheets/noise.pdf [Accessed: 2011-05-20]

[4] European Commission. (2002, 07) Directive 2002/49/ec of the european parliament and of the council of 25 june 2002 relating to the assessment and management of environmental noise. Official Journal of the European Communities. [Online]. Available: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2002:189:0012:0025:EN:PDF [Accessed: 2011-05-30]

[5] S. Santini, B. Ostermaier, and A. Vitaletti, “First experiences using wireless sensor networks for noise pollution monitoring,” in *REALWSN ’08: Proceedings of the workshop on Real-world wireless sensor networks*. New York, NY, USA: ACM, 2008, pp. 61–65.

[6] L. Filipponi, S. Santini, and A. Vitaletti, “Data collection in wireless sensor networks for noise pollution monitoring,” in *Proceedings of the 4th IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS’08)*, Santorini Island, Greece, Jun. 2008.

[7] Metravib.com. Wed007 noise dosimeter exposimeter. Website. [Online]. Available: http://www.01db-metravib.com/environment.13/products.16/wed007.460/?L=1 [Accessed: 2011-05-20]

[8] SoundEar. Soundear pro noise measuring system. [Online]. Available: http://www.soundear.com/images/stories/produktblade/SoundEarPRO_PP_UK.pdf [Accessed: 2011-05-20]

[9] APL Systems. (2011, 05) Apl aures. APL Systems. [Online]. Available: http://www.apl.fi/index.php?page_id=4277 [Accessed: 2011-05-20]

[10] I. Hakala, I. Kivela, J. Ihalainen, J. Luomala, and C. Gao, “Design of Low-Cost Noise Measurement Sensor Network: Sensor Function Design,” in *Proc. of The First International Conference on Sensor Device Technologies and Applications (SENSORDEVICES 2010)*, July 2010.

[11] *IEEE standard Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Std.

[12] I. Hakala and M. Tikkakoski, “From vertical to horizontal architecture: a cross-layer implementation in a sensor network node,” in *InterSense ’06: Proceedings of the first international conference on Integrated internet ad hoc and sensor networks*. New York, NY, USA: ACM, 2006, p. 6.

[13] I. Hakala and C. Gao, “Sync2sink: combining synchronization and routing for ieee 802.15.4-based sensor networks,” in *Technical Report*, 2009.

[14] C. Gao and R. Jäntti, “A reactive power-aware on-demand routing protocols for wireless ad hoc networks,” in *Proc. IEEE VTC 2004 Spring*, Milano, Italy, May 2004.

[15] K. Srinivasan and P. Levis, “RSSI is Under-Appreciated,” in *EmNetS, 2006*, 2006.

[16] R. Maheshwari, S. Jain, and S. R. Das, “A measurement study of interference modeling and scheduling in low-power wireless networks,” in *SenSys ’08: Proceedings of the 6th ACM conference on Embedded network sensor systems*. New York, NY, USA: ACM, 2008, pp. 141–154.

[17] N. Boughanmi, Y. Song, and E. Rondeau, “Wireless networked control system using ieee 802.15.4 with gts,” in *in proc. 2nd Junior Researcher Workshop on Real-time Computing (JRWRTC) 2008*, 2008.

[18] A. Bachir, D. Barthel, M. Heusse, and A. Duda, “Hidden nodes avoidance in wireless sensor networks,” in *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, vol. 1, June 2005, pp. 612–617 vol.1.

Tiered Wireless Sensor Network Architecture for Military Surveillance Applications

Louise Lamont, Mylène Toulgoat, Mathieu Déziel
Communications Research Centre
Ottawa, Canada
Email:{louise.lamont, mylene.toulgoat, mathieu.deziel}@crc.gc.ca

Glenn Patterson
Newtrax Technologies Inc.
Montreal, Canada
email: gpatterson@newtrax.com

***Abstract*—This paper presents a novel tiered sensor networking architecture that employs advanced Wireless Sensor Network (WSN) technologies for military operations. This architecture results in an agile surveillance system with a focus on improved operational flexibility and usability. Performance measurements using an in-house simulator are provided using two different scenarios to demonstrate the system's great agility and expandability, operating from possibly a small-scaled single cluster to a network of many chained hop-to-hop connections offering a large coverage area.**

Keywords: ***Tiered Network Architecture; Scalability; deployments; field trial.***

I. INTRODUCTION

The military areas of operation are becoming less contiguous, creating broad surveillance areas that are increasingly more difficult to monitor. The security and force protection of observation posts, for instance, face particular challenges especially when relocation may not be an option due to the requirements of the mission. In addition, the more urban deployment locations of modern military operations include buildings and other man-made structures that block lines of sight creating challenges for reconnaissance systems.

To meet the needs of contemporary deployments, the military requires a sensor system that can detect, classify, and localize hostile forces 24 hours a day in all weather conditions. This system must enhance the surveillance of both critical terrain and nomadic installations to support the monitoring of cease-fire lines, demilitarized zones, encampments, and other high-value assets. As such, the sensor system should be able to reduce the operators' workload while providing greater persistence, thereby freeing up troops for other tasks.

Conventional platform-based military sensor surveillance systems are usually large and expensive, requiring substantial manpower to operate and monitor [1]-[4]. These wireless systems have the ability to sense phenomena from their surrounding environment and communicate the gathered data to a base unit or gateway where the information is sent via long-haul communication to a command and control unit. The deployment requires that the sensors be placed strategically at a certain distance to the gateway to ensure that the sensor nodes are in line-of-sight (LOS) to the gateway or the base station. This deployment configuration results in limited coverage and single point of failures when deployed in complex terrain. Sharing detections and validation of events between sensor nodes is non-existent, which can result in high-levels of false alarms. The application of these systems is limited to predefined and fixed monitoring tasks. For fast and effective deployment, the usability of such systems is very constrained.

Wireless Sensor Networks (WSN) have gained popularity particularly with the proliferation in Micro Electro-Mechanical Systems (MEMS) which have made possible the use of large networks of small wireless sensors that are inexpensive compared to the traditional sensors. Through distributed coordination, WSN are envisioned to enhance situational awareness and improve the effectiveness of military operations. This new generation of WSNs consists of collaborative sensing nodes that are equipped with many transducers, a processor, memory, batteries, and a radio that supports the formation of an ad hoc network for extended communication coverage [5]-[7]. Sensor nodes communicate in a multi-hop fashion to reach a gateway. This type of architecture can be referred to as a planar wireless sensor network. The gateway provides wireless long-haul connectivity between the sensor nodes and the backend command and control station. The gateway bridges the sensed data and alarms to a remote user using beyond line-of-sight (BLOS) communication. The advantages of having the sensor nodes communicate in an ad hoc fashion are that the network is more robust, link redundancy increases system reliability, and sensor nodes can be deployed more rapidly. Additionally, the sensor nodes can perform cross-cueing to validate events before sending the information to the gateway, hence reducing the number of false alarms. The drawback of transmitting the sensory data in a multi-hop fashion is that the throughput per node falls asymptotically with the number of nodes N as $O\sqrt{(1/N)}$. Hence, when large areas need to be monitored, the number of sensor nodes as well as the number of hops to reach the gateway increases resulting in a decreased communication throughput and an increased delay.

A tiered networking architecture can be used to facilitate scalability and address this problem. A number of gateways

The work described herein was funded by Defence Research and Development Canada (DRDC.)

can be deployed and the sensor nodes can associate with the closest gateway to form clusters [8-10]. The gateways form the second level of the hierarchy and also form an ad hoc network. When using clustering techniques, one can reduce the number of hops required to reach the gateways and decrease the bottleneck around the gateway. Based on this motivation, we designed and implemented a tiered embedded WSN that exhibits a hierarchical networking architecture that we called Self-healing Autonomous Sensor Network (SASNet).

The SASNet system tackles the fundamental requirements of deployment effectiveness, usability, scalability, reliability, robustness and offers an agile surveillance system with a focus on improved operational flexibility and usability for military applications. SASNet aims at providing a holistic solution that facilitates rapid network deployment and concept of operations.

The remainder of this paper is organized as follows: Section 2 gives an overview of the SASNet network architecture. Section 3 covers the deployment effectiveness and usability. In section 4, the performance of the tiered network is discussed as well the limitations of the network architecture. Section 5 provides the conclusion.

II. SASNET ARCHITECTURE

SASNet is a tiered embedded wireless sensor network that exhibits a hierarchical networking architecture, containing: low cost disposable sensors, the sensor nodes, for extended monitoring coverage; resource-rich specialized nodes, the fusion nodes, for aggregation, database and application processing; and a management node for the command and control by a remote operator.

A. SASNet Hardware

The sensor nodes form the level 1 tier of the network. A sensor node is composed of a transducer board for sensing, an ultrasound board for localization, and a Newtrax WN-200 board [11] for communication as shown in Figure 1. The sensor nodes communicate with each other on a non-IP network in non-standard frequency bands anywhere between 300MHz and 1, 100MHz. They use also this band to communicate with the fusion node belonging to their cluster. The communication board supports data rates in the order of kbps.

The fusion nodes form the level 2 tier of the network. The fusion node hardware is shown in Figure 2. The fusion node consists of a TS7800 ARM embedded computer, a BU-353 GPS receiver, and three radios:

- The first radio operates within the tier-1 network to communicate with the sensor nodes in the lower UHF band.
- The second radio is a more capable radio (in term of data rate) using IEEE 802.11b. It is used to communicate using IP with other fusion nodes at the level 2 tier network in the higher UHF band at a maximum rate of 11Mbps.
- The third radio is a WiMax link between the fusion nodes and the management node. It supports data rates in the order of tens of Mbps.

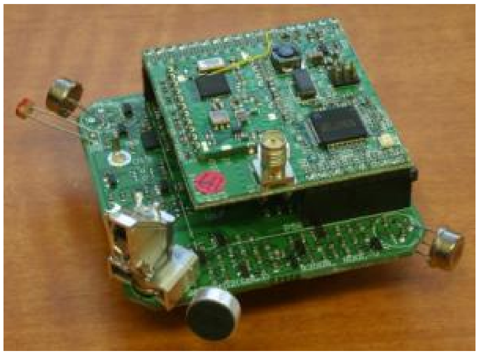


Figure 1. Sensor Node casing.

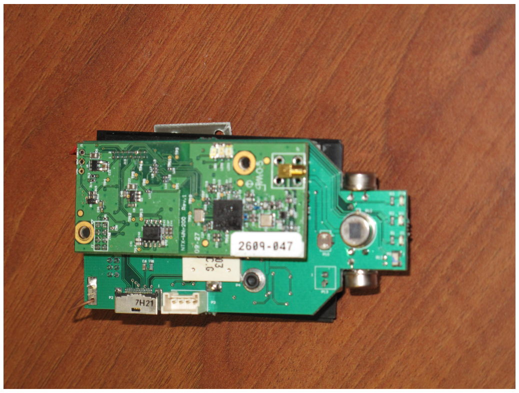


Figure 2. Fusion Node Hardware.

The TS7800 ARM holds the database software, the fusion node application software, and the level 2 tier routing software. The database is contained in the flash memory.

The management node is at the level 3 tier of the network. The management node hardware consists of a laptop computer and a WiMax radio operating in the 5.8GHz band. The laptop holds the Graphical User Interface (GUI) software.

B. SASNet Networking Archirecture

The SASNet hierarchical networking architecture is depicted in Figure 3. The sensor nodes lie at the first level (tier-1) of the hierarchy, where they perform basic operations and provide extended monitoring coverage. Sensor nodes are equipped with on board transducers such as acoustic, seismic, passive infra-red (PIR), magnetic and piezo-electric. They can detect the event of interest, validate the detection by performing cross-cueing, and facilitate classification. Each sensor node in the network acts as a router, forwarding data packets for its neighbor nodes. They form an ad hoc network on the fly and support a single radio interface for

Copyright (c) The Government of Canada, 2011. Used by permission to IARIA. ISBN: 978-1-61208-144-1

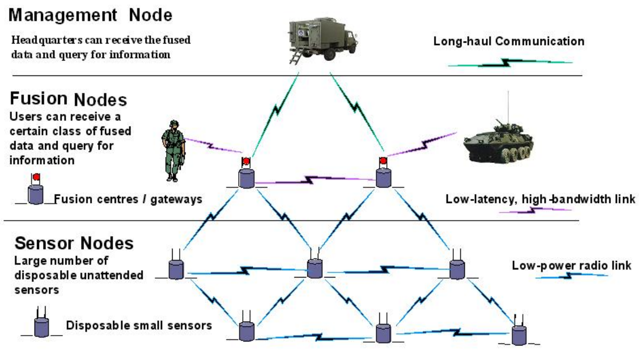

Management Node
Headquarters can receive the fused data and query for information
Long-haul Communication
Fusion Nodes
Users can receive a certain class of fused data and query for information
Fusion centres / gateways
Low-latency, high-bandwidth link
Sensor Nodes
Large number of disposable unattended sensors
Low-power radio link
Disposable small sensors

Figure 3. SASNet Operational Architecture Overview.

bi-directional communication between the sensor nodes and the fusion node.

At the second level (tier-2) of the hierarchy, the fusion nodes provide a more comprehensive function such as database synchronization, cluster formation, application logic formation, and commanding. The fusion nodes receive information requests from users, keep track of command/response queries, task the sensor nodes, aggregate information, and store history of events that have occurred in the area covered by the fusion node. The fusion nodes can also act as actuators in the network, for example, to trigger an onboard or nearby camera to obtain near real time imagery. Unlike the typical WSN, fusion nodes at the second level of the hierarchy also form an ad hoc network enabling extended coverage for larger deployment support. They are equipped with multiple radio interfaces for communication with the sensor nodes and other fusion nodes and for long-range data communications to reach the BLOS management node. The sensor nodes and fusion nodes form clusters that interconnect through capable fusion nodes to construct an unattended ground sensor system.

The management node (MN) at the third level provides the global view of the system for application, for operational control, and for system management. The fusion node uses a long-haul communication link to communicate with the management node. Authorized users can flexibly access the system from the fusion nodes at level 2 or from the management node at level 3. Using a handheld device, a laptop or a station PC, a user with proper authorization can query and subscribe to events, receive the alerts, and view histories of the system activities.

III. DEPLOYMENT EFFECTIVENESS AND USABILITY

SASNet employs the concept of a sensor toolbox allowing for the assigned users to rapidly deploy the system. The sensor toolbox is a lightweight piece of equipment in which the soldier finds necessary tools to instrument an area of interest where remote surveillance tasks are conducted. The toolbox typically contains:

- Multiple sensing nodes supporting multiple sensors per unit (motion, vibration, sound, magnetic field, etc) and a communication module.
- One or multiple ”fusion nodes” that manage the clusters, each consisting of a group of sensor nodes in an area of interest performing one common or multiple sensing tasks.
- One advanced sensor node per cluster, for example an EO sensor, capable of local image analysis.
- Handheld device unit(s) (HHD) used for deploying the network, tasking a cluster, and monitoring activities.

During the deployment, the fusion nodes are placed first and the user must ensure that they communicate with the management node. The fusion nodes are equipped with a GPS receiver and send their position to the management node and the hand-held device. The fusion nodes can be connected directly to the management node using long-haul communication or to another fusion node using ad hoc routing technology when long-haul communication is not available on all fusion nodes. Once the fusion nodes have been deployed successfully, the sensor nodes are deployed and associate with the nearest fusion node to form clusters.

During the deployment, the hand-held device user needs to quickly establish and verify the desired network communication coverage and effectiveness. As the nodes are being deployed, LEDs on the sensor nodes indicate if a network connection has been established to the fusion node or to another sensor node. Network formation is done autonomously. Once the sensor nodes in the system have calculated their position, completed auto-configuration and formed the network, users that are authorized to configure the system view the sensor nodes and the network status on their monitoring interface. Once all sensor nodes in a cluster have been deployed, the network formation of the cluster is verified by the user by tasking the system. After the nodes placement and tasking from the hand-held device have been completed, the task is sent to the management node using the database synchronization scheme. The operator can then add additional tasks if necessary. The hand-held device user can retreat to a covered area that may be a distance away from the monitored area.

Authorized users can formulate the query and/or the event subscription with desired contextual information such as period of time, geographical area, type of event to trigger on and an action to perform. An event subscription sets up the actions following a detection trigger, including further detection and confirmation of the event, sending an alarm for the event trigger, etc. A query can retrieve the past activity reports in the system. Users and operators can be at a distance of a few hundred meters to up to 1 km, or in some cases to up to 10 km from the instrumented area.

ISBN: 978-1-61208-144-1

The user interface on the hand-held device allows navigation through the system information and quick setup of event subscription, tasking, and queries. The user can easily select the geographical area of interest, a period of interest in time, a pre-configured type of event and action such as classification of detected vehicle and alarm with event image reporting.

The operator at the command and control station has more time and can formulate more complex tasks that can be dispatched to the right area and nodes in the network. The useful basic patterns may include as examples, motion pattern detection such as stop, turning direction, and detection contextual information setup such as time, location, and count.

Event subscriptions and associated tasks can be flexibly modified during the operation to monitor new events or to obtain different contextual information of the monitored events. Upon the subscribed event triggering, the network performs the actions as assigned in the corresponding task to collect, coordinate, aggregate the detection information and to send report/alarm to the required destination device(s).

After completion of the SASNet operation for a certain mission, the system is decommissioned and reusable equipments are collected and placed back into the SASNet toolbox.

IV. PERFORMANCE MEASURES OF THE NETWORK

In this section, we use a simulation environment to evaluate the performance of the SASNet system in two different topologies.

The simulation environment is based on the discrete-event model using the OmNet++ architecture and runs the same protocol stack firmware as run on the nodes to simulate a wireless mesh network. The simulator can be easily configured via configuration scripts and results are obtained in the form of reports and detailed event lists. It is also possible to visualize the network topology at any time using Matlab/Octave.

The performance of the network was analyzed using simulation results for two deployment scenarios, namely a single-hop cluster and a multi-hop linear network. We considered the following performance metrics: average one-hop latency, average single node throughput, average rate of packet loss.

A. Scenario 1: Single-Hop Cluster Network

The first scenario represents a cluster of sensor nodes that connect directly to a fusion node in a single hop. We deployed clusters of sizes $N = \{5, 6, 7, 8, 9, 10, 12, 14, 15\}$ nodes in a star-topology where the fusion node is placed at the centre of a circle of radius 35m and the sensor nodes are placed at equal distances along the perimeter of the circle. This represents a worst-case congestion scenario as all nodes are in RF range of one another and therefore can experience interference from any other node in the network.

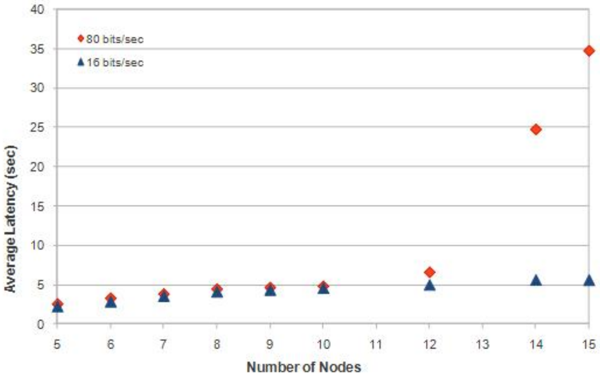


Figure 4. Average Latency for one-hop cluster networks where each sensor node sends data at 16bits/sec and 80bits/sec respectively.

Once the cluster networks form in the simulator, we send 10 byte packets from every sensor at the same instants at two different rates: every second and every five seconds. This gives a packet rate of 16 bits/second and 80 bits/second from each sensor node respectively. We transmit packets continuously over a 4500 second interval.

In Figure 4, we show the average packet latency of the single-hop cluster networks of varying size at both packet rates. We see that when the sensor nodes transmit packets every 5 seconds, the latency per packet is between 2 and 6 seconds as the cluster size increases. For cluster sizes less than ten nodes when we send packets every second, the latency on a typical node is the same as when we send packets every five seconds. However for cluster sizes larger than 12 nodes, we see that the congestion in the network at 80 bits/second causes the latency to sharply increase.

In Figure 5, we show the average packet loss ratio for both sensor node traffic rates. Here we see that for clusters of less than 8 nodes and packet generation every 5 seconds (16 bits/sec) we get very low packet loss. However as the cluster size increases, the congestion of the medium sharply pushes the loss rate above 20%. For packets generated every second (80 bits/sec), the packet loss rate increases linearly above 10% for clusters of size 6 and greater. Sensor networks are known to exhibit severe packet losses due to congestion of a shared medium when supporting high data rates and this is often referred to as the Data Implosion Problem when many nodes simultaneously transmit to a centralized fusion node. This problem is especially exacerbated here because all nodes are in interfering RF range with each other and were forced to connect directly to the fusion node to create a worst-case scenario. The performance could be improved by deploying several fusion nodes to effectively share the network load.

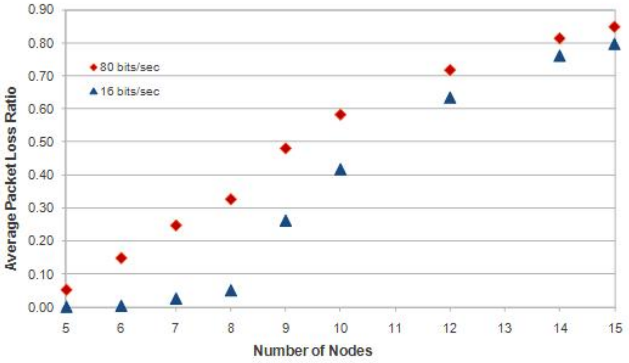


Figure 5. Average Packet Loss Ratio for one-hop cluster networks where sensor nodes send data at 16bits/sec and 80bits/sec respectively.

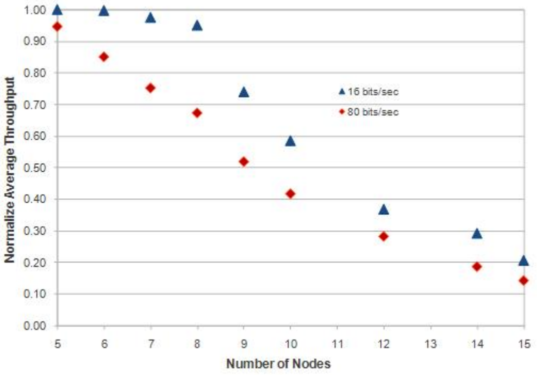
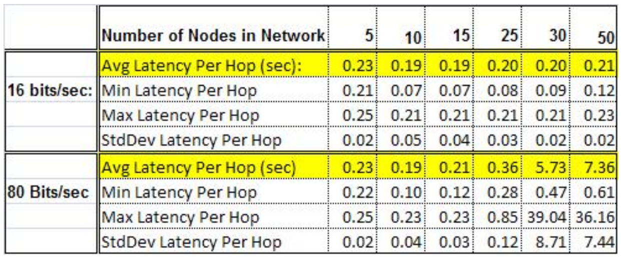


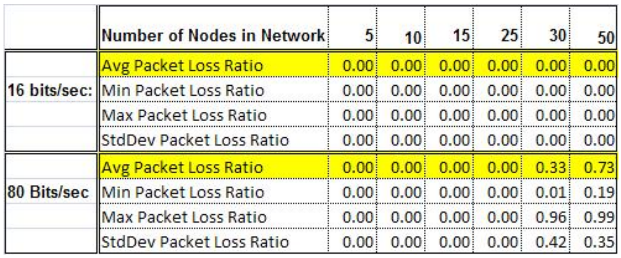
Figure 6. Average node throughput for one-hop cluster networks where nodes send data at 16 bits/second and 80 bits/second respectively normalized by that packet generation rate.

Table I
LATENCY STATISTICS FOR MULTI-HOP LINEAR NETWORKS OF VARIOUS LENGTHS AT DATA RATES OF 16 BITS/SEC AND 80 BITS/SEC.



	Number of Nodes in Network	5	10	15	25	30	50
16 bits/sec:	Avg Latency Per Hop (sec):	0.23	0.19	0.19	0.20	0.20	0.21
	Min Latency Per Hop	0.21	0.07	0.07	0.08	0.09	0.12
	Max Latency Per Hop	0.25	0.21	0.21	0.21	0.21	0.23
	StdDev Latency Per Hop	0.02	0.05	0.04	0.03	0.02	0.02
80 Bits/sec	Avg Latency Per Hop (sec)	0.23	0.19	0.21	0.36	5.73	7.36
	Min Latency Per Hop	0.22	0.10	0.12	0.28	0.47	0.61
	Max Latency Per Hop	0.25	0.23	0.23	0.85	39.04	36.16
	StdDev Latency Per Hop	0.02	0.04	0.03	0.12	8.71	7.44

Table II
PACKET-LOSS STATISTICS FOR MULTI-HOP LINEAR NETWORKS OF VARIOUS LENGTHS AT DATA RATES OF 16 BITS/SEC AND 80 BITS/SEC.



	Number of Nodes in Network	5	10	15	25	30	50
16 bits/sec:	Avg Packet Loss Ratio	0.00	0.00	0.00	0.00	0.00	0.00
	Min Packet Loss Ratio	0.00	0.00	0.00	0.00	0.00	0.00
	Max Packet Loss Ratio	0.00	0.00	0.00	0.00	0.00	0.00
	StdDev Packet Loss Ratio	0.00	0.00	0.00	0.00	0.00	0.00
80 Bits/sec	Avg Packet Loss Ratio	0.00	0.00	0.00	0.00	0.33	0.73
	Min Packet Loss Ratio	0.00	0.00	0.00	0.00	0.01	0.19
	Max Packet Loss Ratio	0.00	0.00	0.00	0.00	0.96	0.99
	StdDev Packet Loss Ratio	0.00	0.00	0.00	0.00	0.42	0.35

In Figure 6, we show the effective average throughput of a node in the cluster networks for the two different packet generation rates. To compare the results between the two packet rates, we have normalized each curve by the packet generation rates. For packets generated every 5 seconds, the throughput stays very close to the maximum achievable (16 bits/second) for cluster sizes below 9 nodes and then quickly decreases as the cluster size is increased towards 15 nodes. When packets are generated every second, the throughput drops off for all cluster sizes above 5 nodes in a linear manner.

B. *Scenario* 2*: Multi-Hop Linear Network*

The next set of scenarios studied were multi-hop linear networks of fixed distance between consecutive nodes with the fusion node at the head of the network. In all scenarios, the fixed inter-node distance was 70m between all nodes. At 15dBm transmission power, this creates a network such that any node is only in RF range of its two adjacent neighbors. We experimented with network lengths of $N = \{5, 10, 15, 25, 30, 50\}$ nodes ($N - 1$ sensor nodes and 1 fusion node).

For each N, we studied the performance of the linear network by sending one 10 byte packet from every sensor at the same instants at two different rates: every second and every five seconds. This gives a packet rate of 16 bits/second and 80 bits/second from each sensor node respectively. We transmit packets continuously over a 4500 second interval.

Compared to the star-topology, the linear network where each node is only in RF range of two adjacent neighbors will experience much less congestion of the RF medium. However as we shall discuss later, linear networks have their own congestion problems. In Tables I, II, and III we show the statistics for the latency, packet-loss, and normalized throughput for both data rates with linear networks of various number of nodes.

If we look at the performance at 16 bits/sec data rate, we see that we can successfully deliver all packets generated with optimal latency (about 0.2 sec/hop). When the data rate is increased to 80 bits/sec, for $N \leq 25$ we can deliver all packets with optimal or near optimal latency (0.356 sec/hop in a 25 node network). However for $N \geq 25$ nodes, we see that the performance begins to degrade. In particular, we see nodes close to the fusion node with latencies near 40 seconds for both 30 and 50 node linear networks with corresponding packet-loss rates of 95% and 99.5% respectively.

Unlike the star-topology network previously examined, the performance degradation as the length of the linear

Table III
THROUGHPUT (NORMALIZED BY THE DATA RATE) STATISTICS FOR MULTI-HOP LINEAR NETWORKS OF VARIOUS LENGTHS AT DATA RATES OF 16 BITS/SEC AND 80 BITS/SEC.

	Number of Nodes in Network	5	10	15	25	30	50
16 bits/sec:	Avg Normalized Throughput	1.00	1.00	1.00	1.00	1.00	1.00
	Min Normalized Throughput	1.00	1.00	1.00	1.00	1.00	1.00
	Max Normalized Throughput	1.00	1.00	1.00	1.00	1.00	1.00
	StdDev Normalized Throughput	0.00	0.00	0.00	0.00	0.00	0.00
80 Bits/sec	Avg Normalized Throughput	1.00	1.00	1.00	1.00	0.67	0.27
	Min Normalized Throughput	1.00	1.00	1.00	1.00	0.04	0.01
	Max Normalized Throughput	1.00	1.00	1.00	1.00	0.99	0.81
	StdDev Normalized Throughput	0.00	0.00	0.00	0.00	0.42	0.35

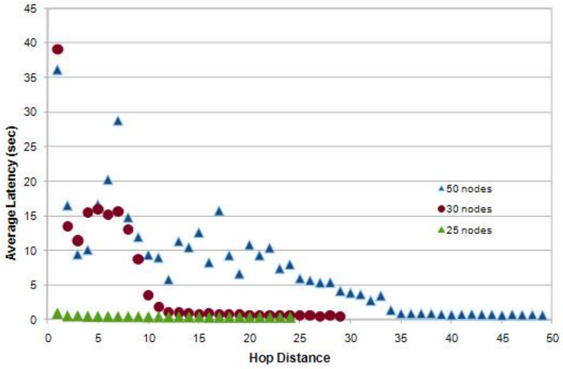


Figure 7. The Average Latency for a multi-hop linear network plotted as a function of hop distance to the fusion node for linear networks of length 25, 30, 50 nodes. Data packets are generated at each node at a rate of 80 bits/sec.

network increases is the result of buffers being filled at down stream nodes close to the fusion node. Basically, the nodes closer to the fusion node must route all the traffic generated further upstream and over an extended period of time, their buffers become full and they will drop packets when this happens. Again, this is a manifestation of the data implosion problem of large-scale sensor networks.

In fact, for $N \geq 30$ the average values in Tables I, II and III are somewhat statistically misleading on account of this data implosion. We see in these tables, that for more than 30 hops in the network, the variance on the latency, packet-drop, and throughput is on the same order of magnitude as the average value. In Figure 7, 8 and 9 we plot the average latency, packet-drop, and throughput of each node in a linear network to illustrate the relationship between the hop-distance to the fusion node and the node performance for $N = \{25, 30, 50\}$ nodes.

Figure 7 demonstrates well the rapidly increasing per-hop latency as packets from downstream nodes get congested in the buffers of nodes closer to the fusion node. For $N = 30$ nodes, the latency starts to increase around the 11^{th} node and gets rapidly worse as we approach the fusion node, and thus about $1/3$ of nodes experience latencies far from the optimal value of about 0.2 sec/hop. For $N = 50$ nodes, the latency starts to increase around the 34^{th} node and gets rapidly worse as we approach the fusion node and so about $2/3$ of the nodes experience latencies far from the optimal value.

Looking at Figure 8, we see that as expected the packet-loss is much higher for the nodes closer to the fusion node. In fact for 50 node networks, the packet-loss is actually relatively low for hop-distances greater than 35 hops. At hop 34, the packet loss sharply swings to almost 100%. This same phenomenon happens at the 10^{th} hop for a 30 node linear network. However, we should note here that if the network were allowed to run for a longer interval and continue to generate packets every second, we could expect that these down-stream sensor nodes farther from the fusion node would one-by-one begin to experience exponentially increasing congestion and at this point there would be essentially no throughput in the network. Also note that contrary to our intuition, the node closest to the fusion node does not have the highest packet-loss rate. For example when we have a 50 node linear network, there is a region of about 30 nodes starting at the 5^{th} node that has almost 100% packet-loss rates. The first 4 nodes after the fusion node have packet-loss rates between 75% and 90%. This phenomenon can be explained as follows. At first, congestion affects the nodes closest to the fusion node. However, as their buffers fill up and they drop packets this phenomenon spreads one node at a time down the network towards the last node in the network. After a certain point, there is so much congestion due to packet transmissions and retries that the packets from the last $1/3$ of the network no longer even reach the nodes closest to the fusion nodes and so the load on these nodes is slightly less than the nodes in the middle of the network and they can successfully route more packets.

We also note here that the degradation of linear network performance at 80 bits/sec data rate is also a function of the time-window over which the packets are generated. For shorter windows, the linear network can successfully handle all the packets. For example, with a linear network of 50 nodes we can generate packets for 78 seconds (a relatively short window) continuously before the nodes close to the fusion node begin to experience buffer congestion and the resulting high packet-loss. For a 30 node network, we can support the 80 bit/sec packet rate for about 300 seconds before buffer congestion and the resulting high packet-loss. Indeed, we can imagine many sensor network applications where there is a large but short-lived burst of data (for example a vehicle moving quickly by all the sensor nodes) and so our networks could support the data generated in these cases. The durations we have tested in our simulations are really meant to give a worst-case and long-term picture of the network performance.

ISBN: 978-1-61208-144-1

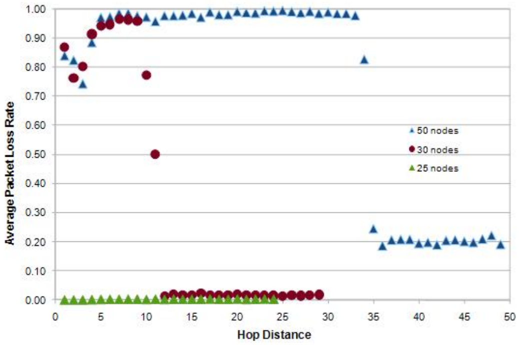


Figure 8. Average Packet Loss as a function of hop-distance to the fusion node for linear networks of length 25, 30, 50 nodes. Data packets are generated at each node at a rate of 80 bits/sec.

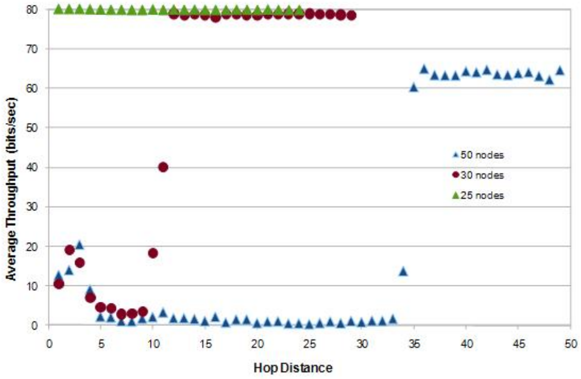


Figure 9. Average Throughput as a function of hop-distance to the fusion node for linear networks of length 25, 30, 50 nodes. Data packets are generated at each node at a rate of 80 bits/sec.

V. CONCLUSION

In this paper we have presented a tiered network architecture that results in an agile surveillance system with a focus on improved operational flexibility and usability. It was shown that two topology families, pure star and pure linear, are special-cases of what we could reasonably expect to form in an actual deployment of sensors in the field. In many applications it is reasonable to expect that not all sensor nodes will be in range of the fusion node and there will be some leap frogging because some sensors will be in range of 3 or more devices and so a pure linear network will form. Thus, what might be expected to form in the field would be closer to a hierarchical tree topology which will alleviate the buffer congestion seen in long linear networks and the wireless medium congestion seen in larger star-topology networks.

REFERENCES

[1] http://defense-update.com/products/f/falcon-watch.htm

[2] http://defense-update.com/products/m/MIS.htm

[3] http://defense-update.com/products/s/SCOUT-UGS.htm

[4] http://defense-update.com/products/r/rembassII.htm

[5] http://www.selex-sas.com/SelexGalileo/EN/Business/Products/Advanced_Sensors/index.sdo

[6] http://www.sownet.nl/download/T-Node_product_sheet.pdf

[7] http://www.exensor.com/index.php?s=systems&k=products&ok=yes

[8] M. Zhang, J. Song, and Y. Zhang, ‘Three-Tiered Sensor Networks Architecture for Traffic Information Monitoring and Processing,” *Intelligent Robots and Systems (IROS 2005)*, 2005.

[9] Y. Ye, V. Hilaire, A. Koukam, and W. Cai, “A Cluster Based Hybrid Architecture for Wireless Sensor Networks,” Information Science and Engineering (ISISE ’08), 2008.

[10] O. Gnawali, B. Greenstein, K. Jang, A. Joki, and J. Paek, “The Tenet Architecture for Tiered Sensor Networks”, *Proc. the 4th international conference on Embedded networked sensor systems*, 2006.

[11] http://www.newtrax.com/

ISBN: 978-1-61208-144-1

Virtual Ground Truth in Vehicular Sensing Experiments: How to Mark it Accurately

Girts Strazdins*†, Artis Mednis*†, Reinholds Zviedris*†, Georgijs Kanonirs*, Leo Selavo*†
*Digital Signal Processing Laboratory
Institute of Electronics and Computer Science
14 Dzerbenes Str., Riga, LV 1006, Latvia
†Faculty of Computing
University of Latvia
19 Raina Blvd., Riga, LV 1586, Latvia
Email: {firstname.lastname}@edi.lv

***Abstract*—Road surface quality monitoring is an important requirement for efficient, safe and comfortable transportation. However, the data collection is made difficult by the scope of the data source. Therefore, participatory sensing is a promising approach for road damage assessment. We are developing a vehicular participatory sensing application using Android smart-phones for pothole detection. This paper describes lessons learned from our field tests, which have exposed the deficiencies in terms of collected data quality. Nevertheless, the tests provide invaluable experience for planing future field tests and improvements to the test execution procedure for vehicular sensing researchers. Based on empirical and analytical results, we conclude, that semi-automated ground-truth reference point recording by a human observer in a moving vehicle while doing the actual data collection is imprecise as a consequence of multiple technical and human factors. We also discuss the motivation, why careful pothole position marking and categorization by walking along the test track is capable of providing highly accurate ground-truth.**

Keywords-real-world experiment experience; data quality; vehicular sensing; participatory sensing; Android OS.

I. Introduction

Road surface damage (potholes, bumps, gaps, etc.) are a serious issues causing distractions for safe and comfortable transportation. Both drivers and road maintainers are interested to fix the problems as soon, as possible. To fix them, they first have to be identified. Centralized road inspection is difficult due to the scope of road infrastructure. Several pothole reporting systems already exist, including web sites, such as [1] [2]. However, they rely entirely on manual reports of individuals. The requirement for manual human interaction implies the low report rate, which, we believe, can in turn be increased dramatically by automated pothole detection and reporting systems.

We are developing a pothole detection system based on participatory sensing using Android smartphones in driving vehicles. Accelerometers and Global Positioning System (GPS) are used to detect and geotag potholes encountered during the ride. The system is envisioned to be added as a service to navigation systems, such as Waze [3], which people use on daily basis. That would enable large-scale pothole detection, reports to responsible authorities as well as publicly available pothole visualization map. Systems architecture and general principles are described in the Section II.

To evaluate our approach, a set of field tests (described in Section III) with multiple Android-based smartphones in a driving vehicle were performed and acceleration sensor data for more than five hours of driving were collected. The analysis of the data revealed both positive and negative results. On one hand, the initial ground-truth marking methodology was deficient. On the other hand invaluable knowledge about vehicular sensing experiment planning, execution and accurate ground-truth marking, and practical experience with multiple, distinct, Android devices was gained. We believe, that our concluding recommendations (described in Section IV) will improve the quality and reduce the time investment for experimental evaluation process of vehicular sensing application researchers. We state the proposed recommendations as our main contribution in this paper.

II. System Architecture

The system (see Figure 1) consists of Android smartphones located in vehicles, acting as mobile agents in urban environment; and central server, which aggregates the data and provides web interface. 3-axis accelerometers are required for pothole detection, GPS for event geotagging and communication (either WiFi or Cellular) for data exchange with the server. The requirements are not too restrictive, as the components are available on most of Android phones.

The phone should be either fully charged or connected to a car charger, as the data collection process may require significant amount of energy. The most consuming components are the touchscreen, which can be turned off if no user interaction is required, and GPS, which can be optimized by combination of other localization methods [4].

The sensor sampling and event detection (in the particular case - potholes, but other types of events can be supported in general) is performed on the phone. Only reported events

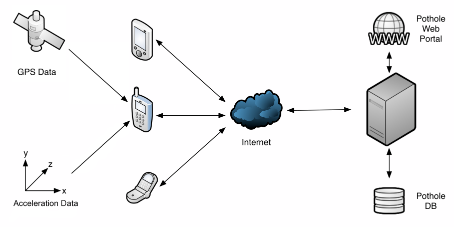


Figure 1. Our system architecture. Mobile phones collect sensor data, extract events which are exchanged with the central server, where data is aggregated and web interface is provided.

are sent to the server, without raw sensor data. The server aggregates reported data and sends back other participant collected events upon request. Hence the data transmission channel utilization is very low, and the system has high scalability, which is an important factor for platforms intended for public and large-scale usage.

The system is intended for use in urban environments, at speeds up to 70km/h ($\approx$ 37mph).

The event detection involves accelerometer data processing. We have adapted multiple algorithms from our own [5] and other researcher prior work experience. Briefly, our algorithms include vertical-axis acceleration and its standard deviation thresholding by amplitude. More in depth algorithm analysis is out of the scope of this paper.

III. Field Tests

To evaluate the approach, data collected in real-world urban environment is required. Therefore a set of field tests were performed. Data was collected by multiple Android devices in a driving vehicle. Multiple drives were performed. Additionally, a laptop with external GPS device and a microphone was used as a reference, detecting potholes by audio data processing, using RoadMic methodology, described in our previous work [6].

Online event detection and audio notification was implemented for debugging reasons. Additionally, all the raw data was recorded for offline processing and analysis, to tune and assess the detection algorithms and collected data quality.

The test track (Figure 2) was selected due to three significant features it possesses: short enough to repeat multiple laps; diverse road segments, both very smooth and very rough; located in a realistic urban environment.

A. Test setup

The experimental setup and collected data characteristics is shown in Table I. Overall, the track is 4.4km long (2.73 miles). Five different Android smartphones were used: Samsung i5700, HTC Desire, Samsung Galaxy S, HTC Desire Z, and HTC HD2. Five different test drives were performed (25th and 28th of January, 10th and 28th of February, and 24th of March, 2011), using three different vehicles: BMW 323 Touring, Mitsubishi Space Wagon, and Mazda 323F. Besides accelerometer sensor, microphone data was recorded for future use. The first and second drives contained 3 test laps each, the third - one lap. The fourth and fifth drives contained 10 laps each.

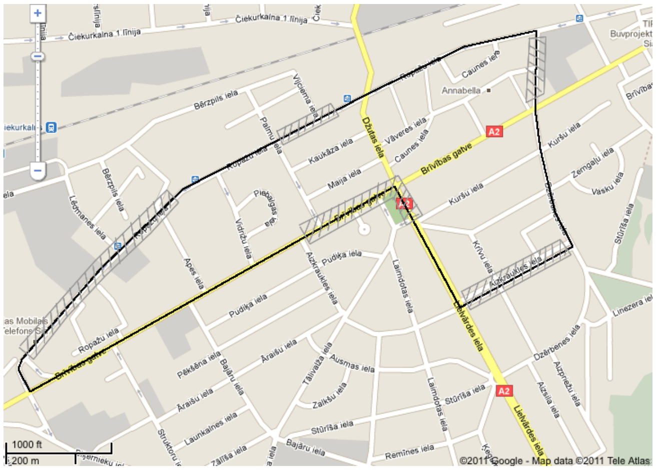


Figure 2. Used experimental test track, 4.4km long, single and multi lane streets in urban environment. Regions marked have most of the potholes.

During the first two drives no ground-truth was marked and the corresponding collected data is therefore only usable to get an impression of Android smartphone peculiarity, distinct device, vehicle and environmental factor impact on sensor data quality.

The goal of the third drive was an insight into ground-truth marking. A PC laptop with an external GPS and our custom built ground-truth marking application was used in the same driving vehicle, where Android phones were collecting accelerometer data. A human operator pressed spacebar each time he experienced a street damage initiated shake, and the software recorded local system time (with millisecond accuracy) accordingly. Offline pothole detection was performed, using multiple detection algorithms on the data collected in the third drive. The detected events were compared against ground-truth points. The results were unsatisfactory - only about 65% of the detected events were in ground-truth point vicinity. Unfortunately, it was unable to distinguish whether the source of the error is ground-truth inaccuracy, detection algorithm inappropriateness, or data set size. And the situation on the road had changed since the first drives, the ground-truth was not usable as a reference for the older data.

Therefore the fourth drive was performed, which was planned to overcome the drawbacks of the first drives. Ten laps of data collection and ground-truth recording with four different Android devices were performed, a total of 44 kilometers, more than two hours of data. Unfortunately inconsistencies in the used semi-automated ground-truth marking in the driving vehicle were discovered. It led to a conclusion, that manual ground-truth pothole marking is a

Table I
TEST DRIVE CHARACTERISTICS

Date	Vehicle	Phones	Laps	Minutes	Kilometers	Ground truth
2011-01-25	BMW 323 Touring	Samsung i5700, HTC Desire, Samsung Galaxy S	3	32	13.2	No
2011-01-28	Mitsubishi Space Wagon	Samsung i5700, Samsung Galaxy S	3	35	13.2	No
2011-02-10	Mazda 323F	Samsung i5700, HTC Desire	1	11	4.4	Yes, incomplete
2011-02-28	Mazda 323F	Samsung i5700, HTC Desire, Samsung Galaxy S, HTC Desire Z	10	131	44.0	Yes, incomplete
2011-03-24	BMW 323 Touring	Samsung i5700, HTC Desire, Samsung Galaxy S, HTC HD2	10	119	44.0	Yes, complete
Total	3	5	27	328	118.8	3

better method for accurate reference point selection. A more in-depth analysis is described in Section III-B.

The fifth drive was performed in conjunction with manual ground-truth marking. A total of 108 potholes were marked on the 4.4km long test track (Figure 4c). The points were recorded on pedestrian sidewalks and GPS signal was interrupted by high buildings. Therefore offline position correction was performed, by calculating simple perpendiculars to the mean trajectory of all the 10 laps of the fifth test drive, as shown in Figure 3. Preliminary analysis shows that collected data and marked ground-truth in the fifth drive are sufficiently accurate to perform further pothole detection algorithm evaluation.

The next two subsections describe the negative and positive inferences of the collected data.

B. The Dark Side

The offline data processing revealed several deficiencies of the fourth test drive. Semi-automated ground-truth marking mechanism seemed to be attractive due to two reasons. First of all, in such a way only the encountered potholes would be recorded. Second, it seemed a natural approach, compared to additional two hour walk with manual point-of-interest marking on the GPS device (*Walking GPS* approach [7]). As it turned out, the capabilities of semi-automated ground-truth point marking were overrated. It was imperfect both in terms of accuracy and time-efficiency.

Overall 1326 locations were marked during the 131 minute drive, shown in Figure 4a. Such a set of points cannot be used as ground-truth directly, therefore it was refined: only the points which had at least c other points (from other, distinct, laps) in their vicinity (no further than d meters) remained. A total of 273 ground-truth locations (21% of initial 1326 locations) remained after the refinement procedure with parameters $c = 6$ and $d = 5$ (Figure 4b). Although visually the refined locations correspond to expectations, more in depth analysis showed, that their usability is doubtful due to the following reasons:

1) There is no classification possibilities. All the potholes are considered equal, regardless of their actual size and significance. Multiple pothole type support in ground-truth marking software would be practically useless, as the human operator would not be able to categorize the potholes in a short period of time.
2) There is always a distance between the real pothole and the location of the button press, which is, unfortunately, undetermined and cannot therefore be eliminated by simple shift operations.
3) During a two hour drive the human ground-truth marking operator is loosing attention. Hence the last laps have less accurate ground-truth locations.
4) Lack of precisely defined methodology (when to press the button) leads to inconsistent results - a particular pothole is recorded in some laps, ignored in the rest.
5) GPS inaccuracy has a greater impact compared to Walking GPS approach, where it can be mitigated by standing a while at the same location and allowing GPS signal to stabilize.
6) The amount of shake the human perceives depends not only on the size and type of road irregularity, but also on vehicle speed and technical condition. Hence insignificant potholes might get recorded and significant ones ignored.
7) During the fourth test drive audio signal on a laptop was recorded and pothole detection by previously evaluated and reliable RoadMic [6] methodology was performed. And the results contained contradiction: while increasing algorithm threshold, detection accu-

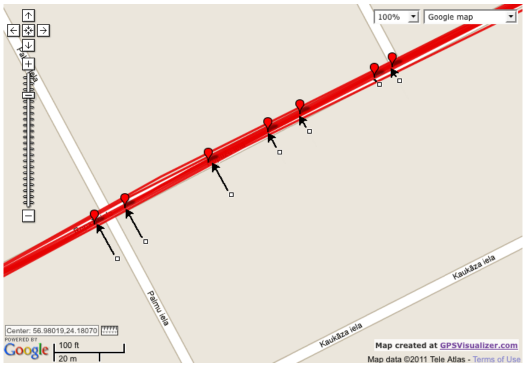
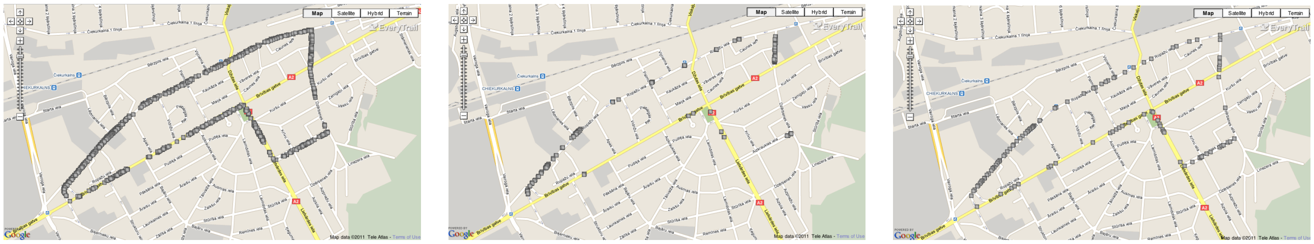

100%
Google map
Palmu iela
Kaukāza iela
Kaukāza iela
Center: 56.98019,24.18070
POWERED BY Google
100 ft
20 m
Map created at GPSVisualizer.com
Map data ©2011 Tele Atlas - Terms of Use

Figure 3. Manually marked ground-truth position improvement, by calculating perpendiculars to the test drive trajectory



(a) 1326 ground-truth points recorded semi-automatically during the 131 minute long 4th drive

(b) 273 refined ground-truth points

(c) 108 manually marked ground-truth points

Figure 4. Ground-truth, acquired semi-automatically in the fourth drive (a), its refined version (b) and manually marked ground-truth of the fifth drive (c)

racy had to increase and reach 100%, but it decreased. That caused a strong suspicion about the ground-truth accuracy.

8) Needless to say, the refinement procedure design, implementation and tuning took more time and effort than would Walking GPS approach require.

The arguments listed above led to an unpleasant conclusion - the collected ground-truth is inaccurate and cannot be used as a reference for further algorithm analysis.

Another problem was related to incorrect data storage. It was assumed, that acceleration values will not exceed $32m/s^2$ ($\approx 3.27g$). They were stored in 16-bit variables, supporting values -32.768..32.767m/s^2. However, Samsung i5700 reported values greater than $32m/s^2$ (which, we believe, is a bug of this specific device). It led to a number of overflows, which required additional time and effort.

Besides systematic problems unexpected ones were also encountered. Two of four Android devices experienced a reboot during the fourth ride. The data collection was restarted as soon as possible, but significant amount of data was lost: 47 and 16 minutes. The reboot reason is unclear, but it is probably related to incorrect OS handling of intense audio data recording, requiring large data buffers and fast flushing to SD card. In the fifth drive, audio recording was disabled. But it turned out to be a bad design decision, as it involved last-minute code modifications and consecutive bugs which resulted in partial data loss for two devices: 47 and 20 minutes. Additionally, one of the phones ran out of battery in the fourth drive, 19 minutes of data were lost.

Preliminary audio data analysis showed, that RoadMic approach cannot be transferred to Android phones directly due to dynamic range compression used, but further analysis must be performed to evaluate potential of pothole detection from audio signal using Android phones.

C. The Bright Side

Although the collected data did not satisfy all of our research needs, it was valuable for a number of reasons.

First of all, an insight into Android OS impact on sensor data collection was acquired. On one hand, Android handles a lot of processes which had to be done manually on an embedded system. It converts raw values to SI system, handles chip management, provides simple programming interface. On other hand, it limits the freedom available on customized embedded devices. Accelerometer sensor data is reported *as fast as possible*, without any guarantees of minimum or maximum latency. The achieved sampling rate was relatively low and device dependent.

The most unexpected difference between Android devices was accelerometer sensor output. Although the four devices have two common vendors (Samsung and HTC), every device had different sensor sampling rate, ranging from 26Hz up to 98Hz, see Table II. The sensor sensitivity and noisiness was also different. For comparison, we calculated standard deviation for vertical axis acceleration for the same 500 second time period from the fifth drive (fourth drive for HTC Desire Z, as it was not present on the fifth drive) for all the devices. Samsung i5700 has significantly higher deviation compared to other devices.

GPS accuracy was better than we expected, having mean Android-reported error under 7 meters with more than 7 satellites visible on average (Table II).

The diversity of the test drive environment (snow and potholes on the road varied a lot during the two month period) and setup (different vehicles) provide data for vehicle and environment variety impact assessment. But it will only be usable, when the correctness of pothole detection algorithms will be proved.

And, last but not least, the rather negative experience transforms into conclusions on how to perform the experiments in higher quality and with higher success rate.

IV. Experience and Recommendations

This section describes recommendations based on the performed vehicular sensing experiments on the Android platform.

1) Do not mark ground-truth positions in a driving vehicle: detailed motivation is described above, in Section III-B. The recommended solution is to walk along the test track with a GPS device, stay a couple of seconds at each position of interest allowing GPS signal to stabilize, record the

Table II
SENSOR DATA DIFFERENCES BETWEEN DISTINCT ANDROID SMARTPHONES. AVERAGED OVER 10 MINUTE DRIVE.

Device	Accelerometer		GPS			
	sampling rate, Hz	**Z-axis StdDev, g**	**accuracy, m**	**locations missing, %**	**visible satellites**	**avg. visible satellites**
Samsung i5700	26	0.3076	-	4	4-9	6.99
HTC Desire	52	0.1215	4.21	0	4 - 11	9.68
Samsung Galaxy S	98	0.1171	6.35	0	5 - 8	6.35
HTC Desire Z	73	0.1536	3.41	0	7 - 11	9.63
HTC HD2	47	0.1242	1.78	7	4 - 10	7.73

position, and add an event category to it. Offline correction may be required.

2) Ground-truth is temporary accurate: even in a few days situation on the road can change. Therefore ground-truth must be recorded as close to the actual data, as possible.

3) Android devices are different: a complete experimental evaluation must be performed on more than one device to get valid results.

4) There is a non-deterministic delay between actual sensor sampling and data reception in the software: this fact must be taken into account in situations where the subject is moving at significant speeds.

5) During long test drives miscellaneous errors can occur preventing the data collection: It is advisable to monitor the devices continuously.

6) Although Android is an open platform, it has remarkable hardware access restrictions compared to customized embedded platforms (sensor motes): it is reasonable to invest the saved application creation time to develop more sophisticated data processing algorithms.

7) Device power supply is an underrated problem: for each device a decision has to be made - should it require a power supply during the ride or not? And a supply must be provided if necessary.

8) Insignificant data recording may distract the essential data collection: collect just the required information. Otherwise, software bugs and hardware limitations could degrade the sensing process.

9) Different value scales cause problems with data interpretation: it is advisable to convert all the data to a unified scale. Human-readable values are preferred over raw values.

10) It is hard to get the overall understanding of large data set and location data: visualization techniques are recommended to simplify pattern perception.

11) Unsynchronized time for devices will lead to inefficient post-processing: it is advisable to synchronize time of all the devices before the experimental tests. Note, that ”Use network-provided values” under date&time settings on Android devices does not mean synchronization with NTP servers!

V. DISCUSSION

One of the main problems with ground-truth is the lack of objectiveness. If data processing and event detection system is considered as a formal logic, ground-truth serves as a set of axioms used to prove further algorithm accuracy. It is, however, hard to argue on the correctness of the ground-truth itself. Three basic requirements for ground-truth as an axiomatic system for vehicular sensing experiments:

1) *Usability* - Ground-truth must represent real environmental and road conditions
2) *Consistency* - Algorithm analysis and comparison to ground-truth should contain no contradiction
3) *Completeness* - Ground-truth should be usable as etalon for analysis of any algorithm that detects the particular event type

According to Goedel’s Incompleteness theorem, consistency and completeness cannot be guaranteed simultaneously in general case.

For Walking-GPS approach, usability follows from its definition: we mark positions of encountered real potholes on city streets. For semi-automated approach it is arguable: we mark positions, where it *feels similar to pothole*. Evaluation of usability is, however, subjective - locations of particular events are recorded by humans and represent their viewpoint.

Consistency in this context requires clear response to ”what kind of event is located at x?” for each x. If a location exists for which it is not clear, whether there is a pothole or not, it is a contradiction. Semi-automated approach suffers from such inconsistencies, as described above. Walking-GPS, and any other methods contain a certain degree of position deviance due to GPS inaccuracy. But for each recorded ground-truth point it is clear - there is a real pothole in close vicinity.

Completeness includes multiple event category support, as some of the detection algorithms might be intended for event segregation or intensity detection. Manual ground-truth marking satisfies this requirement.

An alternative approach would be to use a well-known, previously proved algorithm as a ground-truth, to which all the new algorithms are compared. However, in reality, perfect algorithms with 100% accuracy are rarity. If there is a well-known algorithm with 90% accuracy as a reference and a new algorithm matches its results with 90% accuracy, how accurate really is the new algorithm, $90 * 90 = 81\%$? Or is it 100% accurate and the 10% are lost due to reference algorithm’s imperfection?

Accordingly, we argue, that it is hardly possible to prove that a particular ground-truth is objective and accurate, as it contains significant amount of subjectiveness. Still, manual ground-truth marking provides a reasonable degree of trust.

VI. Related Work

Vehicular sensing systems have been proposed by researchers previously, including BusNet [8], Pothole Patrol [9], Nericell [10], RoadMic [6]. Besides vehicular applications, other types of mobile participatory sensing dealing with environmental monitoring do exist, including BikeNet [11]. However, in this paper we do not concentrate on data analysis algorithms or communication and content delivery protocols, rather on effective and efficient data collection process and recommendations for improved real-world experiments and deployments.

This paper shares common structural and ideological patterns with prior research papers concentrating on real-world wireless sensor network experiences and deployment recommendations in different application areas, including precision agriculture [12], environmental [13] and wild animal monitoring [14], [15]. To the best of our knowledge, this is the first paper describing field test recommendations especially for vehicular sensing applications using mobile phones, having specific requirements and characteristics.

VII. Conclusion and Future Work

In this paper, we described our experience from urban vehicular sensing experiments with Android smartphones detecting potholes by analyzing accelerometer data. Although the collected data is deficient, we draw multiple highly valuable conclusions. First of all, we admit, that semi-automated ground-truth location marking by a human operator pressing a button during the test drive is subject to multiple errors due to both technical limitations and human factors. Manual pothole marking and categorization while walking along the test track is recognized as the right method for ground-truth recording, but offline position correction is recommended. We also detect differences between distinct Android devices, most significant of which is the difference in accelerometer sampling rate and deviation.

We believe, that our experience will help to improve efficiency and reduce time and effort fur further experiments using Android platform for vehicular sensing researchers. Our future work includes further evaluation of our pothole detection algorithm accuracy.

VIII. Acknowledgement

This work has been supported by European Social Fund, grants Nr. 2009/0219/ 1DP/1.1.1.2.0/APIA/VIAA/020 ”R&D Center for Smart Sensors and Networked Embedded Systems” and Nr. 2009/0138/ 1DP/1.1.2.1.2/09/IPIA/ VIAA/004 ”Support for Doctoral Studies at the University of Latvia”. Special thanks to Rinalds Ruskuls and Vadims Kurmis who provided phones for testing.

References

[1] http://potholes.co.uk, Last access: 27.05.2011.

[2] http://bedrukarte.lv, Last access: 27.05.2011.

[3] http://world.waze.com/, Last access: 27.05.2011.

[4] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, “VTrack: Accurate, Energy-Aware Road Traffic Delay Estimation Using Mobile Phones,” in *Proc. SenSys’09*, 2009, pp. 85–98.

[5] G. Strazdins, A. Mednis, G. Kanonirs, R. Zviedris, and L. Selavo, “Towards Vehicular Sensor Networks with Android Smartphones for Road Surface Monitoring,” in *Electronic Proceedings of CONET’11*, 2011.

[6] A. Mednis, G. Strazdins, M. Liepins, A. Gordjusins, and L. Selavo, “RoadMic: Road Surface Monitoring Using Vehicular Sensor Networks with Microphones,” in *Proc. of Networked Digital Technologies, NDT’10*, 2010, pp. 417–429.

[7] R. Stoleru, T. He, and J. A. Stankovic, “Walking GPS: A Practical Solution for Localization in Manually Deployed Wireless Sensor Networks,” *Proc. IEEE LCN’04*, pp. 480–489, 2004.

[8] K. De Zoysa, C. Keppitiyagama, G. Seneviratne, and W. Shihan, “A public transport system based sensor network for road surface condition monitoring,” in *Proc. NSDR’07*, 2007, pp. 9–14.

[9] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, “The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring,” in *Proc. MobiSys’08*, 2008, pp. 29–39.

[10] P. Mohan, V. N. Padmanabhan, and R. Ramjee, “Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones,” in *Proc. SenSys’08*, 2008, pp. 357–358.

[11] S. Eisenman, E. Miluzzo, N. Lane, R. Peterson, G. Ahn, and A. Campbell, “BikeNet: A mobile sensing system for cyclist experience mapping,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, pp. 6:1–6:39, December 2009.

[12] K. Langendoen, A. Baggio, and O. Visser, “Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture,” in *IPDPS’06*, 2006, pp. 1–8.

[13] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, “The Hitchhiker’s Guide to Successful Wireless Sensor Network Deployments,” in *Proc. SenSys’08*, 2008, pp. 43–56.

[14] R. Zviedris, A. Elsts, G. Strazdins, A. Mednis, and L. Selavo, “Lynxnet: Wild animal monitoring using sensor networks,” in *Proc. REALWSN’10*, 2010, pp. 170–173.

[15] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi, “Hardware design experiences in ZebraNet,” in *Proc. SenSys’04*, 2004, pp. 227–238.

Development of a Mobile Cardiac Wellness Application and Integrated Wearable Sensor Suite

Paul J Fortier
Benjamin Viall
Electrical and Computer Engineering
University of Massachusetts Dartmouth
North Dartmouth, MA., USA
pfortier@umassd.edu
bviall@umassd.edu

***Abstract*— This paper describes the design process for a mobile patient-centric, self-monitoring, symptom recognition and self intervention system supporting chronic cardiac disease management. The system design was undertaken in five phases, refining our concept from crude prototype to brass-board system ready for product development and experimental testing. Our system is composed of a mobile smart phone and wearable sensor suite linked through blue tooth and cell phone technology to a backend data repository, data mining, knowledge discovery, knowledge evolution and knowledge processing system, providing clinical data collection, procedural collection, intervention planning, medical situational assessment and health status feedback for users. The system aids patients in learning to recognize disease specific symptoms and understand the effect on their health of adherence to interventions.**

Keywords-* *Collaboration tools, Sensor Design, Cardiac Wellness

I. INTRODUCTION

Cardiovascular disease represents a significant public health problem affecting approximately 6 million Americans last year alone at a cost of 37.2 Billion dollars [1]. Heart failure, as a chronic and progressive illness, primarily of the elderly, has a negative effect on patients' quality of life with symptoms affecting well being, limiting normal daily living activities, and increasing the risk of multiple hospitalizations [2]. Faulty self-care behaviors including the inability to recognize symptoms and seek timely treatment are linked to hospitalization in this population [3]. Self-care of heart failure is difficult for many patients since it involves daily monitoring of symptoms, dietary restrictions, and correctly taking multiple medications [4].

Symptom awareness and monitoring, important for self-care, are confounded by the often insidious yet subtle changes in severity of symptoms, as well as due to advanced age and cognition, leading to discounting of early symptoms of heart failure decompensation, such as fatigue and dyspnea, to aging [5].

Recent heart failure medical management advances include the use of implanted devices to provide early cues to decompensation through monitoring of heart rate, fluid status, and heart rhythms [6]. Unfortunately, not all persons with heart failure are able to receive such devices for a variety of reasons including cost, eligibility and geographic proximity to a clinic that monitors such devices. Wearable low cost devices easily placed by a patient or clinician are needed to support more frequent access to real-time cardiac physiologic data [7].

A number of researchers are attempting to develop interventions to improve symptom recognition, interpretation and ultimately improve treatment seeking behaviors [8]. Before tailored, patient centric heart failure care models can be developed, more information is needed about the process of symptom recognition and the validity of health related feedback in improving self-care in heart failure patients. Currently, no research has shared data derived from technological monitoring processes with patients themselves with the goal of making them an active partner in the process of performing self-care. Increasing patient's ability to identify and report these correlates may increase the ability to develop interventions that improve symptom recognition and treatment seeking behaviors.

The problem with conventional out-patient post cardiac disease management is the inability to monitor and assess patient health status in real-time within the home setting. Most post hospitalization care protocols require patient visitation by a clinician to assess health and progress. These visitations occur sporadically and are not necessarily focused on patient needs. Patients in such home care settings do not typically comprehend their own disease status or changes correctly. Patients have not been trained before leaving the hospital environment on what symptoms are important and how to assess subtle, yet important changes to these symptoms.

Our research fills the need for tools that aid both the patient and clinician in improving individual patient outcomes. The primary function of our tool is to assist patients through simple interactions and feedback in learning how to recognize the relative status of their health condition, to recognize and understand symptom and status changes either positively or negatively and to develop and put into

ISBN: 978-1-61208-144-1

place self interventions to implement in between clinical health care professional encounters.

II. BACKGROUND

The theoretical framework that guides our study is the situation specific Theory of Heart Failure Self-Care, fig. 1. The theory posits that self-care is a naturalistic decision making process consisting of two types of behaviors: self-care maintenance and self-care management. In self-care maintenance, persons with heart failure follow the advice of a provider to take medications, follow a diet and make healthy lifestyle choices such as getting a flu shot [9,10]. Self-care management is an active process of heart failure symptom monitoring that begins with recognizing symptoms, evaluating the symptoms, deciding on a course of action, taking action and finally evaluating the effectiveness of the selected course of action (treatment evaluation). All of these behaviors require practice to develop skills and confidence in the ability to undertake the behaviors.

The self-care management portion of the model is patient-centered with the patient making treatment decisions based on knowledge and context. Often self-care decision making is embedded in other processes and not a discrete task thus increasing the complexity of the process.

Symptom recognition, evaluation and treatment implementation are critical yet extremely hard to realize goals in chronic heart failure patient populations [11]. Prior research suggests that a number of factors influence self-care behaviors in persons with heart failure. First, patients fail to recognize the symptoms of an acute heart failure exacerbation leading to delay in seeking treatment, resulting in costly hospitalizations [12,13]. Second, knowledge about appropriate self-care actions to maintain health in heart failure is low [14]. Finally, making linkages between acute symptoms and appropriate actions requires practice and education beyond the hospitalization period.

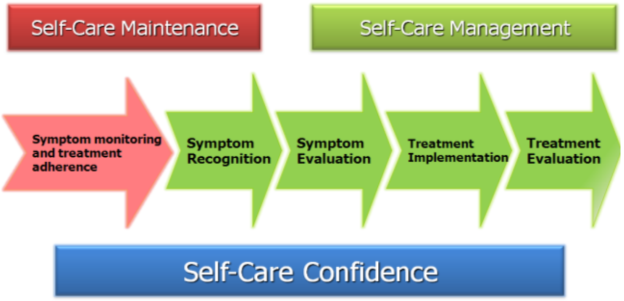


Figure 1: Theory of heart Failure Self-care [10]

In patients with cognitive changes, the ability to make linkages between symptoms and actions may require new interventions to support the development of this critical self-care behavior.

III. MOBILE CHRONIC HEART FAILURE MONITORING AND ASSESSMENT

In our mobile monitoring system, physiologic and perceptual mobile instruments are used to gather patient specific information. Gathered data include, physiologic measurements, perceptual measurements, psychosocial measurements, health history and ontological information. This information is collected and fused into our system's knowledge base using four forms of case equivalence (structural equivalence, functional equivalence, conceptual equivalence and temporal equivalence) forming new clinical use cases mined for new clinical knowledge [15]. The detection of patterns and sequences in time oriented clinical data is an important component of our analysis and takes into account subtle differences in how individual patients react to their malady and care interventions. [16].

Health monitoring, wellness and assessment application algorithms use these and a variety of other data sets to assess patient status and develop patient-centered interventions. For example, the physiological data sets are collected through the wearable sensors, during home visits, during clinical visits and during unplanned health crisis events providing a means to construct a temporal map of the patient's physiologic health viewable at any instance of time or range of time.

IV. SYSTEM DESIGN, ARCHITECTURE AND OPERATIONS

The need to develop a more effective and seamless integration of all forms of related data into health care delivery has been described for over a decade [17,18], with progress mostly in health care related business processes. Our System and user interface design included the nurse and patient from the beginning to improve on acceptance and use.

Clinical knowledge is created during interactions between the patient and the nurse, between a patient and automated system or nurse and automated system with all clinical events stored in a knowledge base, fig. 2. Patient centric medical knowledge is made available to nurses and patients to aid in outcome improvement through two collaborative applications [19,20,21].

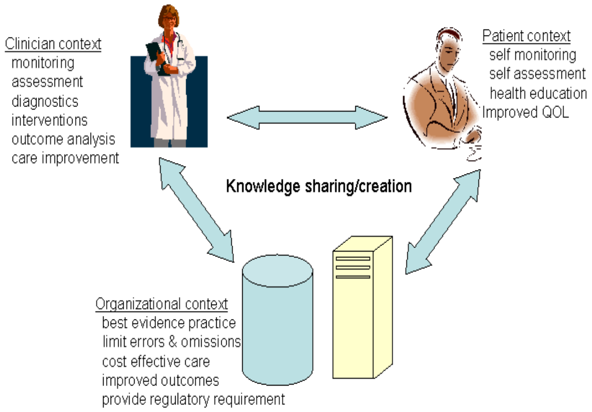


Figure 2: Clinical knowledge creation model

ISBN: 978-1-61208-144-1

Our system is intended to mimic a knowledgeable mentor, guiding users (both patient's and nurses) through clinical data collection, situational assessment and decisions using information specific to the individual patient. Through iterative use, successful health management strategies can be learned, practiced and honed. Another desired outcome is the ability to uncover new practice knowledge using data mining methods which have seen very limited use to date in non-hospital based health care settings.

V. System Operations and use

To improve both patient and clinicians ability to recognize subtle changes in symptoms requires more frequent collection, analysis and presentation of patient physiologic and perceptual data be performed. Our system uses a mobile smart phone and wearable sensor suite linked through blue tooth technology connected with a backend data repository, data mining and knowledge processing system, fig. 3, to improve data collection, real-time assessment and interventional decision support.

The front end patient and nurse mobile monitoring devices and patient wearable sensors act as the clinical data collection, procedural collection, intervention planning, medical situational assessment and instructional feedback platform. The backend inference engine is based on the integration of case-based and rule-based reasoning subsystems [22]. Clinical knowledge is represented as a set of data rules and associated meta-rules, ranked using evidence-based and usage relationships [23]. An approximate answer to a clinical problem is derived using rules and similarity computations [24,25,26] computed under four different contexts, depending on the phase of the decision process that the nurse or patients are in [15,22].

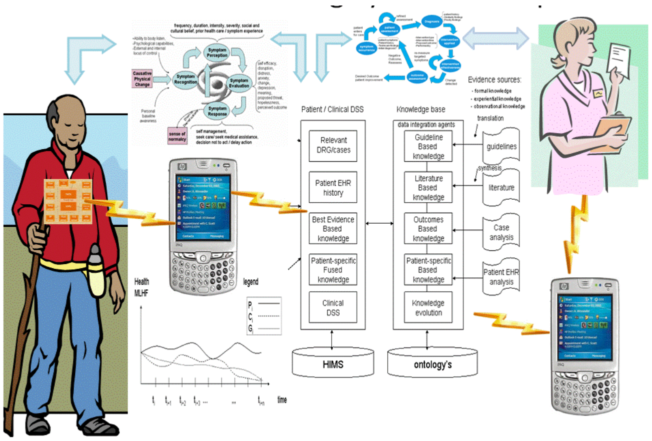

frequency, duration, intensity, severity, social and cultural belief, prior health care / symptom experience
-Ability to body listen, -Psychological capabilities, -External and internal locus of control
Causative Physical Change
Symptom Perception
Symptom Recognition
Symptom Evaluation
Symptom Response
self efficacy, disruption, distress, anxiety, change, depression, meaning, proposed threat, hopelessness, perceived outcome
Personal baseline awareness
sense of normalcy
self management, seek care/ seek medical assistance, decision not to act / delay action
Diagnosis
patient assessment
symptom occurrence
intervention applied
intervention effectiveness
outcome assessment
Desired Outcome patient improvement
Negative Outcome, Reassess
Evidence sources:
- formal knowledge
- experiential knowledge
- observational knowledge
translation
synthesis
Patient / Clinical DSS
Relevant DRG/cases
Patient EHR history
Best Evidence Based knowledge
Patient-specific Fused knowledge
Clinical DSS
Knowledge base
data integration agents
Guideline Based knowledge
Literature Based knowledge
Outcomes Based knowledge
Patient-specific Based knowledge
Knowledge evolution
guidelines
literature
Case analysis
Patient EHR analysis
HIMS
ontology's
Health MLHF
legend
P
C
G
time

Figure 3: System conceptual architecture

In order to improve the efficiency of case lookup, multi-context clusters of cases are formed using declarative, procedural and semantic context. Similarity searches focus on multiple events temporal sequences to determine how this patient's present scenario relates to past cases [24] both general and specific. For trend analysis the system examines how data items transition over time [23] and relate to past cases stored in the knowledge repository or represented in the knowledge ontology.

A nurse can invoke the patient collaboration application to perform a remote virtual home visit, to extract both spatial and temporal physiologic and perceptual measurements. A nurse, selects patients from those assigned and available on their device screen, selects and performs a variety of physical and psychological health assessments, evaluates patient status using collected and historic information and plans patient interventions focused on accomplishing desired patient outcomes (e.g. improved quality of life), fig. 4.

The clinical nursing mode application provides for active patient monitoring and extraction of stored and real-time measurements, configurable as an automated process. A second application supports nurse visual assessment of assigned patients. At the highest view, a nurse can request visual annotations (e.g. colored icons, green for all is well, yellow for some issues, orange for serious issues and red for dire issues) to indicate patient aggregated health assessment. Using visual patient representations a clinician can easily select a specific patient data set to drill down into for prior and present assessments using a variety of visual aids. For example, through trend analysis graphs, fig. 5, indicating improvement or degradation, or detailed physiologic graphs showing real-time measurements and historical measurements with expanded details available due to events such as a clinical office visit or hospitalization (e.g. blood workups, etc.).

The nurse can further choose intervention analysis and planning tools to examine which form of intervention is best suited for this specific patient [22]. The clinician can also reconfigure the patient wearable sensor suite or cell phone applications to change the periodicity of measurements as well as sensor sensitivity or even add new applications as they become available.

A second collaborative application supports patient self reporting and self analysis of health and wellness status. Patients select applications supporting lifestyle, physiologic and psychological assessments. Applications implement questionnaire based instruments such as the Duke Activity Status Index (DASI) or Heart Failure Somatic Awareness Scale index (HFSA). Once self evaluations are completed, patients can transmit data to the backend database or store locally. Self reporting and patient symptom self recognition is based on the symptom self recognition model [10], fig. 1. This model requires patients to be actively engaged in their health management. Using the tool set, a patient can further use visual assessment tools to examine progress of their self health management plan. Such feedback supports patient learning.

A third self management and intervention mobile application, allows a patient to explore a set of self interventions to (e.g. exercise regime, diet alteration, etc.)

ISBN: 978-1-61208-144-1

aid in symptom and overall health management. Patients can examine what the impact on their health could potentially be if they adhere to a specific intervention. Collaborative patient self management has been shown [27, 28, 29] to be an effective tool in helping patients understand, respond to and manage their health condition making them an active participant in their self care.

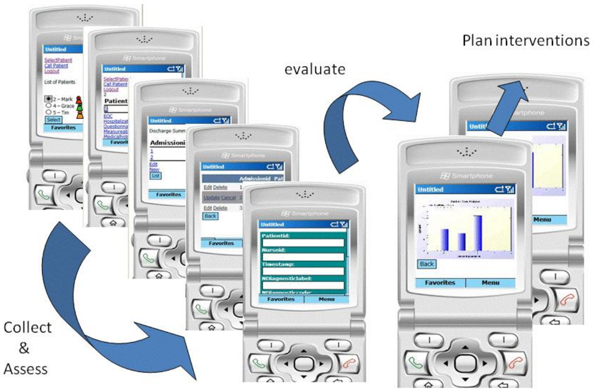


Figure 4: Clinician Example usage sequence

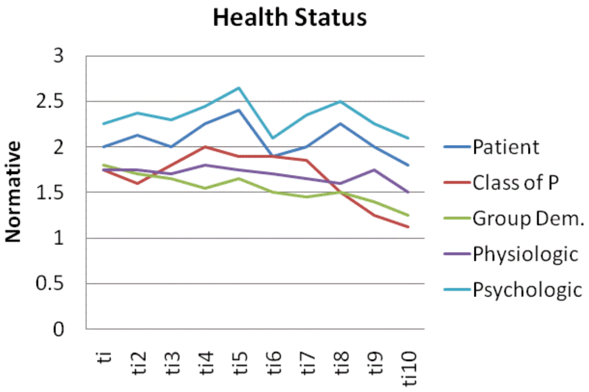


Figure 5: Trend analysis graph

VI. SYSTEM DESIGN

Phase one of five phases within our design odyssey (fig.6) involved the design of all sensors from scratch so that each could be designed with the end system on a chip (SOC) goal in mind. During this phase of design (completed fall 2010) all components were designed and tested using a microprocessor as the core computing engine, along with discrete printed circuit board (PCB) designs for each of the seven sensors.

The electro cardiogram (ECG) Sensor, fig. 7 (top left), was designed using a three lead concept. The sensor operates by measuring the electrical activity called action potential generated by the heart muscles during heart contractions, representing a graphical mapping of measurements taken of the heart's electrical activity over time. The signals generated are measured using electrodes that convert the electrical potential into a measureable voltage signal. Our sensor operates by extracting measurements using a two lead approach with the third being body ground. The two electrodes rest on the right and left chest. The measured voltage is in the 1 to 5 mV range and is measured from the AgCL electrodes. The signal is then amplified using a high gain (e.g. 100) amplification factor.

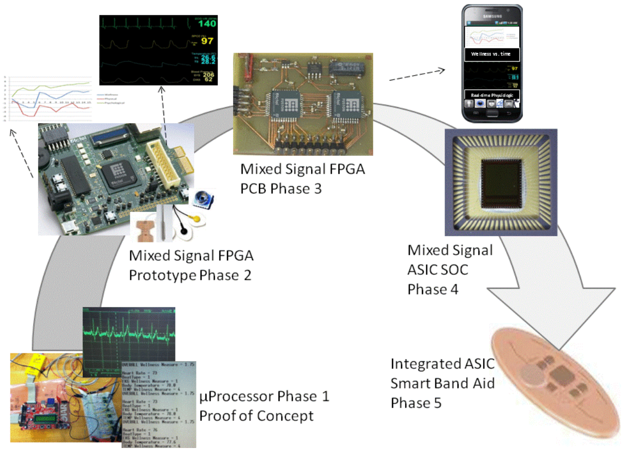


Figure 6: System Design Phases

The amplified signal is then isolated and filtered using a band pass filter (e.g. 0.04 – 150 HZ). The transducer output is then sent to a digital signal processor that extracts out the ECG signal, determines the principally important points (PIP) and then sends these to the core processing engine for ECG classification. Our sensor classifies ECG signals into one of six classes; normal, slow-fast heart rate, irregular heart rhythm, abnormality in ECG P wave, abnormality in ECG QRS wave, or abnormality within the ECG QRS wave and the ECG P waves combined.

The Oxygen Saturation sensor (SO2) sensor was designed using tradition pulse oximetry concepts. Two light emitting diodes are used that produce red and infrared beams at 660 nm and 940 nm respectively. These signals are pulsed at approximately 30 times per second with a synchronous on cycle, followed by a pause cycle allowing for compensation due to ambient light. A photo-detector is used to measure the received light using the transmission method. The received red and infrared signals are extracted and used to compute a ratio of red to infrared light measured. This ratio is then used to compute the SO2 percentage. In general typical ratios of .5 equate to an SO2 of 100%, a ratio of 1 to an SO2 rate of 82% and a ratio of 2 to an SO2 of 0%. Our computations and calibrations use models well known in the industry. The SO2 sensor front end design is shown in fig. 7 (bottom left).

The present sensors were implemented on a PCB and subsequently re-implemented in a mixed signal FPGA core. We are presently in the process of taking the core designs and redesigning for a mixed signal system on a chip (SOC) which will then be fabricated into our final form factor (e.g. the smart band aid, fig. 6, phase 5).

The third sensor is a simple temperature sensor. The design of the temperature sensor was accomplished using a 2 wire thin film resistive temperature detector (RTD) and a simple translation component that interprets the detected voltages across the RTD wires and converts these into a temperature using a simple model. Using these basic sensors, all other cardiac measurements (e.g. blood pressure, blood flow, pulse rate, respiration rate) are derived using synthetic sensor concepts built using synthetic models [30].

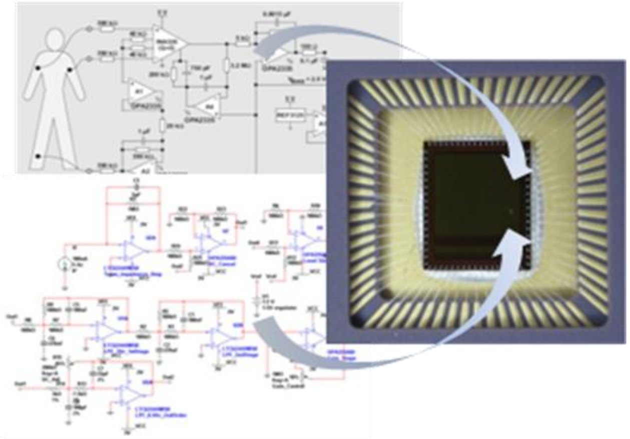


Figure 7: ECG/O2 Sensors Design and Integration

For example, to calculate the pulse rate the ECG wave interval is tracked and counted. The count of waveforms per minute is then used to compute a pulse rate averaged over time to reflect the clients pulse. Likewise for respiration rate, the ECG can be used to extract out a background variation signal in the amplitude of the ECG peaks in synch with the clients respiration rate.

Likewise the client's blood pressure and blood flow can also be calculated from the ECG and SO2 values generated from the sensors using relationships exhibited in the signals along with functions derived from the client's physiology or using an additional acoustic transducer that measures flow variations. Our final design will choose one based on a least number of transducers principle.

The block diagram for our sensor is shown in fig.8. There are many novel features being developed as part of our system. First, we have adopted a cost cutting approach in terms of dollars, size, weight and power driven by the use of a minimum number of physical transducers to deliver the required measurements effectively and non-intrusively [31]. Secondly, we have gone for a single system on a chip design that will minimize the number of electronic elements that will be integrated into the final form factor (the smart band aid). Third, we have integrated the sensor processing with a cell phone mobile application to minimize the need for extensive on chip storage. Data storage needs will be limited to a maximum of three days of sensor readings based on present calculations.

The SOC sensor suite also consists of position and motion sensors integrated on chip allowing us to compensate for motion artifacts, reducing false alarms and producing more accurate readings based upon a patient's true movements. The entire device is linked to a cell phone via a blue tooth transmit and receive component integrated on the chip. The only external component needed is the antenna. Three internal subsections are used to compute ECG, SO2 and temperature readings from the raw transducer measurements. These measurements are then used in the synthetic sensor derivation unit (fig.8) to synthetically derive all other measurements. All physical and derived cardiac measurements are then used by the cardiac wellness engine (along with mobile phone based psychological assessments) to compute a patients relative cardiac health and cardiac health events (to signal adverse conditions automatically).

The board level version of the system, fig. 6 phase 3, will be completed by August 2011. This system was designed by a team of graduate computer engineering students, led by the authors who will complete all integration, testing and fielding of the system in an experimental study to begin in January of 2012. The final product, shown in fig.6 phase 5, is projected to be available after the completion of the experimental study which runs for an 18 month period.

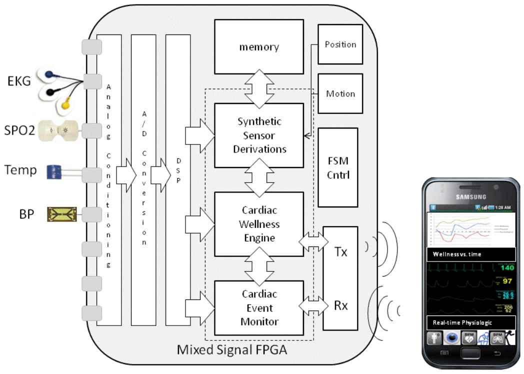


Figure 8: Wearable Sensor Architecture and Interface

VII. SUMMARY

We presented a novel mobile collaborative decision support tool consisting of an integrated non-intrusive sensor suite, mobile smart phone application and backend server. The system is designed for patient and clinical nurses and utilizes specific patient physiologic and psychosocial information and evidence-based nursing knowledge to offer real-time guidance to the patient and clinician mimicking that of an expert mentor. The system's architecture is presented from three different viewpoints; an informational view, an operational perspective and a architecturall design view. In the informational description, we utilized multiple sources of information to construct patient specific health assessments and wellness measures based on real-time and historic patient-centric data. In the operational system description, particular emphasis was placed on describing the steps patients and clinicians utilize in performing data collection actions, patient assessments, patient evaluations and intervention planning and execution.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under grant #EIA0218909 and NUWCDIVNPT under N00178-04-D-4109-N410.

REFERENCES

[1] American Heart Association. Heart and Stroke Statistical Update 2009. Retrieved on 3/12/2010, http://amhrt.org/statistics/index.html.

[2] American Heat Association, "Heart failure hospitalization rates rise among nation's seniors", American Heart Association heart disease and stroke statistics: 2008 update. http://amhrt.org/statistics/index.htmlRetrieved on 12/12/2008.

[3] C.Annema, M.Luttik, and T. Jaarsma,. Reasons for readmission in heart failure: Perspectives of patients, caregivers, cardiologists and heart failure nurses. Heart & Lung, 38(5),2009, pp. 427-434.

[4] K.Grady, K. Dracup, G. Kennedy, D. Moser, M. Piano, L. Stevenson, and J. Young, "Team management of patients with heart failure: a statement for healthcare professionals", Cardiovascular Nursing Council of the American Heart Association. Circulation 2000; 102, pp. 2443-2456.

[5] E. Stoller, L. Forster, R. Pollow, and W. Tisdale, "Lay evaluation of symptoms by older people: An assessment of potential risk". Health Education Quarterly 1993; 20, pp.505-522.

[6] E. Popojes and J. Bochmer, "On device monitoring and predicting heart failure exacerbation". Current Treatment options in Cardiovasc Medicine 2008; 10: pp. 371-379.

[7] J. McManus, J. Salinas, and J. Holcomb, "The Ability to Capture "Real-Time" Physiologic Patient Data Using the Trauma Vitals System During Air Transport National Association of EMS Physicians, Registry Resort, Naples, FL from http://www.allacademic.com/meta/p61904_index.html, accessed 7/19/2011.

[8] C. Jurgens, "Soamatic awareness, uncertainty, and delay in care seeking in acute heart failure". Research in Nursing & Health, 2006; 29: pp 74-86.

[9] B. Carlson, B. Riegel, and D.Moser, "Self care abilities of patients with heart failure". Heart and Lung, 30(5), 2001, pp. 351-359.

[10] B. Riegel,and V. Dickson,. A situation-specific theory of heart failure self-care. Journal of Cardiovascular Nursing, 23, 2008, pp. 190-196.

[11] C. Lee, N. Tkacs, and B. Riegel, The influence of heart failure self care on health outcomes: Hypothetical cardio-protective mechanisms. Journal of Cardiovascular Nursing, 24(3), 2009, pp. 179-187.

[12] L. Evangelista, K. Dracup, and L. Doering, "Treatment-seeking delays in heart failure patients". The Journal of Heart and Lung Transplantation, vol 19, 2000, pp. 932-938.

[13] C. Jurgens, D. Moser, R. Armola, B. Carlson, K. Sethares, and B. Riegel, Symptom Clusters of Heart Failure, Research in Nursing and Health, 2009, Vol. 32, No. 5, pp. 551-560.

[14] N. Artinian, M. Magnan, M. Sloan, and M. Lange, "Self-care behaviors among patients with heart failure", Heart & Lung, 2002, vol 31, pp. 161-172.

[15] T. Mitsa, P. Fortier, G. Yang, A. Shrestha, N. Dluhy, and E.O'Neill, "A Dynamic, Client-Centric Point-of-care System for the Novice Nurse", Communications of the Association for Information Systems Journal, Information Systems and Healthcare XXI: Volume 19 Article 36, June, 2007. Available at http://aisel.aisnet.org/cais/vol19/iss1/36

[16] M. Prcela, D. Gamberger, and A. Jovic, Semantic web ontology utilization for heart failure expert system design, Medical Informatics Europe, pp. 851-856, 2008.

[17] J. Sensemeier, "Advancing the state of data integration in healthcare". Journal of Healthcare Information Management, 17(4), pp. 58-61, 2003.

[18] P. Brown, and P. Sonken, "Evaluation of the Quality of Information Retrieval of Clinical Findings from a Computerized Patient Database Using a Semantic Terminological Model", JAMIA, vol. 7/4, 2000, pp. 392-403.

[19] A. Chapyala, Data Mining in Home Health Care Management, University of Massachusetts Dartmouth, ECE Technical Report UMD-ECE-TR-2009-101, 1/23, 2009.

[20] R. Antony, Development of Hit Ratio Algorithm and Matching Priority Index Supporting Patient Case Indexing, University of Massachusetts Dartmouth, ECE Technical Report UMD-ECE-TR-2009-102, 3/6, 2009.

[21] A. Sishtla, Similarity Search Using Frequent Case and Rule Based Indexing Techniques, University of Massachusetts Dartmouth, ECE Technical Report UMD-ECE-TR-2008-103, 3/28, 2008.

[22] P. Fortier, "Improving student nurses clinical care experience through the use of a computerized Mobile hand-held Decision Support System", IADIS mobile learning Conference, April, 2008, pp. 10.

[23] D. Wang, "Mining T-outliers in High Dimensional Time Series Financial Data". Ph.D. Dissertation, ECE department, University of Massachusetts Dartmouth. 2005.

[24] B. Joshi., Assessing the Significance of MLHFQ Attributes from a Spatial/temporal Perspective using Data Mining Techniques, University of Massachusetts Dartmouth, ECE Technical Report UMD-ECE-TR-2008-104, 12/16, 2008.

[25] S. Chandran, Improving Chronic Heart Failure Nurse Care Through Mobile Monitoring of Subtle Behavioral / Physiological Changes, University of Massachusetts Dartmouth, ECE Technical Report UMD-ECE-TR-2008-105, 12/16, 2008.

[26] P. Narra, Improving Chronic Heart Failure Patient Self Care Through Mobile Monitoring of Subtle Behavioral / Physiological Changes, University of Massachusetts Dartmouth, ECE Technical Report UMD-ECE-TR-2008-106, 12/ 16, 2008.

[27] A. Jovicic A. Holroyd-Leduc, and S. Straus, Effects of self management intervention on health outcomes of patients with heart failure: a systematic review of randomized controlled trials, BMC Cardiovascular Disorders, Volume 6 Number 43, 2006, 8 pp.

[28] T. Koerlling M. Johnson, R. Cody, and K. Aaronson, "Discharge education improves clinical outcomes in patients with chronic heart failure", Circulation, Volume 111, Number 2, 2000, pp 179-185.

[29] T. Rector S. Kubo, and J. Cohn, "Patients' self assessment of their congestive heart failure, Part 2: Content, reliability and validity of a new measure, The Minnesota Living with Heart Failure questionnaire", Heart Failure, Volume 3, 1987, pp 198-209.

[30] P. Fortier and K. Dasari, Examining the Role of Metadata in Testing IED Detection Systems, International Test and Evaluation Asssociation Journal - Sept 2009; Volume 30-3 pp. 421-433.

[31] P. Fortier, O.Aljaroudi, P.Boddu, P.Chaiyasucheeva, R. Kanjee, A.Tannous, and W.Turner, "Design of a Flexible Cardiac Health Monitor - Supporting Patient/Clinician Subtle Cardiac Symptom Recognition", Poster, IEEE Sensors 2011.

Design of a Low-Cost 'Constant Phase Angle' Based Sensing System to Detect Natural Milk and 'Synthetic-Milk' Reconstructed from 'Liquid-Whey'

Siuli Das, M. Sivaramakrishna, Bhaswati Goswami
Department of Instrumentation & Electronics Engineering,
Jadavpur University, Kolkata-700098, India
e-mail: bg@iee.jusl.ac.in

Karabi Biswas
Department of Electrical Engineering,
I.I.T Kharagpur, Kharagpur-721302, India
e-mail: karabi@ee.iitkgp.ernet.in

***Abstract*—In this work, a low cost automatic sensing system is proposed to detect 'synthetic milk', which has been reconstructed after adulterating the milk with 'liquid-whey'. The constant-phase angle (CPA) based sensor detects the synthetic milk and a micro-controller based circuit drives a stepper motor to close the valve installed at the outlet of the milk supply line to prevent mixing. The sensor is stick type, rigid and hence it is easy to mount. The electrodes of the sensor are coated with polymethyl methacrylate (PMMA) film, which makes it bio-compatible and suitable for the application.**

Keywords-constant-phase-angle; milk adulteration; liquid-whey; automatic sensing system; phase detector circuit.

I. INTRODUCTION

Milk adulteration is a century old problem [1, 2]. The oldest and simplest method of adulterating milk is by dilution with water to increase volume [3] and then to compensate specific gravity, different types of salt or sugar [4] are used. Sometimes the color change due to adulterants are corrected by the addition of a small amount of coloring matter [5] which may cause serious health problem [6].

A similar type of milk adulteration problem is reported in this work, where the volume is increased by addition of 'liquid-whey' (liquid by-product of cottage cheese) to increase the volume. The liquid- whey addition makes the natural milk a little acidic and lowers its pH value which is compensated by adding NaOH.

It is a well known fact that NaOH may cause health hazards to the patients suffering from heart disease and hypertension. It also deprives the body from utilizing lysine which is required for growing babies. Moreover, some greedy cheese maker uses cheap muriatic acid (a chemical composition of hydrochloric acid) to make cheese from milk, which causes more health problems.

The profit of using liquid-whey, as milk adulterants is double folded. First of all it is in abundance to the cheese maker, so cheap and easily available to the milk supplier. Secondly whey retains many natural properties of milk, so preparation of synthetic milk from liquid-whey is simple and can camouflage the natural milk easily. As a result adulteration of natural milk with liquid-whey became a wide spread fraudulent activity, where huge amount of cottage cheese is used everyday to make different varieties of sweets. Hence, this becomes a serious concern to the dairy firms who buy milk from thousands of different milk suppliers and need a simple, robust and bio-compatible automated sensing system to their milk incoming line for quality control.

In this work a stick type probe, whose electrodes are coated with porous film of PMMA [7], is used as a sensor for developing the sensing system. The advantage is that it can be easily dipped inside the medium and at the same time biocompatible due to the PMMA coating on the electrodes. More over, the detection is based on the principle of change of phase angle with the ionic property of the medium. The phase angle remains constant over a band of frequency, so measurement can be frequency independent, which is an important requirement for automated sensing system. The other advantages are - it is simple, easy to fabricate, cheap and can be replaced easily.

The paper is organized as follows: Section II describes the principle of operation of constant phase angle based sensor to detect milk adulteration. Section III provides fabrication and characterization of the CPA-based sensor. Section IV details the integrated sensing system with the detector circuit. Section V discusses the experimental observations and results. Section VI presents a conclusion.

II. PRINCIPLES OF CONSTANT PHSAE ANGLE –BASED MEASUREMENT

In earlier works, the construction and working principle of the constant phase angle based sensor has been reported [8]. The impedance of the sensor (Fig. 2) dipped inside the medium can be represented as

$$Z(s) = Qs^{-\alpha} \tag{1}$$

where Z is the impedance, Q is a constant, α is a real number and s is the Laplace operator. So, magnitude $|Z| = Q\omega^{-\alpha}$ and phase angle $\theta = -\alpha\pi/2$, where θ is expressed in radians and independent of frequency. When $\alpha = 1$, $Z = Qs^{-1}$ and Z represents a capacitor. Similarly for $\alpha = 0$ and -1, Z represents a resistance and inductance respectively.

For the above sensing system, it has been observed experimentally that the constant phase angle θ is a function of three parameters of the measurement arrangement and expressed as

$$\theta = f(A, t, \sigma) \tag{2}$$

where 'σ' is the property (e.g., ionic, dielectric) of the medium under test, 'A' is the area of contact of the probe with the test medium and 't' is the thickness of PMMA-film coated on the electrodes [7, 9].

In this work, the change in constant phase angle θ for pure milk, adulterated milk and synthetic milk (reconstructed after adding liquid-whey) has been observed to identify synthetic milk. For a particular measurement, area of contact of the probe with liquid medium and the thickness of film coated on the electrodes are kept constant. However the effect of these two parameters needs to be studied for standardizing the sensor and the optimum area and thickness is to be chosen for measurement [10]. It has also been observed that this constant phase angle changes with the change of physical property of the polarizing medium (e.g., ionic concentration). This property of the CPE can be used for sensing purpose- this means the phase angle of the CPE will be different for plain milk and the milk with some impurity.

It has been already mentioned that the phase angle remains constant over frequency which makes the measurement independent of the frequency. However the bandwidth is limited to 10 kHz to 20 kHz. In this report all the measurement is performed at 15 kHz, so that in automatic sensing, if due to environmental or other effect the frequency of the detector circuit shifts, it will not effect the measurement.

III. Fabrication and Characterization of the sensor

The principle of development of thin PMMA-film coating on the probe electrodes are similar as reported by the authors in earlier report [11]. The sensor used in present application is 6 cm long, 6 mm wide, cut from double sided copper cladded PCB, generally used in electrical circuit fabrication. A thin film of PMMA is coated on the electrode surface using spin coating technique. The different film thickness can be achieved by changing concentration of the PMMA chloroform solution and the speed of the rotation of the spin coating machine. In the present study sensor with coating thickness of 18 μm is used.

The fabricated sensor is characterized by measuring impedance (Z) and phase angle θ with a LCR meter (Agilent Precision Impedance Analyser 4294A). The sinusoidal signal frequency is varied from 100 Hz to 4 MHz and peak to peak voltage is 1 V. The phase angle change is noted with standard buffer pH 4.0, pH 9.2 solutions, and also in pure milk. Each measurement is repeated five times and the average value of phase angle is plotted. It can be observed in Fig. 1, that the sensor gives a constant phase angle in the frequency range 10 kHz-20 kHz. The entire experiment was performed in controlled room temperature of 20^{o}C, since temperature is one of the dependent parameters of pH value.

IV. Design of the sensing system

Fig. 2a shows the proposed sensing system to detect the synthetic milk. The physical dimension of the probe is shown in Fig. 2b. The first block consists of the CPA-based sensor dipped inside the test sample. The output of the sensor is a phase angle (which remains constant over a frequency band). A phase detector circuit [12] measures the phase angle and gives voltage output. The display will glow indicator circuit whenever adulteration is detected. The microcontroller based stepper motor drives the control valve so that whenever adulteration is detected the control valve at the outlet of the milk line shuts off to prevent mixing.

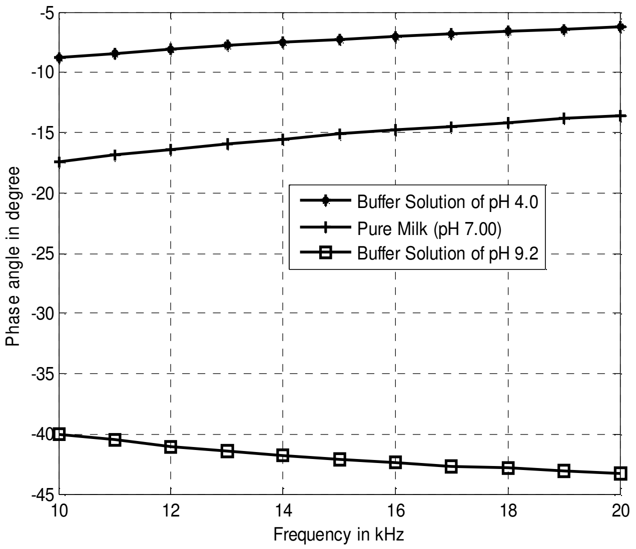


Fig. 1: Constant Phase behaviour of the sensor in three standard test medium.

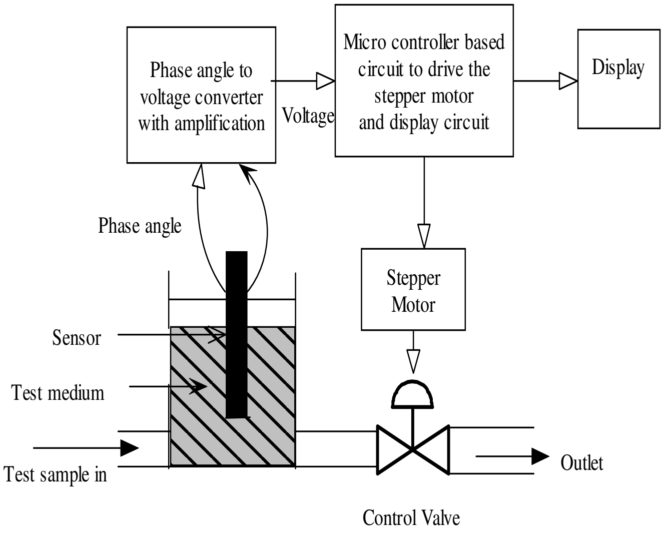

Phase angle to voltage converter with amplification
Voltage
Micro controller based circuit to drive the stepper motor and display circuit
Display
Phase angle
Stepper Motor
Sensor
Test medium
Test sample in
Outlet
Control Valve

Fig. 2a: Block diagram of the proposed sensing system.

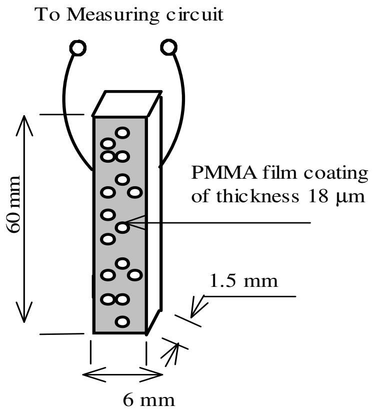


Fig. 2b: Schematic diagram of the probe
(Dimensions are not in scale)

A. Sensor block

The sensor block consists of the CPA sensor dipped inside the test medium. The interaction of the sensor with the test medium changes its impedance. In this measurement the “change of phase angle” in different test medium is considered as the sensor output. In Eqn. 2, it has been mentioned that the change of phase angle is dependent upon the ionic property of the test medium, hence the milk sample consisting of different ions will result different values of the phase angle. The phase angle mode is advantageous as the subsequent signal transduction becomes easy [8]. More over, for a particular measurement the phase angle remains constant over a frequency band, which is an essential requirement for automated sensing system [13, 14] where change of signal frequency for interference or other effect will not hamper the measurement.

B. Phase angle detector

The phase detector circuit [11] is shown in the Fig. 3. In this circuit four op-amps and one XOR gate are used. XOR gate mainly perform the phase detection. First two Op-Amps (Amplifier 1, Amplifier 2) act as amplifier circuit and second two (Comparator 1, Comparator 2) are comparators. The lower part of the circuit i.e., Amplifier 2 and Comparator 2 are used for reference signal. In the inverting terminal of Amplifier 2, a resistor of 1 kΩ is connected whereas in the Amplifier 1, an element with different phase angle is connected. So the outputs from the Amplifier 1 and 2 have some phase difference when a sinusoidal excitation is applied. Then the output signals are fed to the comparator. The comparator converts the sinusoidal signal to square wave.

From the truth table of the XOR gate it can be seen that when *d*, the phase difference between the two inputs of the XOR gate is zero the output voltage v_0 = 0, and when the phase difference is 90^0 the output will be maximum (half of the V_{DD} applied to the XOR gate chip). The output of the XOR gate is fed to a low pass filter containing a resistance and capacitance. So, it gives dc output voltage proportional to the phase difference between the two inputs.

Fig. 3 shows the phase detector circuit used for the sensing system. Though the circuit is a standard one, but as the constant phase angle is to be measured over a frequency range, the phase characteristics for individual component is to be studied. To do that, frequency response of the individual op-amp has been studied before fabricating the complete circuit.

C. Actuator

The micro-controller based stepper motor driver closes the control valve to perform the actuation. The output voltage corresponding to the pure milk is stored in the micro-controller and whenever a deviation occurs the driver circuit of the stepper motor is activated. This closes the control valve at the outlet and cuts off the milk supply line.

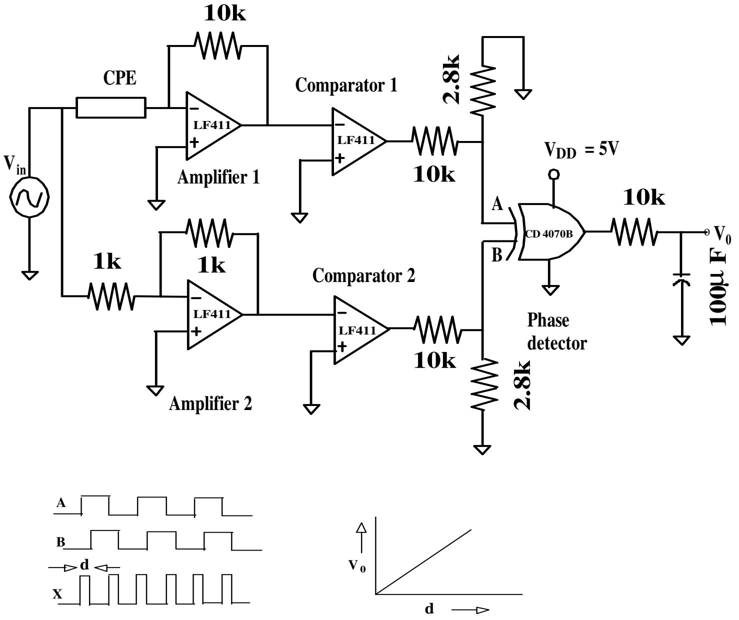


Fig. 3: Phase detector circuit

D. Display unit

The display unit consists of LCD display and a LED array system. The output voltage can be read from the LCD and a comparator based LED driver system is used to glow the ‘RED’ or ‘GREEN’ LED to indicate the status of the milk.

V. Experiments, Results and Discussion

A. Sample preparation

250 ml of pure-milk with pH value 7.00 (complete specification is given in Appendix-I) is taken in a 500 ml beaker and boiled for 15 minutes. Then 1.5 ml of “muriatic acid (a chemical composition of HCl) is added to it. In few minutes light green color liquid (120 ml) is formed in beaker while white colored solid part gets separated. This light green colored liquid is filtered by filter paper into the conical flask and this liquid is called as ‘liquid-whey’. The white solid part is termed as cottage cheese. The chemical composition of these cottage cheese and whey is provided below.

TABLE 1
COMPOSITION OF COTTAGE CHEESE AND WHEY

	Cottage cheese	Whey
Fat (%)	25-35	0.30
Protein (%)	15-25	0.70
Lactose (%)	2.0-2.5	4.40
Minerals (%)	0.3-0.4	0.60

ISBN: 978-1-61208-144-1

The above mentioned 'liquid-whey '(pH value 4.33-4.98) is added to the pure milk which makes the mixture acidic (lowers the pH value). To compensate that small amount of NaOH is added till the pH value matches the original pH value of the pure milk. This reconstructed 'synthetic milk' is the test sample.

B. *Experiment*

The experiments are carried out as discussed in Section III. The sensor is dipped inside the tests samples (first column of TABLE-2) and then the value of pH is measured by precision pH meter (Systronics Digital pH Meter 335). The magnitude and phase angle values of impedance of the sensor are measured by using precision LCR meter (Agilent Precision Impedance Analyzer 4294A) varying input frequency. And then output voltage is noted from the LCD by varying the frequency of the input signal. The results are tabulated in TABLE-2.

C. *Results and Discussion*

From Fig. 1, it is apparent that the sensor shows almost constant phase angle in the frequency zone 10 kHz–20 kHz and can be used for the proposed constant phase angle based sensing system.

The second and third columns of TABLE-2 show the pH value of the test samples and corresponding phase angle obtained by the sensor at 15 kHz. Point to be noted the phase angle is almost constant in the frequency range (10 kHz-20 kHz) which changes with the different ionic property of the test medium.

From Fig. 4, it can be observed that the output voltage of the sensing system also remains almost constant in the prescribed frequency zone (15 kHz-20 kHz). Though output voltage versus phase angle (Fig. 5) does not show a linear curve but exhibits a definite relationship, i.e., with the increase of phase angle (i.e., decrease of ionic concentration), output voltage decreases.

TABLE 2
PERFORMANCE OF THE SENSING SYSTEM IN THE TEST MEDIUM AT 15 KHZ FREQUENCY

Test samples	pH value	Phase angle	Voltage (V)
pH4.0	4.00	-7.24	2.4
Pure milk	7.00	-15.15	3.75
Pure milk+15% whey adulteration	6.95	-13.02	3.55
Pure milk+30% whey adulteration	6.87	-11.81	3.41
Synthetic milk (reconstructed after adding 30% whey and NaOH)	7.00	-12.81	3.48
pH9.2	9.20	-42.11	6.2

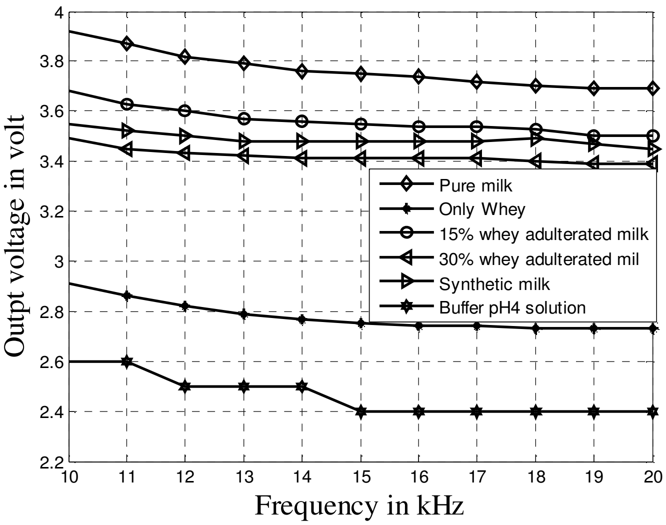


Fig. 4: The output of the sensing system for different test samples.

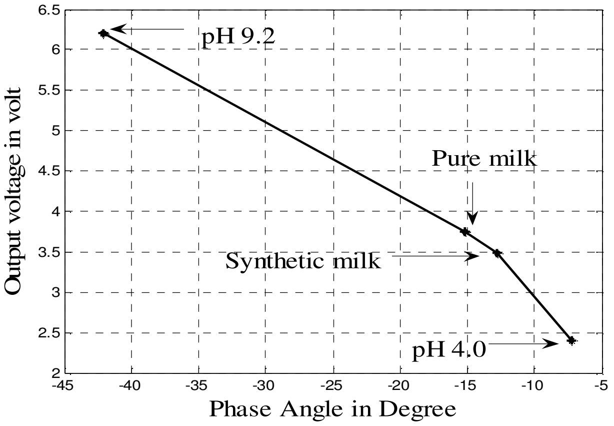


Fig. 5: The output voltage of the sensing system versus the phase angle of the sensor.

It will be worth mentioning here, that after adding NaOH to the whey adulterated milk though the pH value is brought to the same value of the pure-milk, the sensor could identify between these two test samples and shows different phase angles. This is also reflected in the voltage output (3.75 volt for pure milk and 3.48 volt for Synthetic milk).

VI. CONCLUSION AND FUTURE WORKS

In this work, a low cost automatic sensing system is designed to detect synthetic milk. The synthetic milk is reconstructed after adulterating pure milk with 'liquid-whey'. The main difference between the pure milk and synthetic milk is the presence of different ions which the sensor is capable to detect. Moreover, the measurement is based on the change of the value of the phase angle due to the presence of ions with different concentration in the test medium. The phase angle is again constant over a frequency range, which is an added advantage for such automatic sensing systems. The instrumentation system is such designed, that whenever adulteration is detected the micro-controller based stepper motor shuts off the control valve in the milk supply line and prevents mixing pure milk with adulterated milk.

The stick type PMMA-film coated probe is rigid and hence, easy to install in the system and also replace whenever necessary. The PMMA coating makes it biocompatible, an essential requirement for the system.

The proposed system can detect synthetic milk when 30% whey is added, but need to increase the resolution of the system. Further research work is on in this direction.

Modification of the electronic transduction circuit to improve sensitivity and linearity remain as one of the future scopes of work. Field testing of the sensing system is to be carried out and the life time of the sensor as well as other components of the system is to be investigated in future.

ACKNOWLEDGMENT

The authors are grateful to UGC-DAE CONSORTIUM for SCIENTIFIC RESEARCH, Kolkata, India for their help and co-operation while carrying out the experiments. The authors are also grateful to the Department of Science and Technology, West Bengal, India for their financial support. The authors also like to thank Prof. S. K. Gangopadhaya of Mohanpur Dairy Institute, WB, for his help and suggestions.

REFERENCES

[1] James Galloway, "Great Fare of London", The Lancet, 335, 2000, pp. 323-324.

[2] Harvey Gem, "The adulteration of milk", The Lancet, 69, 1857, pp. 156.

[3] John C. Baker, and Lucius L. Van Slyke, "A method for the preliminary detection of abnormal milk based on the hydrogen ion concentration", The Journal of Biological Chemistry, 1919, pp. 357-371.

[4] M. J. Reis Lima, Sílvia M. V. Fernandes, and António O. S. S. Rangel, "Sequential injection titration of chloride in milk with potentiometric detection", Food Control, 15, 2004, pp. 609-613.

[5] Hsi-Ya Huang, Ying-Chieh Shih, and Yun-Chieh Chen, "Determining eight colorants in milk beverages by capillary electrophoresis", Journal of Chromatography A, 959, 2002, pp. 317-325.

[6] Joseph G. Hattersley, "The Negative Health Effects of Chlorine", The Journal of Orthomolecular Medicine, 15, 2000, pp. 89-95.

[7] Karabi Biswas, Studies on Design, Development and Performance Analysis of Capacitive Type Sensors, Ph. D Thesis, Electrical Engineering Department, IIT Kharagpur, 2006.

[8] Karabi Biswas, Siddartha Sen, and Pranab Kumar Dutta, "A Constant Phase Element sensor for monitoring microbial growth", Sensors and Actuators B: Chemical 119, 2006, pp. 186-191.

[9] Siuli Das, Mulinti Sivaramakrishna, Manideepa Dey, Bhaswati Goswami, and Karabi Biswas, Performance of a Constant Phase Element (CPE) sensor to detect adulteration in cow-milk with whey, 8th International IEEE SENSORS 2009 Conference, 25-28 October, Christchurch, New Zealand, 2009, pp. 745-750.

[10] Siuli Das, Mulinti Sivaramakrishna, Karabi Biswas, Bhaswati Goswami, Performance study of a 'constant phase angle based' impedance sensor to detect milk adulteration ", Sensors and Actuators A: Physical, vol. 167, June 2011, pp. 273-278.

[11] K. Biswas, S. Sen, and P. K. Dutta. "Realization of a constant phase element and its performance study in a differentiator circuit," IEEE Transactions on Circuits and Systems-II, Express Briefs, vol. 53, September 2006, pp. 802-806.

[12] Darold Wobschall, Circuit Design for Electronic Instrumentation, McGraw-Hill, USA, 1979.

[13] Keat Ghee Ong, J. Samuel Bitler, Craig A. Grimes, Libby G. Puckett, and Leonidas G. Bachas, "Remote Query Resonant-Circuit Sensors for Monitoring of Bacteria Growth: Application to Food Quality Control", Sensors, 2, 2002, pp. 219-232.

[14] P Cady, "Progress in impedance measurement in microbiology", In: Sharpe AN, Clark, DS, eds. Mechanizing Microbioiogy**.** Springfield, IL, USA: Charles C. Thomas, 1978.

APPENDIX-I

Specification of the milk used for experimentation
Mother Dairy Cow Milk

Vitamin A Enriched
Homogenised & Pasteurised
Net content 500ml
Enriched with Vit A 2000 IU per 1000ml
Milk Fat 3.5% Minimum
Milk SNF 8.5% Minimum

---

<u>Nutritional information per 100 ml.</u>

Protein(g)	: 3.20	Fatty Acids(g)	: 0.008
Carbohydrate	: 4.5	Mineral(g)	: 0.70
Added vit(IU)	: 200	Vitamin(mg)	: 4.9
Energy value (K cal)	: 62	Fat(g)	: 3.5
Cholesterol(g)	: 0.01		

---

Marketed by : Mother Dairy Calcutta
P.O. Dankuni Coal Complex,
Dist. Hooghly(WB), Pin- 712310
Customer Care No. 033-2659-2342

MMPO Regn. No. 187/R-MMPO/95,

A West Bengal Govt. Project

Electrical Equivalent Modeling of Single Component Fractional Order Element with Porous Surface

Debasmita Mondal
Department of Electrical Engineering
Indian Institute of Technology Kharagpur
Kharagpur, India
debasmita.ee@iitkgp.ac.in

Karabi Biswas
Department of Electrical Engineering
Indian Institute of Technology Kharagpur
Kharagpur, India
karabi@ee.iitkgp.ernet.in

***Abstract*—A single component fractional order element (FOE) is indigenously developed and its electrical equivalent model is proposed. The FOE exhibits fractional order behavior due to anomalous diffusion of ions through the porous surface of the electrode. The model includes the effect of the polarizable solution and diameter of the pores on the parameters of the FOE. The simulation results of the proposed electrical equivalent model obtained by changing its parameters are discussed along with the experimental results illustrating the behavior of single component FOE.**

***Keywords*-Single component fractional order element (FOE); anomalous diffusion; porous electrode.**

I. Introduction

Design and development of fractional order element (FOE) has become an important field of research owing to the growing interest in studying fractional order systems. The earlier technique for realization of a fractional order element was using different combinations of RC ladder networks [1], [2], [3], [4]. But recently some authors have reported realization of FOE by using physico-chemical phenomenon [5]. A single component fractional order element has been developed by the authors indigenously [6], [7], [8], where a capacitor type probe with porous surface behave as an FOE when dipped in ionic medium.

To understand the exact relationship among the fractional exponent of FOE and the fabrication parameters, the mechanism due to which the fractional order behavior is exhibited must be studied. Literature shows that anomalous diffusion of ions [9], [10] through fractal surface results in fractional order behavior. In case of the single component FOE, the porous PMMA coating on the electrode surface consists of self similar spherical pores. Thus the electrode surface is fractal in nature, ion diffusion through which results in the fractional behavior.

It is well known that using different circuit combinations of resistance (R) and capacitance (C), i.e., cross RC ladder network, domino ladder, tree structure, an FOE can be realized. Thus a single component FOE can also be modeled using some combinations of R and C.

In this paper, the relation between the parameters of single component FOE and the fabrication criteria has been identified. Also the effect of diffusion of ions through the porous electrode surface on the behavior of the FOE is explored. An electrical equivalent model is proposed which is required for designing the single component FOE with desired specifications.

The paper is organised in four sections. Section II deals with the basics of single component fractional order element and the proposed electrical equivalent model for it. In Section III the experimental results showing the behavior of single component FOE as well as the simulation results obtained from the electrical equivalent model are discussed. Finally, in Section IV, concluding remarks and future scope have been looked into.

II. Electrical equivalent model of single component FOE

A single component fractional order element is fabricated by coating a copper plated epoxy glass with a porous film of poly-methyl-methacrylate (PMMA). This probe is dipped in a polarizable medium. The polymer coated copper probe when dipped in polarizable solution behaves as a fractional order element. The diagram of single component FOE is shown in Figure 1.

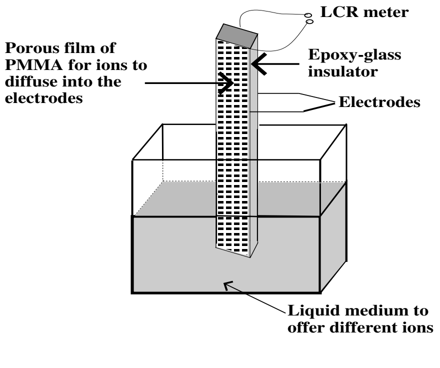

LCR meter
Porous film of PMMA for ions to diffuse into the electrodes
Epoxy-glass insulator
Electrodes
Liquid medium to offer different ions

Figure 1. Diagram of single component FOE

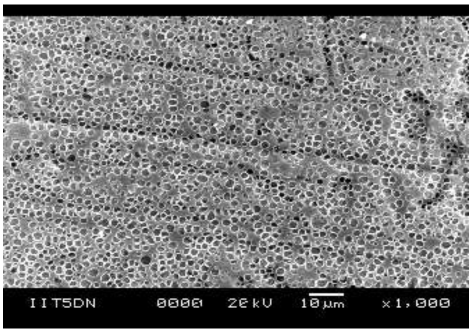


Figure 2. SEM image of the electrode surface of single component FOE with coating thickness $5\mu m$; Pore diameter = $0.85\mu m$

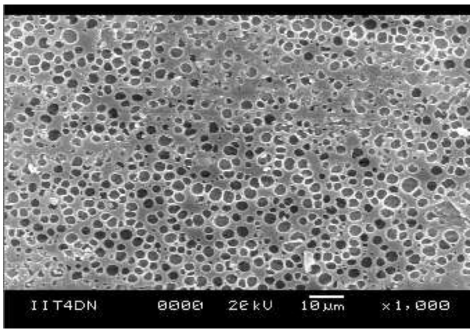


Figure 3. SEM image of the electrode surface of single component FOE with coating thickness $10\mu m$; Pore diameter = $1.27\mu m$

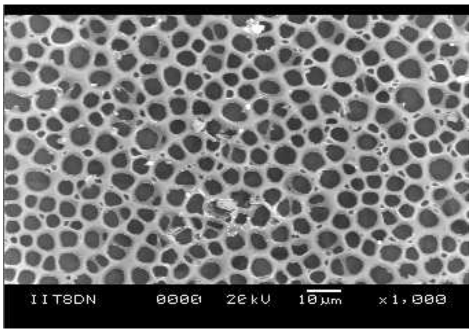


Figure 4. SEM image of the electrode surface of single component FOE with coating thickness $15\mu m$; Pore diameter = $2.24\mu m$

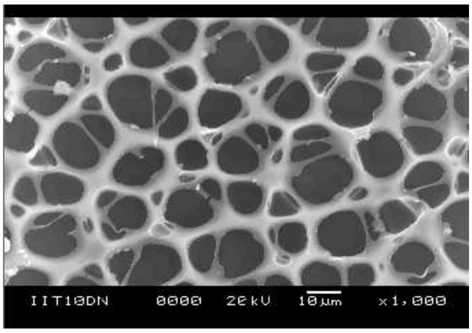


Figure 5. SEM image of the electrode surface of single component FOE with coating thickness $25\mu m$; Pore diameter = $4.85\mu m$

The impedance of a fractional order element is given by

$$Z(s) = Qs^{-\alpha} \tag{1}$$

where 'α' is the fractional exponent of the FOE.

Single component FOEs having different values of 'α' can be realized by varying the following fabrication parameters
i. Thickness of PMMA coating
ii. Concentration of the polarizable solution
iii. Area of contact of the electrode surface with the solution.

The fractional behavior of the probe arises due to the anomalous diffusion of ions through the porous surface of the single component FOE. In ordinary diffusion, the mean square displacement ($\langle r^2 \rangle$) of diffusing particles is directly proportional to time (t), i.e., $\langle r^2 \rangle \propto t$. On the other hand, in case of anomalous diffusion, $\langle r^2 \rangle$ has a power law distribution on time, i.e., $\langle r^2 \rangle \propto t^{\beta}$, where $\beta < 1$. This occurs due to the distribution of time constant resulting from ion diffusion through porous surfaces.

The surface of the FOE is porous in nature and the pores are more or less circular as is evident from the Scanning Electron Microscope (SEM) images (Figures 2, 3, 4, 5).

It can be seen from Figures 2, 3, 4, 5 that the pore diameter has an one to one relationship with the thickness of the PMMA film, i.e., the pore size increases with increase in thickness of the PMMA coating and vice versa.

The pores on the electrode surface are assumed to be spherical. At different layers of the film, the pore diameter is different. As the ions penetrate through the porous film, the resistance offered by the solution increases due to decrease in pore size (Figures 2, 3, 4, 5). The ions continue to penetrate till the pore diameter is less than the diameter of the ions. At this point the ions cannot move any further and become stationary.
Let, diameter of ion be 'd_i';
Thickness of the PMMA coating be 't';
Pore diameter corresponding to coating thickness 't' be 'D';
Number of branches the ions penetrate be 'n' and
Ratio of decrease of the pore diameter between two successive layers be 'a'.

The ions can penetrate till the pore diameter is less than 'd_i'. This can be mathematically represented by

$$\frac{D}{a^n} \leq d_i \tag{2}$$

The movement of the ions through the porous film can be modeled using a tree network consisting of resistances and capacitances. The electrode surface consists of numerous pores. Considering a single pore, the cross sectional view of the film can be represented as shown in Figure 6.

The porous structures are considered to be self similar in nature, i.e., from each pore 'N' number of smaller pores

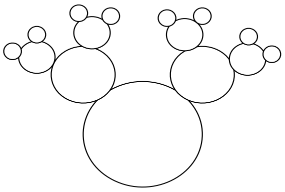


Figure 6. Self similar porous structure on the electrode surface

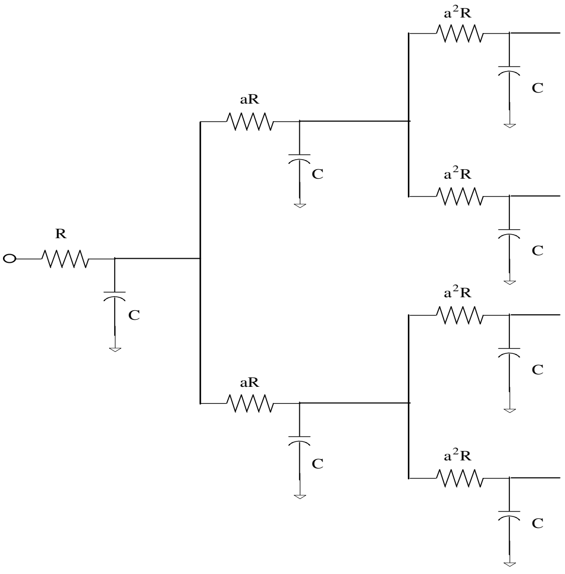


Figure 7. Equivalent tree network of the porous electrode surface

branch out and this structure continues. Each spherical pore can be represented by a simple series RC circuit. The equivalent circuit is shown in Figure 7.

The resistance value depends on the resistivity (ρ) of the polarizable solution and the surface area of the pore.

$$R = \frac{\rho D}{4\pi(\frac{D^2}{4})} = \frac{\rho}{\pi D} \tag{3}$$

The capacitance C is the interfacial capacitance between two layers of the film.

$$C = \frac{4\pi(\frac{D^2}{4})\epsilon}{D} = \pi\epsilon D \tag{4}$$

where, ϵ is the permittivity of PMMA. The impedance of the circuit shown in Figure 7 can be represented by [11]

$$Z_n = R + \frac{1}{j\omega C+}\frac{2}{aR+}\frac{1}{j\omega C+}\frac{2}{a^2R}\cdots \tag{5}$$

The tree form of Figure 7 is equivalent to the ladder network [12] as shown in Figure 8.

In the above circuit, 'N' is the number of sub branches from each node.
Geometric ratio of resistance $(g) = a/N$

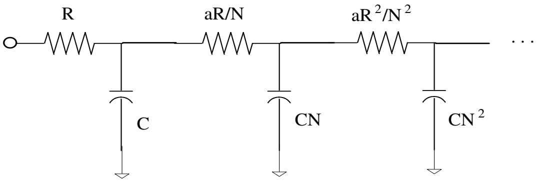


Figure 8. Equivalent ladder network of the porous electrode surface

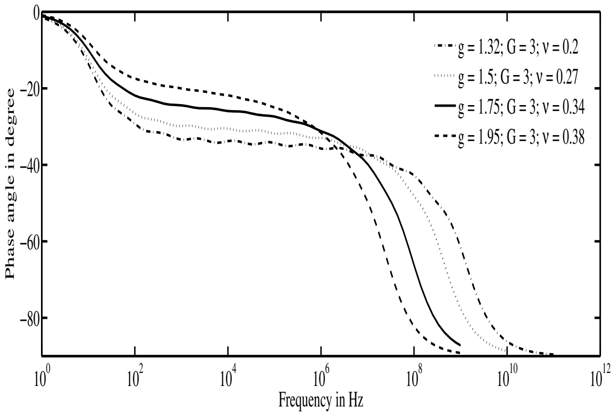


Figure 9. Simulation results showing change in 'α' with change in ratios 'g' and 'G'

Geometric ratio of capacitance $(G) = N$
Parameter 'ν' is such that [3], [13]

$$\nu = \frac{lnG}{ln(Gg)} = \frac{lnN}{lna} \tag{6}$$

Thus, the fractional exponent of the ladder network is given by

$$\alpha = 1 - \nu = 1 - \frac{lnG}{ln(Gg)} = 1 - \frac{lnN}{lna} \tag{7}$$

'ν' can be defined as the fractal dimension of the single component FOE. For the model to behave as a fractional order element, the value of 'N' should be greater than 1.

III. Inferences from the electrical equivalent model

Simulation of the ladder network given in Figure 8 shows that the value of 'α' depends on the ratios 'g' and 'G' (Figure 9).

It has been observed that by changing the product of R and C the bandwidth of constant phase angle can be shifted. On increasing the product RC the bandwidth shifts to lower frequency range and vice versa. This phenomenon is illustrated in Figure 10.

Also the frequency range of constant phase angle can be varied by changing the value of 'n' (Figure 11).

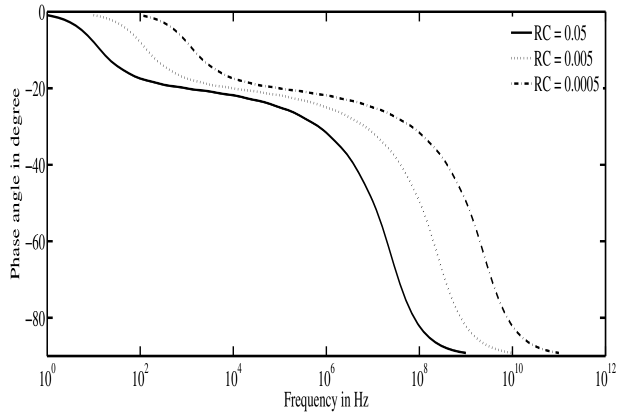


Figure 10. Simulation results showing effect of the product RC on bandwidth of constant phase angle

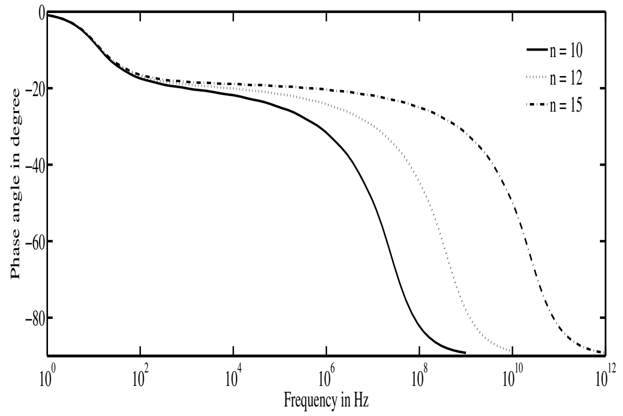


Figure 11. Simulation results showing change in bandwidth of constant phase angle with the value of 'n'

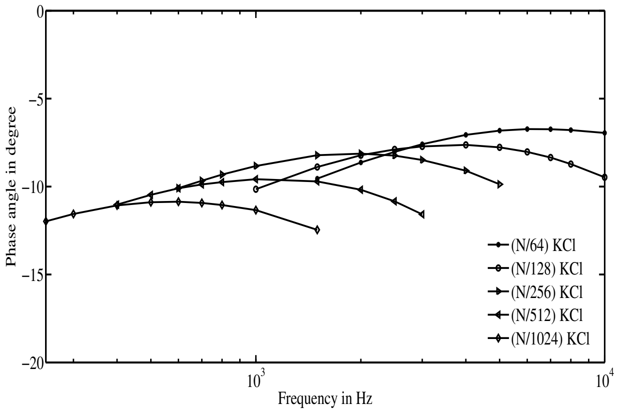


Figure 12. Experimental results showing shift in bandwidth of constant phase angle and change in 'α' value with concentration change of KCl solution for FOE5

From (7) it can be observed that 'α' depends on 'a' and 'N'. The value of 'α' increases with an increase in 'a'. It has been found experimentally that 'α' increases with the thickness of the PMMA coating (Table. I). Hence, 'a' is directly proportional to the PMMA film thickness. The fractional exponent, 'α' also depends on the concentration of the polarizable medium. This dependence can be incorporated in the theoretical model in terms of the value of 'n'. Experiments show that the constant phase angle of a single component FOE increases with the decrease in concentration. This can be seen from Figure 12 and Tables. II and III. Thus, we can infer that with concentration of the ionic medium, the value of 'n' increases. The increase in 'n' results in decrease in value of 'α' and increase in bandwidth of constant phase angle. From (2) it is obvious that if 'n' increases keeping 'D' and 'd_i' constant, the value of 'a' decreases and in turn 'α'. An important observation is that the bandwidth of constant phase angle shifts to lower frequency range as concentration decreases. This is because when concentration of the polarizable solution decreases, its conductivity decreases. Hence, the resistance offered by the solution increases. The interfacial capacitance remaining constant, the product of R and C increases, thus shifting the bandwidth of constant phase angle to lower frequency range. This is evident from Figure 12.

Keeping concentration of polarizable solution fixed, i.e., 'n' constant, if different polarizable solutions are used then the solution having ions of larger radius results in smaller 'α' value as evident from (2) and Tables. II and III.

IV. Conclusion and Future Work

In this work, an electrical equivalent model of the single component FOE has been proposed which takes into account the effects of anomalous diffusion of ions through porous surface and fractal dimension. The factors affecting the value of the fractional exponent 'α' and also their relationship with 'α' have been identified. The electrical equivalent model will help in predicting the behavior of single component FOE and also designing an FOE with desired specifications. The exact mathematical relation among the fabrication parameters and 'α' is yet to be established and needs further research.

References

[1] G. E. Carlson and C. A. Halijak, "Approximation of fractional capacitors $(1/s)^{1/n}$ by a regular newton process," *IEEE Transactions on Circuits and Systems*, vol. CAS-11, pp. 210–213, June 1964.

[2] S. C. D. Roy, "On the realization of a constant-argument immittance or fractional operator," *IEEE Transactions on Circuit Theory*, vol. CT-14, pp. 264–274, September 1967.

[3] K. B. Oldham and C. G. Zoski, "Analogue instrumentation for processing polarographic data," *Journal of Electroanalytical Chemistry*, vol. 157, pp. 27–51, 1983.

Table I
EXPERIMENTAL RESULTS SHOWING THE CHANGE OF 'α' WITH PMMA COATING THICKNESS

Name	Thickness (μm)	Pore size (μm)	Polarizable medium	Conductivity (mS/cm)	Q	α	Frequency range
FOE1	10	1.27	pH7.89	0.24	20.34×10^4	0.31	$100\ Hz - 100\ kHz$
FOE2	15	2.24	pH7.89	0.24	46.28×10^4	0.32	$100\ Hz - 4\ kHz$
FOE3	45	12.37	pH7.89	0.24	73.08×10^6	0.66	$200\ kHz - 1\ MHz$
FOE4	50	14.72	pH7.89	0.24	46.41×10^7	0.76	$200\ kHz - 1\ MHz$

Table II
EXPERIMENTAL RESULTS SHOWING THE EFFECT OF CONCENTRATION OF POLARIZABLE SOLUTION (KCl) ON 'α' AND BANDWIDTH OF CONSTANT PHASE ANGLE; IONIC RADIUS OF K = 1.38 $Angstrom$

Name	Thickness (μm)	Pore size (μm)	Polarizable medium	Conductivity (mS/cm)	Q	α	Frequency range of constant phase angle	Bandwidth (decade)
FOE5	12	1.64	(N/64) KCl	2.7	7.07	0.084	$1.5\ kHz - 20\ kHz$	1.12
FOE5	12	1.64	(N/128) KCl	1.52	16.51	0.094	$1\ kHz - 10\ kHz$	1
FOE5	12	1.64	(N/256) KCl	0.85	34.24	0.099	$600\ Hz - 5\ kHz$	0.92
FOE5	12	1.64	(N/512) KCl	0.48	75.02	0.115	$400\ Hz - 3\ kHz$	0.875
FOE5	12	1.64	(N/1024) KCl	0.27	136.92	0.127	$250\ Hz - 1.5\ kHz$	0.78

Table III
EXPERIMENTAL RESULTS SHOWING THE EFFECT OF CONCENTRATION OF POLARIZABLE SOLUTION ($NaCl$) ON 'α' AND BANDWIDTH OF CONSTANT PHASE ANGLE; IONIC RADIUS OF Na = 1.02 $Angstrom$

Name	Thickness (μm)	Pore size (μm)	Polarizable medium	Conductivity (mS/cm)	Q	α	Frequency range of constant phase angle	Bandwidth (decade)
FOE5	12	1.64	(N/32) NaCl	3.4	13.26	0.097	$800\ Hz - 10\ kHz$	1.097
FOE5	12	1.64	(N/64) NaCl	2.1	25.78	0.099	$500\ Hz - 6\ kHz$	1.079
FOE5	12	1.64	(N/128) NaCl	0.92	68.51	0.115	$200\ Hz - 2\ kHz$	1
FOE5	12	1.64	(N/256) NaCl	0.59	197.05	0.138	$100\ Hz - 800\ Hz$	0.9

[4] M. Sugi, Y. Hirano, Y. Miura, and K. Saito, "Frequency behavior of self-similar ladder circuits," *Physicochemical and Engineering Aspects*, pp. 198–200, 2002.

[5] G. W. Bohannan, "Application of fractional calculus to polarization dynamics in solid dielectrics materials," in *PhD dissertion*, 2000, Montana, State University, Bozeman.

[6] K. Biswas, S. Sen, and P. K. Dutta, "Realization of a constant phase element and its performance study in a differentiator circuit," *IEEE Transactions on Circuits and Systems-II, Express Briefs*, vol. 53, pp. 802–806, September 2006.

[7] K. Biswas, *Studies on Design, Development and Performance Analysis of Capacitive Type Sensors*. PhD thesis, Indian Institute of Technology Kharagpur, India, Department of Electrical Engineering, February 2007.

[8] M. Sivaramakrishna, S. Das, K. Biswas, and B. Goswami, "Characterization of a fractional order element realized by dipping a capacitive type probe in polarizable medium," in *Symposium on Fractional Signals and Systems Lisbon09*, November 2009.

[9] J. Bisquert and A. Compte, "Theory of the electrochemical impedance of anomalous diffusion," *Journal of Electroanalytical Chemistry*, vol. 499, pp. 112–120, 2001.

[10] A. Lasia, "Electrochemical impedance spectroscopy and its applications," *Modern Aspects of Electrochemistry*, vol. 32, pp. 143–248, 1999.

[11] S. H. Liu, "Fractal model for the ac response of a rough surface," *Physical Review Letters*, vol. 55, no. 5, pp. 529–532, 29 July, 1985.

[12] B. Sapoval, J. N. Chazalviel, and J. Peyrière, "Electrical response of fractal and porous interfaces," *Physical Review A*, vol. 38, no. 11, pp. 5867–5887, December 1, 1988.

[13] M. Moshrefi-Torbati and J. K. Hammond, "Physical and geometrical interpretation of fractional operators," *J. Franklin Inst.*, vol. 335B, no. 6, pp. 1077–1086, 1998.

Electromagnetic Imaging System for Weapon Detection and Classification

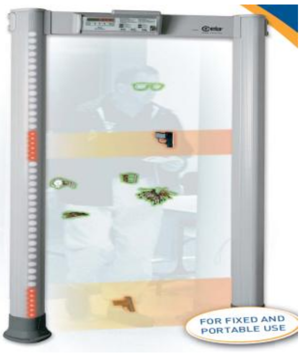
Abdalrahman Al-Qubaa, Gui Yan Tian, John Wilson
School of Electrical, Electronic and Computer Engineering
Newcastle University, Newcastle upon Tyne, NE1 7RU, UK
Abd.qubaa@ncl.ac.uk, g.y.tian@ncl.ac.uk, john.wilson2@ncl.ac.uk

***Abstract*- The detection of concealed weapons is one of the biggest challenges facing the security community. It has been shown that each weapon can have a unique fingerprint, which is an electromagnetic signal determined by its size, shape, and physical composition. Extracting the signature of each weapon is one of the major tasks of any detection system. This paper addresses the issue of identifying conductive objects based on their response to electromagnetic fields. A system developed at Newcastle University using a walk-through metal detector with a giant magneto-resistive sensor array to measure the spatial magnetic field is used in the study. Threat and non-threat objects have been tested. Image visualization and feature extraction of the electromagnetic field were carried out. Classification was performed using cross correlation. Promising results indicating the feasibility of using electromagnetic imaging to identify objects have been found.**

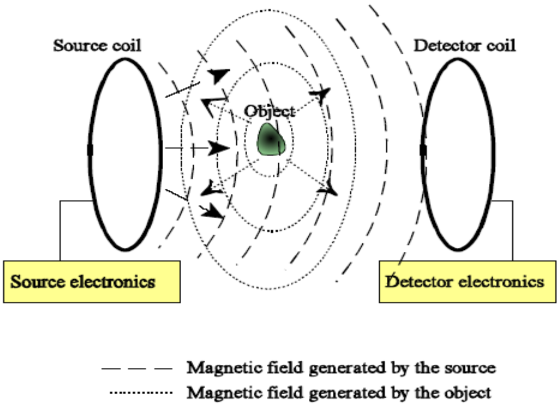
Keywords*- sensor array; electromagnetic imaging; weapon detection; object recognition; metal detection and classification.

I. INTRODUCTION

There is a growing need for effective, quick and reliable security methods and techniques to identify weapon threats using new screening devices [1]. Electromagnetic (EM) weapon detection has been used for many years, but object identification and discrimination capabilities are limited. Many approaches and systems/devices have been proposed and realised for security in airports, railway stations, courts, etc. The fact that most weapons are made of metallic materials makes EM detection methods the most prominent and systems/devices built on the principle of EM induction have been prevalent for many years for the detection of suspicious metallic items carried covertly [2]. Walk-through metal detector (WTMD) and Hand-held metal detector (HHMD) are commonly used devices for detecting metallic weapons and contraband items using the EM field. Most of the WTMD and HHMD units use active EM techniques to detect a metal objects [1, 3, 4] see Fig. 1. Active EM means that the detector sets up a field by a source coil and this field is used to probe the environment. The applied/primary field induces eddy currents in the metal under inspection which then generate a secondary magnetic field that can be sensed by a detector coil. The rate of decay and the spatial behaviour of the secondary field are determined by the conductivity, magnetic permeability, shape, and size of the target. Sets of measurements can be then taken and used to recover the position, the size and the shape of the objects.



a)


Source coil
Detector coil
Object
Source electronics
Detector electronics
Magnetic field generated by the source
Magnetic field generated by the object

b)

Fig.1: a) PMD2 WTMD metal detector[3]. b) Diagram of a metal detector with an object inside the detection space[4].

Many other EM techniques have been also used in WTMD, such as Microwave imagers based on the EM Reflectometer principal [5], a wide bandwidth, time-domain EM sensor system to measure the eddy current time-decay response of a wide variety of metal targets [6], other advanced EM technique such as a magnetic real-time tracking vector gradiometer RTG using high-resolution fluxgate magnetometers used for incorporation into an unmanned underwater vessel to improve mine detection which comprises three primary three axis sensors and one three-axis reference sensor [7].

Currently available weapons detection systems are primarily used to detect metal and have a high false alarm rate because the WTMD works on adjusting threshold to discriminate between threat items and personal items, depending on the mass of the objects which means increasing the false alarm level (Fig. 2) [8, 9], also a human body has affected the sensitivity of the detector so when dealing with low conductivity or small materials, the human body could give a stronger signal than the material. This would cause the material to pass undetected, giving poor reliability [10].

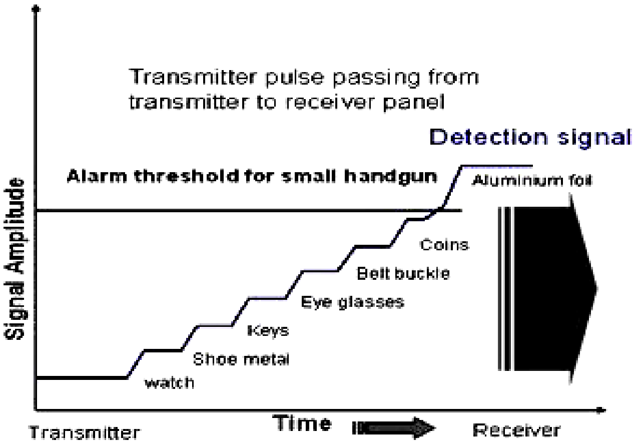


Fig. 2: Cumulative signal effects in active walkthrough weapon detector.

It can be concluded that the current EM imaging systems have several drawbacks such as: low-resolution images, the shape captured of EM signal not corresponding to the actual shape of objects, lack of detection with multiple objects, high cost for the 2-D array and the signal received correspond to the metal part only.

The aim of this study is to improve the characterisation capabilities of EM systems, especially in terms of object identification and classification, using visualization of an EM signal. A system developed at Newcastle University [11] and built in a lab using an ex-service CEIA WTMD, with the addition of a giant magneto-resistive (GMR) sensor array to capture the EM data, is used in this study. Features are extracted and integrated from the EM image to visualize, identify and classify metallic objects using cross correlation techniques.

This paper is organized as follows: Section II will describe the system design and set up. Section III will present the details of the classification approach and results and is followed by the conclusions and future work in Section IV.

II. TEST SYSTEM AND PRINCIPLES OF OPERATION

A. System design

The system used for the experimental tests is based around an array of NVE GMR sensors [12] used in conjunction with the excitation coil in an ex-service CEIA WTMD. Fig. 3 shows a block diagram of the system.

Fig. 4 illustrates the experimental system, converted from a typical walk through system. The signals from the sensor array are amplified using an array of signal amplifiers based on INA111 instrumentation amplifier. Data acquisition is performed using an 80 channel PXI based National instruments data acquisition system. The use of the PXI based system allows data to be acquired on 40 channels at a rate of 125kS/s or 80 channels at a rate of 62.5kS/s. The channel count is further increased to a maximum of 160 by the use of multiplexer circuits. A variable excitation waveform is provided by a function generator, the signal from which is also used for data acquisition synchronisation. The signal is amplified by a Kepco BOP 36-12ML bipolar power supply operating in constant current mode.

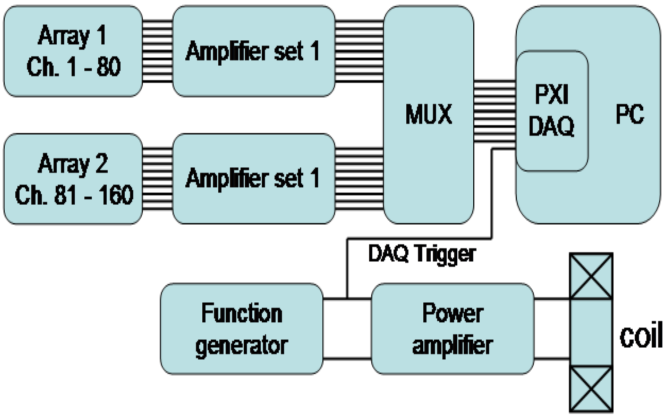


Fig. 3: System block diagram

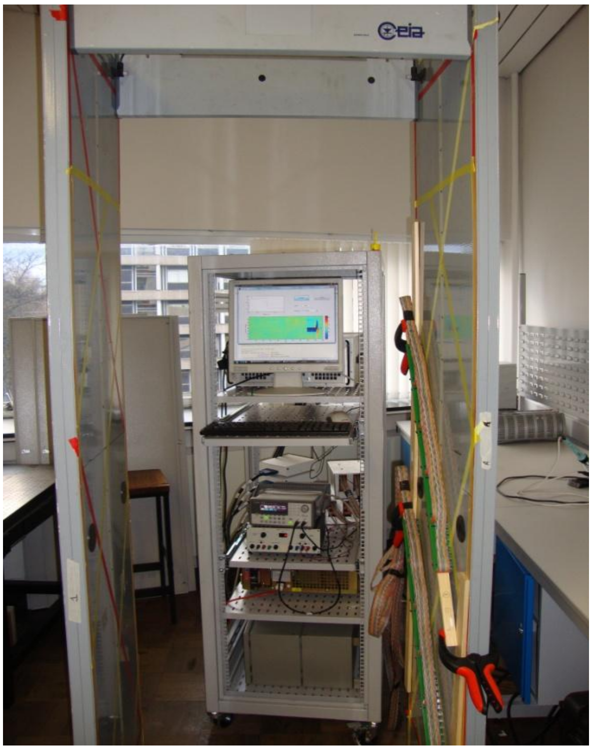


Fig. 4: System set up in a lab at Newcastle Uneversity.

The AAL002-02 GMR sensors were chosen because of their sensitivity and noise suppression compared with other common sensors such as Hall Effect models [13]. The AAL002-02 has a linear range of 0.15mT–1.05mT and a sensitivity of 4.5µV/T. The L in the sensor model name indicates that low hysteresis (maximum 2%) GMR material has been used fabricate the sensor, this was chosen because it was initially intended to utilise an applied magnetic field varying from 0 to a maximum value and the lower hysteresis would minimise error at low fields. However, after initial tests, it was found that a more stable signal could be achieved by biasing

ISBN: 978-1-61208-144-1

the sensor response into its linear region using a DC offset superimposed on the excitation signal.

B. *Electromagnetic field imaging*

Fig. 5 illustrates the different metallic objects and their EM field images. The samples represent common threat and personal objects carried by peoples. Five experiments have been carried out with each item and their capturing condition can be summarized below:

1. The object under inspection is moved through the detector with data acquired at a pulse repetition rate of 500Hz.
2. Sets of 10 pulses are averaged to produce a single pulse response to improve signal-to-noise ratio.
3. A single value is computed from each pulse response. In this case the maximum value of the difference signal (with object and without object) has been used.

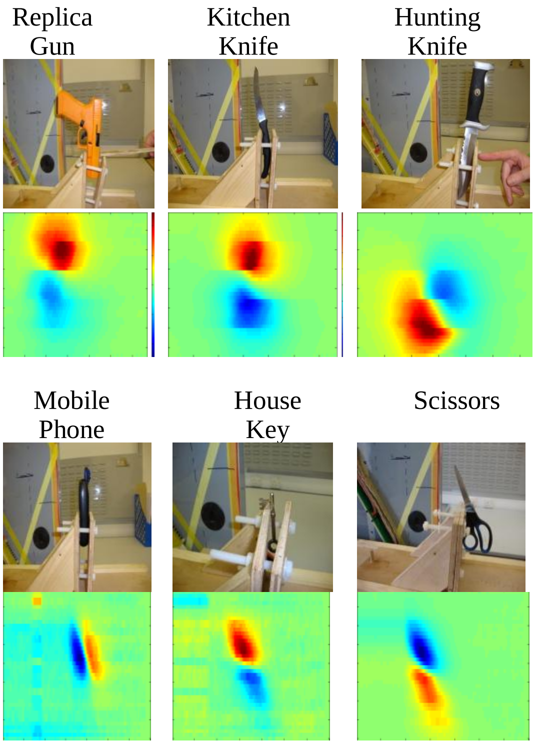


Fig. 5: Samples with the equivalent EMIs.

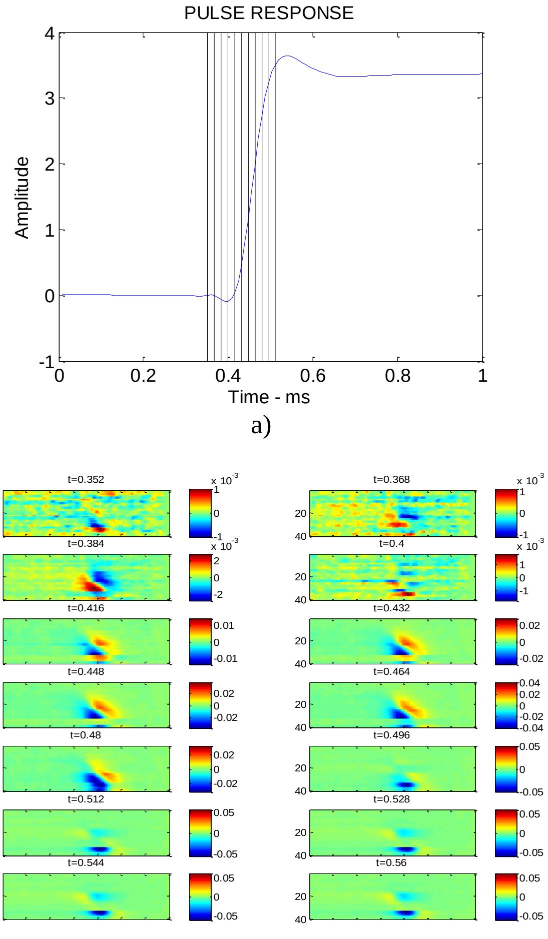
Thus the temporal EM field distribution as the object moves past the array can be determined. The sensors array is aligned with the coil to pick up any distortions in the applied field due to the presence of metallic materials, the interaction between the applied field and any sensor in the array will be capture and the pulse response from a group of sensors will be stored. If no object is present in the WTMD, the field measured by the sensor is unchanged; the presence of a metallic object causes a distortion of the field, which can then be measured by the sensor.

In the system, pulsed excitation is applied to the coil. Pulsed excitation provides the opportunity to apply an interrogating field with rich frequency components in a single waveform. In the tests detailed in this paper, a pulse repetition frequency of 500Hz is used with a pulse width of 1ms and an applied current of 0.5A – 1.5A.

C. *Transient analysis*

In order to extract more information about the objects in the WTMD from the test results, a form of transient analysis has been employed. In this transient EM signature imaging technique, the pulse response from each sensor is analysed and processed into sections, or time slots, as shown in Fig. 6a.

The values of the samples in each time slot are averaged, and using the data from all sensors for the whole test, an image is built up for each time slot. Fig. 6b shows a sequence of these transient images for the hunting knife.



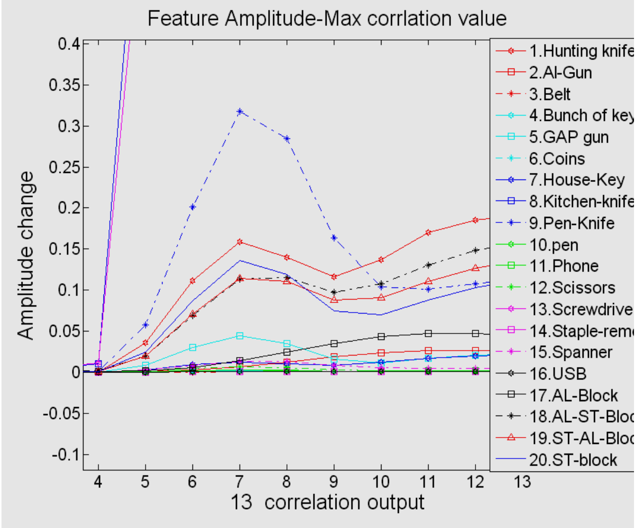
a)

b)

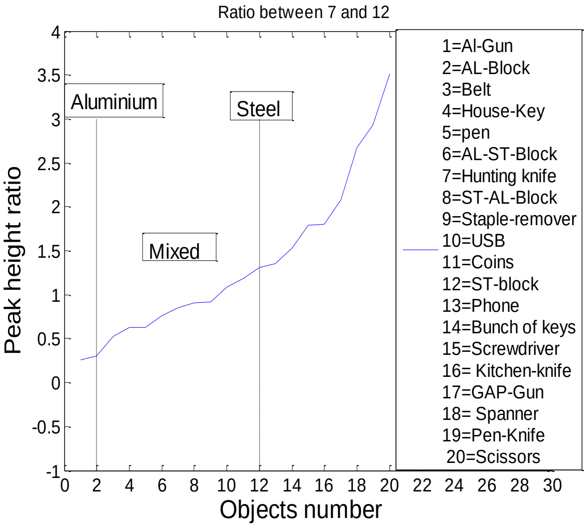
Fig. 6: a) Pulse response with time slots marked, b) Result of imaging the transient response from the hunting knife.

III. OBJECT CLASSIFICATION

Analysis of the transient image sequence can be used to extract more information about the object under examination especially for object classification. For example, it has been observed that aluminium objects exhibit a tendency for the EM signature to appear later in the image sequence and to increase in intensity over time. In contrast, the EM signatures corresponding predominantly to ferromagnetic objects, such as the hunting knife, have a tendency to appear earlier in the sequence, peak in amplitude at a particular point and to change in distribution over time. Consequently, through analysis of the image sequence generated by the transient analysis, any object detected by the system can be classified. An example of this is shown in Fig. 7, where a cross correlation technique has been applied to the transient image sequence [14] and processed to classify the objects into paramagnetic (aluminium=AL-block), ferromagnetic (steel=ST-block) and combinations of both. Fig. 7a represents a maximum cross correlation values between each two frames. Fig. 7b shows the result of computing the ratio between two peaks which are evident in the cross correlation plot shown in Fig. 7a, where different objects have unique transient features reflecting materials and geometrical characters. The results can be applied for object classification and are sorted by ascending amplitude. It can be seen from Fig. 7b and Table 1 that a clear distinction can be made between paramagnetic, ferromagnetic and mixed objects thus allowing very good discrimination.



a)



b)

Fig. 7: Material determination through transient analysis; a) Cross-correlation between images in transient sequence for 20 different objects, b) Peak height ratio for results shown in (a).

TABLE 1: CLASSIFICATION RESULTS

Class 1 Para-magnetic	Class 2 Mixed	Class 3 Ferro-magnetic
AL-Gun	ST-AL-Block	ST-Block
AL-Block	AL-ST-Block	Screwdriver
	Hunting-Knife	Kitchen Knife
	House-key	Pen-Knife
	Belt	Gap-gun
	Stapler-rem.	Scissors
	Coins	Spanner
	USB	Bunch of keys
	Pen	Phone

IV. CONCLUSION AND FUTURE WORK

This paper has demonstrated a new EM metal detector system and investigated the feasibility of visualizing the EM signal in a WTMD for object identification and classification purposes. A system to obtain the EM images has been built and features have been selected using the cross correlation between 14 EM frames to discriminate between the 20 different objects depending on the material properties. The system show promising results for the visualisation of EM signal especially in security applications.

In comparison with conventional induction based WTMDs, the GMR array based system has shown great potential in material discrimination as the samples are made from mixed material is clearly distinguished. Whereas the induction based WTMD can only discriminate between metal and non-metal, this system has taken it a step further. The proposed cross correlation technique is more advanced in object characterisation as it depends on the amplitude distribution of the EM field making training possible using a database of objects; unlike traditional thresholding adopted in the induction based system, which largely depends on material volume. On the whole the conventional WTMD based system had a limitation to present results in images, however, the proposed system has superior performance when using proposed sensor in terms of using imaging for localisation and material discrimination as buttressed by the results discussed in this paper.

The discrimination capabilities of the system could be developed to the point that individuals could pass through the system without removing metallic objects from their person. This would be realised through “training” the system to identify threat objects by presenting the system with a wide variety of threat and non-threat objects and programming the response accordingly.

Further study is necessary to extend these results to smaller metallic objects in not controlled environment to investigate concealed weapons. EM images from other objects will be investigated and compared with the results of this study. Other features such as: metallic density, total area of metallic density and metal proprieties will be investigated. Features will be optimised and prepared to use as input data in a classification algorithms.

ACKNOWLEDGMENT

This project is funded under the Innovative Research Call in Explosives and Weapons Detection (2007), a cross-government programme sponsored by a number of government departments and agencies under the CONTEST strategy. The authors would like to thank cross-government departments for joint experimental tests.

REFERENCES

[1] A. Agurto, Y. Li, G. Y. Tian, N. Bowring, and S. Lockwood, "A Review of Concealed Weapon Detection and Research in Perspective", Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control, London, pp. 443-448, 2007.

[2] C. A. Dionne, J. J. Schultz, R. A. Murdock, and S. A. Smith, "Detecting Buried Metallic Weapons in A Controlled Setting using a Conductivity Meter", Forensic Science International Vol. 208, pp. 18-24, 2011.

[3] CEIA Security Metal Detectors. [On-line 24.05.2011] http://www.ceia.net/security.

[4] N. G. Paulter, "Users' Guide for Hand-Held and Walk-Through Metal Detectors", NIJ Guide 600-00, NCJ 184433, Office of Science and Technology, U.S. Department of Justice, Washington, DC 20531, 2001. [On-line 31.05.2011] http://www.nij.gov/pubs-sum/184433.htm.

[5] D. M. Sheen, D. L. McMakin, and T. E. Hall, "Three-Dimensional Millimetre-Wave Imaging for Concealed Weapon Detection", IEEE transactions on microwave theory and techniques, Vol. 49, pp. 1581-1592, 2001.

[6] C. V. Nelson, C. B. Cooperman, W. Schneider, D. S. Wenstrand, and D. G. Smith, "Wide Bandwidth Time-Domain Electromagnetic Sensor for Metal Target Classification", IEEE Transactions on Geoscience and Remote Sensing, Vol. 39, pp. 1129-1138, 2001.

[7] S. Kumar, A. R. Perry, C. R. Moeller, D. C. Skvoretz, M. J. Ebbert, R. K. Ostrom, S. L. Bennett, and P. V. Czipott, "Real-Time Tracking Magnetic Gradiometer for Underwater Mine Detection", Proceedings of Oceans 2004. MTTS/IEEE Techino-Ocean '04, Vol. 2, pp. 874-878, 2004.

[8] US Department of Justice, "Final Report - Demonstration of a Concealed Weapons Detection System using Electromagnetic Resonances", January 2001. [On-line 31.05.2011] http://www.ncjrs.gov/pdffiles1/nij/grants/190134.pdf.

[9] J. Ashcroft, D. J. Daniels, and S. V. Hart, "Walk-Through Metal Detectors for use in Concealed Weapon and Contraband Detection", U.S. Department of Justice, Office Justice Programs, National Institute of Justice, 2003. [On-line 31.5.2011] http://www.nij.gov/pubs-sum/193510.htm.

[10] H. M. Chen, S. Lee, R. M. Rao, M. Slamani, and P. K. Varshney, "Imaging for Concealed Weapon Detection", IEEE Signal Processing Magazine, Vol. 22, pp. 52-61, 2005.

[11] G. Yun Tian, A. Al-Qubaa, and J. Wilson, "Weapon Detection and Classification using Magnetic Sensor Arrays", submitted to Sensors & Actuators A, 2011.

[12] J. Lenz and A. S. Edelstein, "Magnetic Sensors and Their Applications", IEEE Sensors Journal, Vol. 6, pp. 631-649, 2006.

[13] J. P. Sebasti´a, J. A. Lluch, and J. R. L. Vizca´ıno, "Signal Conditioning for GMR Magnetic Sensors: Applied to Traffic Speed Monitoring GMR Sensors", Sensors and Actuators A, Vol. 137, pp. 230–235, 2007.

[14] A. R. Al-Qubaa, G. Y. Tian, J. Wilson, W L Woo, and S S Dlay, "Feature Extraction using Normalized Cross-Correlation for Pulsed Eddy Current Thermographic Images', Measurement Science and Technology, Vol. 21, pp. 115501-115511, 2010.

High Repetition Rate Integrated Streak Camera in Standard CMOS Technology

Jean-Pierre Le Normand, Chantal-Virginie Zint, and Wilfried Uhring
InESS. Université de Strasbourg and CNRS
Strasbourg, France
e-mail: Jean-Pierre.Lenormand@unistra.fr; Virginie.Zint@unistra.fr; Wilfried.Uhring@unistra.fr.

***Abstract*— The capabilities of an Integrated Streak Camera (ISC) prototype fabricated in an AMS 0.35 µm CMOS process to observe light pulses of few nanosecond FWHM (Full width at half maximum) with or without analog accumulation is presented. The sensor consists of a matrix of 64 * 64 pixels including an electronic shutter and an analog accumulation capability. The sensor can operate in single shot mode when the light pulse power is sufficient and in repetitive mode, i.e., it can measure a recurrent light pulse and accumulates the successive photo charges into an internal node, for low light detection. The repetitive mode improves the sensitivity and the signal to noise ratio of the system. The functionality of a streak camera is reproduced through a versatile integrated temporal sweep unit that generates a sampling period continuously adjustable from 125 ps to DC. The distributed architecture and the in-pixel embedded memory allows the sampling rate to exceed 400 GS/s. Results on the dynamic performance of the ISC are presented. The measurements showed a temporal resolution close to the nanosecond and a repetition rate of more than 50 MHz. A 6 ns FWHM light pulse spatially focused on a single pixel row has been successfully measured in single shot and in accumulation mode.**

Keywords - CMOS integrated circuits, image sensors, integrated optoelectronics, time domain measurements

I. Introduction

High speed imaging is used in many scientific fields to measure phenomena which are not observable with naked eyes [1]-[4]. Ultra fast video cameras take a succession of images to observe transient events of about a microsecond [5]. The observation of faster transients is possible using a streak camera [6][7] which produces the one-dimensional spatial (*x*) intensity (*I*) (a streak) of a light pulse event as a function of the time (*t*). The output of the camera is a two dimensional image *I=f(x,t).* The streak camera offers the best temporal resolution in the field of direct detection in optics which can reach less than one picosecond [8]. However the use of vacuum tubes for this camera makes the system delicate, cumbersome and expensive. It also requires a high supply voltage (> 1 kV) and makes use of critical radio frequency electronics. The aim of our work is to develop an Integrated Streak Camera (ISC) in a standard CMOS technology that can reach a temporal resolution of the order of a few hundreds of picoseconds that actually corresponds to the use of streak camera for a lot of applications. Examples are photoluminescence from nanocrystals [9][10], velocimetry, interferometry to study shock waves of various materials [11]-[13], or in-life sciences for fluorescence lifetime imaging [14][15] and Förster resonance energy transfer [16]. In 2003, the first prototype of ISC, based on a pixel array, demonstrated the feasibility of an ISC with a sampling period in the order of one nanosecond and a temporal resolution of about 6 ns [17]. Nevertheless, this sensor operates only in single shot mode and presents a low signal to noise ratio. Indeed, it requires a derivative reconstruction process that increases the high frequency noise and implies a low output dynamic range.

The pixel architecture developed for the second generation of the CMOS spatiotemporal camera includes an electronic shutter and an analog accumulation feature within the pixel in order to increase the system sensitivity. In [18], the static characterization of the accumulation mode has been reported. In this paper, an improvement of the complete architecture and the dynamic performances of the accumulation mode are presented. The new design increases considerably the signal to noise ratio for both the single shot and the repetitive mode. Moreover, the evaluated maximal repetition rate is above 50 MHz which is comparable to the fast repetition rate achievable with a conventional streak camera in synchroscan mode [19]. In the next section, we present the overall architecture and operation of the ISC and its pixel. After that, the dynamic characteristics and some experimental results of optical pulse measurements with and without accumulation are shown.

II. Architecture of the Integrated Streak Camera

A. *Overall architecture and operation principle*

The functionality of the MISC (for Matrix based ISC) sensor is depicted in Fig. 1. For this matrix sensor architecture, in which the spatial axis is directed along the columns and the temporal axis is associated to the rows, the light signal can be correctly discriminated only if the spatial light intensity on a row is well known or, ideally, uniform. The advantage of creating a 2D array is to have several spatial channels in one chip, i.e., a conventional streak camera-like image *I=f(x,t).* One can take advantage of the proposed illumination setup in [20] including a mechanical slit followed by a cylindrical lens in order to carry out the uniform light spreading operation while maintaining the

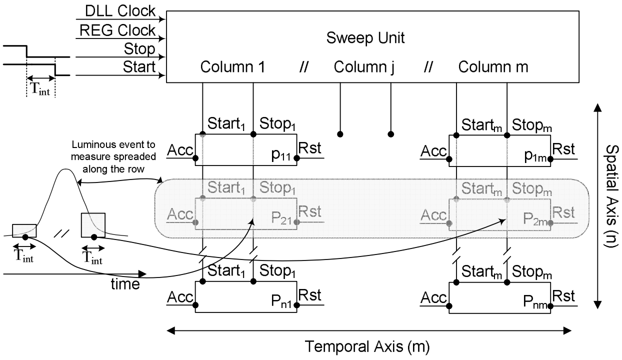


Figure 1. Architecture of the sensor and operation principle.

spatial resolution along the slit.

Let us consider a laser pulse train for the clarity of the explanation. We assume that this luminous event is punctual and focused on the slit, i.e., it is lighting uniformly a single row *i* of the MISC. For each pulse, the optical signal is integrated by the photodetector within the pixels p_{ij} with $j \in [1,m]$. The light is integrated during T_{int} for each pixel, but the beginning of the integration $Start_j$ for each column *j* is shifted by ΔT through the sweep unit. This operation is equivalent to the deflection of the electronic beam in an ordinary streak camera. Next, the corresponding photo-charges are stored and added to the previous ones inside the pixel by activating the *Acc* signal. Note that the operational sample rate of such a device is about 400 GS/s with a sample period ΔT of 150 ps and 64 lines. This very high data rate close to 10 Tb/s, cannot be continuously extracted from the sensor. In order to bypass the bottleneck of the I/O bus of the circuit, this architecture uses an embedded analog memory in each pixel to perform this operation. At the end of the acquisition, the data are extracted at a moderate rate of 20 Mpixel/s with a classical readout unit, not represented on Fig. 1.

The resulting signal of pixel p_{ij} is equivalent to a convolution between the illumination and a rectangle function with duration T_{int} and is given by:

$$p_{ij} = \sum_{k=1}^{N} \left(\frac{1}{m} G_c \int_{j\cdot\Delta T}^{j\cdot\Delta T+T_{\text{int}}} E_{ki}\left(t\right) \cdot dt \right), \quad (1)$$

where *N* is the number of accumulations, G_c is the global conversion gain and E_{ki} is the k_{nth} optical signal received by the row *i*.

B. Description of the sweep unit

The sweep unit previously reported in [18] consists of a cascaded chain of inverter with constant delays and fixed sweep speed. The new architecture of the sweep unit, depicted in Fig. 2, generates a flexible sweep speed in the range from 125 ps/pixel up to the DC operation. It consists of two main delay generators. The fast sweep unit (FSU) is useful for sub-nanosecond sample period operation. It is composed of a Delay Locked Loop (DLL) and two Voltage-Controlled Delay Lines (VCDLs). The observation time of the camera is set by the DLL reference clock period. The mirror VCDLs are driven by the same control voltages V_{hl} and V_{lh} as the master DLL and allow an asynchronous delay generation with respect to the reference clock. This makes it possible for the proposed generator to be launched from zero-tap position upon external triggering signal, ensuring the synchronization between the beginning of the acquisition procedure and the luminous phenomenon to be captured. The output taps of the VCDLs are delayed by an adjustable delay ΔT given by the ratio of the DLL Clock to the number of columns *m*.

The voltage controlled cell is a current starved double inverter with NMOS control transistors. It presents a large delay range and has been optimized for the shortest delay achievable in the CMOS technology employed [21]. The master DLL ensures the stability over temperature and the absolute precision of the delay. Nevertheless, the VCDLs must be exactly matched to ensure the same temporal shifting and duration of integrating window. The *Start* and *Stop* trigger signals are generated externally. Using two independent command signals allows reducing the integration time down to 1 ns or less. Results from a similar FSU have been reported in [22] and are summarized on table I.

The slow sweep unit (SSU) is useful for nanosecond up to DC operation. It uses a fast D flip-flop shift register. The sample period ΔT is equal to the frequency period of the register clock frequency *REG Clock*. The linearity of such a delay generator is very good as it is linked to the clock stability. Nevertheless, in an asynchronous operation, the timing jitter of the generated $start_j$ and $stop_j$ signals with respect to the trigger signal *start* and *stop* is equal to the sample period peak-to-peak.

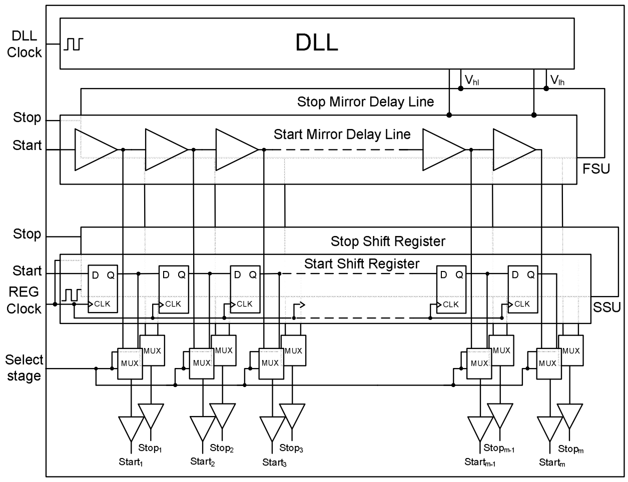


Figure 2. The sweep unit architecture.

ISBN: 978-1-61208-144-1

C. Description of the pixel

The architecture of the pixel is shown in Fig. 3. A Nwell/Psub photodiode is used in order to have a broadband light sensitivity and a low capacitance to enhance the conversion gain. The Start_i and Reset transistors reset the photodiode and the readout node (RN), respectively. Shuttering is carried out by the Stop_i transistor and the charges are transferred from the internal node (IN) to the readout node by the Acc transistor. The *Acc* and *Reset* commands are global for the entire matrix.

The static operation and characterization of the MISC obtained under continuous illumination in single shot and in accumulation mode and has been reported in [23]. In the next section, the repetitive mode with analog on chip accumulation is described and some experimental results are shown.

III. ANALOG ACCUMULATION FEATURE

A. Description of the analog accumulation operation

The accumulation mode is performed by several acquisitions of one repetitive pulse. The initial phase consists in resetting the readout node with the Reset transistors. Then the readout node RN is left floating, ready to accumulate several low intensity acquisitions. Each acquisition is composed of four operating phases: the reset of the photodiode, the integration of the light, the sampling and the charge transfer from the internal to the readout nodes. During the reset of the photodiode the Stop transistor is held high to have the same potential on the photodiode and on the internal node (Fig. 3). The integration begins when the command of the Start_i transistor is at low level and finishes when the command of the Stop_i transistor falls at low level. Indeed, on the falling edge of the signal $Stop_i$ the potential of the photodiode is sampled and hold in the IN. Finally, the charges stored in this node are transferred to RN by applying an intermediate voltage between V_{RR} and V_{RPD} to the gate of the transistor Acc (e.g., 2.5 V) during the transfer time T_{Acc} [18]. In the accumulation mode this sequence is repeated several times, without resetting the readout node.

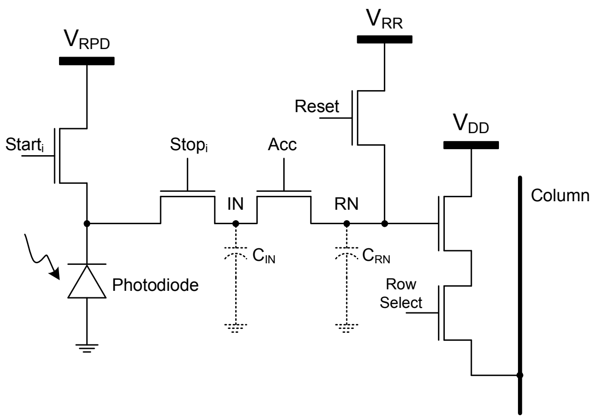


Figure 3. Pixel architecture of the MISC.

B. Analog accumulation performances

The maximal repetition rate of the acquisition is limited by the sum of the reset phase, the observation time $m{\cdot}\Delta T$ and the transfer time T_{Acc}. The reset phase occurs in less than 1 ns. The functions of the RN voltage versus the photodiode reset voltage (V_{RPD}) for different transfer durations from 1 ns up to 10 µs, with $V_{RR} = 3.3$ V and Acc = 2.5 V, are depicted on Fig. 4. These functions are composed of 3 regions. In the first region, the gain is zero, i.e., no charge is transferred as the Acc transistor is turned off, because V_{RPD} is above a threshold voltage (V_{TH}) depending on T_{Acc}. The second region illustrates the charge transfer operation that occurs when the photodiode voltage V_{RPD} is lower than the previous V_{TH} and above a second threshold voltage depending of T_{Acc}. Below this last voltage, the Acc transistor is completely turned on and the RN and IN nodes are shorted, i.e., voltage V_{IN} and V_{RN} are equal, consequently, the gain gets lower [23]. The region 2 is the usable mode for the accumulation process and the simulation results show that the charge transfer efficiency is independent of the T_{Acc} duration if the photodiode reset voltage is well adjusted (Fig. 4). In this case, V_{RN} is given by

$$V_{RN} = \frac{C_{IN}}{C_{RN}}\left(V_{IN} - \left(V_{Acc} - V_{TH}\right)\right) + V_{RN_0} \qquad (2)$$

where V_{RN0} the previous voltage of the RN. This indicates that the maximal repetition rate is mainly due to the temporal window as the reset and transfer phase can be reduced to a few nanoseconds.

This behavior arises from the transient phenomenon of the charge transfer. At the beginning of the charge transfer, the V_{GS} of the Acc transistor is well above its threshold voltage but as soon as the charges are transferred, the potential of the internal node V_{IN} increases very quickly,

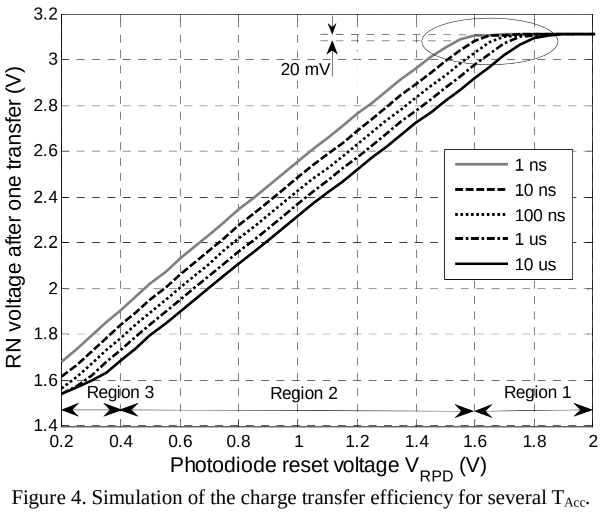


Figure 4. Simulation of the charge transfer efficiency for several T_{Acc}.

i.e., in less than 1 ns, reducing the V_{GS} down to the threshold. Then the transistor is turning off very slowly while the charge flow gets lower and lower. Consequently, when T_{acc} gets longer, more and more of those charges are transferred maintaining the transfer efficiency even with a higher V_{RPD}. For linear operation, the inflections in the V_{RN} to V_{RP} curves visible in the ellipse of the Fig. 4 must be avoid. Consequently, an overvoltage of 20 mV of dark signal must be added at each accumulation. Assuming a dynamic range of 1 V, the number of accumulations is limited to about 50 shots.

IV. MEASUREMENTS

A. Charge transfer efficiency

A 64 × 64 pixels prototype has been realized in the standard AMS 0.35 µm CMOS technology. In order to evaluate the highest repetition rate, the transfer gain efficiency versus the V_{RPD} for different T_{Acc} has been measured (Fig. 5).

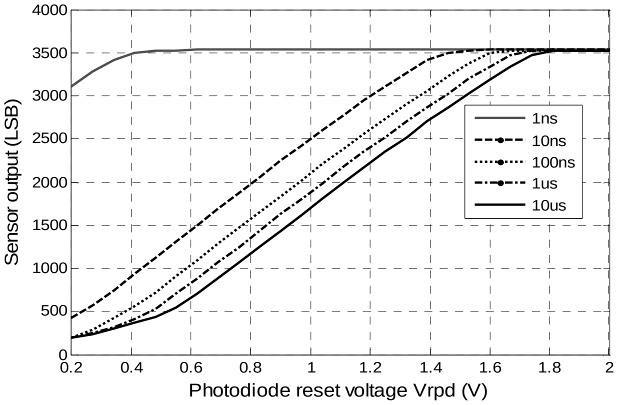


Figure 5. Measurement of the charge transfer efficiency for several T_{Acc}.

Fig. 5 shows that the transfer efficiency is effectively independent of T_{acc} if the photodiode reset voltage is well tuned. For T_{acc} equal 1 ns and 10 ns the curves are shifted to a lower V_{RPD} than expected by the simulation. This is due to the external pulse generator we use to generate the Acc control signal. Indeed, the rising and falling time of the signal delivered by this unit is about 1 ns, consequently, the effective T_{acc} is reduced for a pulse duration a few nanoseconds. Nevertheless, this measurement is the proof that T_{acc} can be reduced to a few nanoseconds, leading to a repetition period very close to the observation time.

B. Test bench for repetitive pulse observation

The experimental setup for the observation of repetitive optical pulses is show in Fig. 6. It is composed of a pulsed laser source, a synchronization unit, a delay line and an optical fiber. The accumulation is possible only if the MISC is synchronized with the source. Since the laser is subjected to jitter, the synchronization is operated by using a photodiode trigger unit. The delivered trigger signal is split between the *Start* signal of the sweep unit and the *Stop* signal. A flexible very short delay (a few nanoseconds or less) of integration ΔT is externally added by a passive jitter-less delay line. The laser source is delayed by an optical fiber to fit the capture temporal window of the MISC. The electronic setup inside the camera is composed of a FPGA that delivers the remaining command signals (Acc transistor, readout, etc.), a 12 bits analog to digital converter and the digital acquisition board. For this measurement, the FSU has been used with a 22 MHz reference clock which corresponds to an observation window of 45.5 ns and a sample period of 710 ps.

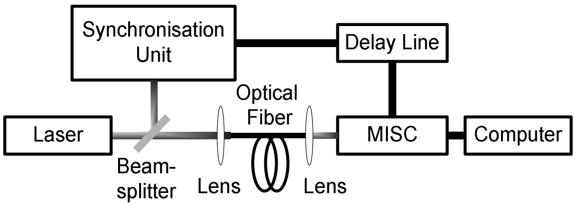


Figure 6. Experimental setup for repetitive pulses observation

C. Observation of light pulses in accumulation mode

A vanadate Q-switched diode pumped solid state laser which delivers pulses of 6 ns FWHM, at 532 nm, with a frequency repetition of 100 Hz has been used as the optical source. The laser beam was spread with a cylindrical lens in order to illuminate uniformly one row on the array sensor. A luminous pulse of low intensity, spatially focused on 20 µm, has been observed with three different operation modes of our sensor. The integration time used to obtain measurements presented on Figs. 7, 8 and 9 was fixed at 1 ns. With this short T_{int}, the convolution effect can be neglected with a pulse FWHM of several nanoseconds. On

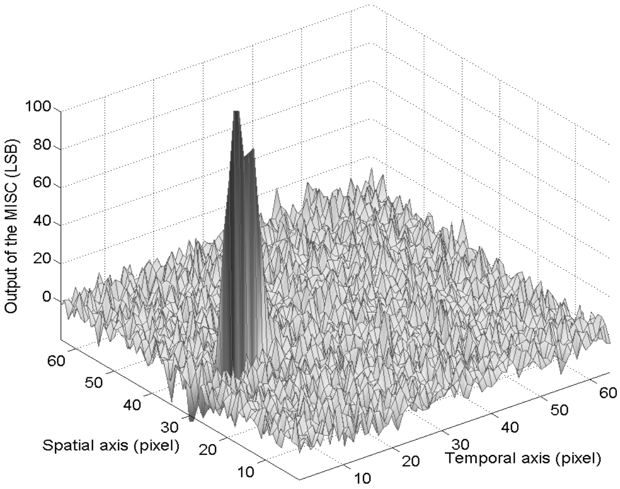


Figure 7. Measurement of a 532 nm laser pulse obtained in the single shot mode acquisition.

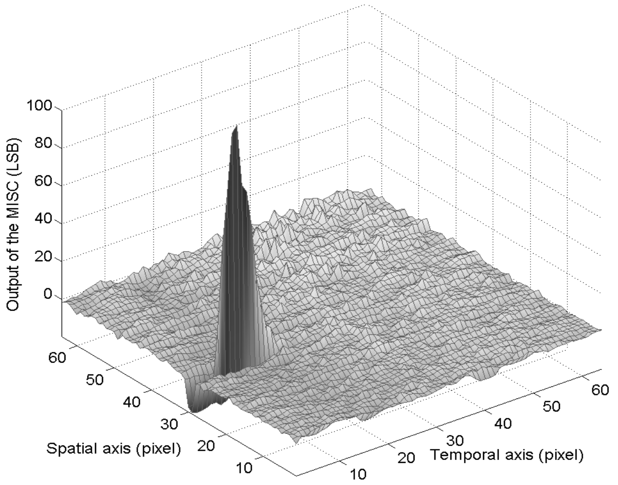


Figure 8. Measurement of a 532 nm laser pulse obtained with the computed averaging of 200 laser pulses.

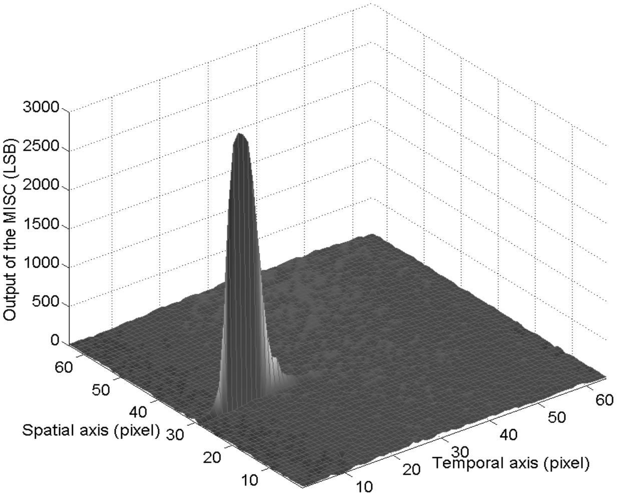


Figure 9. Measurement of a 532 nm laser pulse obtained with 20 analog accumulations.

all these images the dark level of each pixel has been subtracted in order to remove the fixed pattern noise (FPN).

Fig. 7 shows the luminous pulse measured in single shot mode. It is not easy to discern the pulse shape among the noise. Indeed, the signal to noise ratio (SNR), i.e, the ratio between the normalized Gaussian power and the normalized RMS noise, is evaluated to 29.

Fig. 8 presents the same signal measured without analog accumulation but with 200 frame software acquisitions, i.e., the resulting picture is the computed averaging of several single shot laser pulses. Of course the level of the luminous pulses has the same order of magnitude, but in this case, the effect of the readout noise is expected to be attenuated by a ratio of $200^{0.5}$. The measured SNR is 155. The improvement of the SNR by a factor 5.4 is less than then expected one. The difference probably results by some speckle effects than are not averaged with this technique.

Finally, Fig. 9 presents the luminous pulse resulting from 20 on-chip analog accumulations, without external acquisition. The signal amplitude is about 20 times higher than the one obtain in single shot with its associated photodiode KTC noise, whereas KTC noise of the RN and the readout noise are kept constant. Indeed, there are just one Reset of RN and one readout per acquisition The result shows that the readout noise is the dominant noise, thus, to first approximation, the expected SNR improvement ratio is roughly 20 while neglecting the photodiode KTC noise, the photonic noise and the speckle effect. The SNR obtained in this mode is 215, i.e. a improvement ratio of 7.4 compare to the single shot. This is higher than the software averaging for an order of magnitude less acquisition. Consequently, the analog on chip accumulation capability is a powerful feature, which improves the SNR of an ISC. Moreover, the repetition rate in this mode is not limited by the readout time but only by the transfer time. It is thus not limited to a few hundred of Hz but can be higher than 50 MHz.

D. Temporal resolution

During these measurements, the measured FWHM is 8 ± 0.5 pixels. As the sampling period is 710 ps/pixel, it means that the observed pulse duration is 5.7 ± 0.4 ns FWHM, which is very close to the 6 ns measured by a fast PiN photodiode and an oscilloscope. Consequently, the temporal resolution is better than 6 ns at 532 nm. Measurement of a femtosecond laser pulse at 400 nm with an integration duration ΔT of 400 ps indicates that the temporal resolution is 1.1 ns FWHM (see Fig. 10). Preliminary results obtained with the picoseconds laser diode generator described in [24] show a temporal resolution very close to the nanosecond too at 650 nm whereas it is degraded to more than 3 ns at 808 nm. This degradation of the temporal resolution is due to the slow transient response of the used Nwell/Psub photodiode at such a long wavelength.

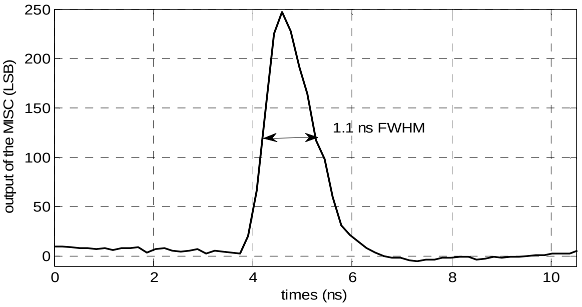


Figure 10. Measurement of a 400 nm femtosecond laser pulse obtained with the computed averaging of 500 laser pulses.

TABLE I. PARAMETERS AND PERFORMANCES OF THE MISC

Quantity	Value
Pixel pitch	20µm × 20µm
Fill factor	47 %
Conversion gain	4.8 ± 0.4 µV/e
Sweep speed (ΔT/pixel)	From 125 ps/pixel up to DC
Sweep speed drift	<0.05%/°C
Timing jitter	<70 ps p-p@150ps/pixel (FSU) [22] < 700 ps p-p@770ps/pixel (FSU) [22] <ΔT p-p Slow sweep unit (async SSU)
Temporal axis linearity	Better than 1%
Temporal resolution	1.1 ns @ 400 nm 3.6 ns@ 808 nm
SNR single shot	29
SNR with 200 software accumulation	155
SNR with 20 on chip analog accumulation	215
Max frame per second	500 Hz
Repetition rate in analog accumulation mode	From single shot up to 50 MHz
Max accumulated shot	50

V. CONCLUSION

This work showed that it is possible to realize an integrated streak camera in CMOS technology operating in a accumulation mode at a high repetition rate of more than 50 MHz. This system, carried out in a 0.35 µm AMS CMOS technology, allows the observation of short luminous phenomena in repetitive mode by analog accumulation of the photo-charges inside the sensor with a temporal resolution close to the nanosecond. The analog accumulation feature increases the SNR proportionally to the number of accumulations. However, the maximum number of accumulations is limited to 50 to avoid nonlinear operation. Combination of both the analog and software accumulation could be used to enhance the SNR to a higher value.

ACKNOWLEDGMENT

This work was supported in part by the French government and the region Alsace. The authors would thank Jérémy Bartringer for its technical assistance.

REFERENCES

[1] R.S.G. Baert, P.J.M. Frijters, B. Somers, C. Luijten, and W. de Boer "Design and operation of a high pressure, high temperature cell for HD diesel spray diagnostics: guidelines and results," *SAE Technical Papers*, No. 2009-01-0649.

[2] H. Zhang, I. Mudawar, and M.M. Hasan, "Experimental assessment of the effects of body force, surface tension force, and inertia on flow boiling CHF," *Int. J. Heat Mass Transfer*, vol. 45, pp. 4079–4095, 2002.

[3] M. Chilvers and C. O'Callaghan, "Analysis of ciliary beat pattern and beat frequency using digital high-speed imaging: comparison with the photomultiplier and photodiode methods," *Thorax*, vol. 55, pp. 314–317, 2000.

[4] S. Hertegard, H. Larsson, and T. Wittenberg, "High-speed imaging: applications and development," *Logoped Phoniatr Vocol*, vol. 28, pp. 133–139, 2003.

[5] T. G. Etoh *et al*., "An Image Sensor Which Captures 100 Consecutive Frames at 1,000,000 Frames/s," *IEEE Trans. on Electron Devices*, vol. 50, no. 1, pp. 144-151, Jan. 2003.

[6] T. Ito, M. Hiramatsu, M. Hosoda, and Y. Tsuchiya, "Picosecond time-resolved absorption spectrometer using a streak camera," *Rev. Sci. Instrum*., vol. 62, pp. 1415–1419, 1991.

[7] M.Y. Schelev, M. C. Richardson, and A.J. Alcock. "Image-converter streak camera with picosecond resolution," *Appl. Phys. Lett*., vol. 18, 354–357, 1971.

[8] W. Uhring, C. V. Zint, P. Summ, Y. Hu, and B. Cunin, "Very high long-term stability synchroscan streak camera," *Rev. Sci. Instrum*., vol. 74, pp. 2646–2653, 2003.

[9] R. Sarkar, A. K. Shaw, S. S. Narayanan, C. Rothe, S. Hintschich, A. Monkman, and S. K. Pal, "Size and shape-dependent electron–hole relaxation dynamics in CdS nanocrystals," *Optical Materials*, vol. 29, pp. 1310-1320, 2007.

[10] K. Zídek, F. Trojánek, B. Dzurnák, and I Pelant, "Spectral and dynamical study of nonlinear luminescence from silicon nanocrystals excited by ultrashort pulses," *Physica E*, vol. 41, pp.959-962, 2009.

[11] Ya. E. Krasik, A. Grinenko, A. Sayapin, and V. Tz. Gurovich, "Generation of sub-Mbar pressure by converging shock waves produced by the underwater electrical explosion of a wire array," *Physical Review E*, vol.73, pp. 057301, 2006.

[12] P. D. Washabaugh and L. G. Hill, "An initial investigation of the sub-microsecond features of dynamic crack propagation in PMMA and the RDX-based explosive PBX 9205," *AIP Conf. Proc*., vol. 955, pp. 727-730, 2007.

[13] J. P. Cuq-Lelandais, M. Boustie, L. Berthe, T. de Rességuier, P. Combis, J. P. Colombier, M. Nivard and A. Claverie, "Spallation generated by femtosecond laser driven shocks in thin metallic targets," *Journal of Physics D*, vol. 42, pp. 065402, 2009.

[14] M. Komura and S. Itoh, "Fluorescence measurement by a streak camera in a single-photon-counting mode," *Photosyntesis Research*, vol. 101, pp. 119-133, 2009.

[15] T. Bensky, L. Clemo, C. Gilbert, B. Neff, M. Moline and D. Rohan, "Observation of nanosecond laser induced fluorescence of in vitro seawater phytoplankton," *Applied Optics*, vol.47, pp. 3980–3986, 2008.

[16] C. Biskup, T. Zimmer, L. Kelbauskas, B. Hoffmann, N. Klöcker, W. Becker, A. Bergmann, and K. Benndorf "Multi-dimensional fluorescence lifetime and FRET measurements," *Microscopy Research and Technique*, vol. 70, pp. 442-451, 2007.

[17] B. Casadei, J-P. Le Normand, Y. Hu, and B. Cunin, "Design and Characterization of a Fast CMOS Multiple Linear Array Imager for Nanosecond Light Pulse Detections," *IEEE Trans. Instrumentation and Measurement Technology*, vol. 52, pp. 1892-1897, 2003.

[18] F. Morel, J.P. Le Normand, C.V. Zint, W. Uhring, Y. Hu, and D. Mathiot, "A New Spatiotemporal CMOS Imager With Analog Accumulation Capability for Nanosecond Low-Power Pulse Detections," *IEEE Sens. J*., vol. 6, pp. 1200-1208, 2006.

[19] W. Sibbett, "Synchroscan streak camera systems," *Proc. SPIE*, vol. 348, pp. 15–26, 1982.

[20] M. Zlatanksi, W. Uhring, J.P. Le Normand, C.V. Zint, and D. Mathiot, "Streak camera in standard (Bi)CMOS (bipolar complementary metal-oxide-semiconductor) technology**",** *Meas. Sci. Technol.* 21, 2010, 115203_1-12.

[21] N. R. Mahapatra, A. Tareen, and S. V. Garimella, "Comparison and analysis of delay elements," *IEEE Circuits and Systems, MWSCAS-2002, The 2002 45th Midwest Symposium on*, vol. 2, pp. 473-476, August 2002.

[22] M. Zlatanski, W. Uhring, J.P. Le Normand, and D. Mathiot, "A Fully Characterizable Asynchronous Multiphase Delay Generator," *IEEE Transactions on Nuclear Science*, 2011, 10.1109/TNS.2011.2106141 (In Press)

[23] F. Morel, J.P. Le Normand, C.V. Zint, W. Uhring, and Y. Hu, "A Spatiotemporal CMOS Imager for Nanosecond Low Power Pulse Detections," *Proceedings of IEEE Sensors*, vol.2, pp. 911–914, 2004.

[24] W. Uhring, C.V. Zint, and J. Bartringer, "A low-cost high-repetition-rate picosecond laser diode pulse generator," *Proc. SPIE*, vol. 5452, pp. 583-590, 2004.

Collaborative Approach for Secure Packet Transfer in Wireless Sensor Networks

Yenumula B. Reddy and Sanjeeve Kafley
Grambling State University
Grambling, LA 71245, USA
ybreddy@gram.edu

Rastko Selmic
Louisiana Tech University
Ruston, LA 71270, USA
rselmic@latech.edu

***Abstract*- Secure data transfer (SDT) in wireless networks is required with minimum overhead. The SDT was done historically through cryptography, authentication, and probability based approaches. Collaborative approach for trust-based packet transfer is new to the wireless sensor network research. In the proposed research, trust value of a node is continuously updated using Sporas formula and repeated trust calculations. The average of these two provides the trust value of a node. The suspicious node will be informed to the neighbor nodes. Further, the neighbor nodes calculate their own trust of a suspicious node using its trust value plus trust factor received from their neighbor. The cooperative and collaborative approaches eliminate the suspicious node from the path quickly and confidently. The results show that the new approach is better than simply using the cooperative way or collaborative approach using Sporas formula.**

Keywords: packet transfer, wireless sensor networks, collaborative approach, protocols, trust-based approach, resource.

1. Introduction

Massive deployment of sensors in hostile areas including forests, biological and chemical fields is very common and requires secure communication. Replacement of failing sensors or adding sensors to cover the black holes is very common in such dangerous places. Since they are organized in an open environment, injecting of bad nodes to corrupt the transmission is possible. Therefore, a secure transmission model is required to avoid the transmission through corrupted node. Further, design of secure communication model between sensor nodes is very difficult. The traditional protocols use exchange and distribute the keys through cryptographic tools for trust evidence. The resource limitations in sensor networks obstruct the use of traditional tools including cryptographic models and protocols for secure communications.

In sensor networks, the topology changes dynamically due to failure of sensors nodes. Further, sensing data and reporting data with limited communication distance requires cooperation of nodes to complete the task. The cooperation happens between the nodes only if they trust their neighbor to transfer the data. The trust management system helps to detect the node that is not behaving as expected in the path.

The trust depends upon the predictable behavior of other nodes in the network and builds upon continuous positive behavior. Further, trust depends upon the degree of belief based upon the experience. Trust is subjective, non-transferable, time dependent, contextual, and unidirectional. Due to the simplicity and effectiveness of the trust and reputation based models researchers' attention is diverted towards these models.

The trust starts with sensing the behavior of the neighbor node. The misbehavior is dropping the packets. The packet dropping may be due to malicious attacks (influence of bad nodes or intruders) on the node or the node is a sink hole. The trust can be measured through repeated positive behavior of the node. Reputation is a tool to detect the good behavior of the neighboring node [1]. The node could be assigned a reputation value to detect the behavior and keep track of the next node (forward path) in the path. To prove the successive node in the path is trustworthy, the current node should maintain a table. The table must contain the number of packets received and transmitted from the successive node. Further, it matches the table by overhearing from the next node, which transmits the packets to its neighbor. All nodes in the path will follow this process. The table includes the number of transmitted packets and will be initialized to zero after a set time. The method is simple with minimum resource utilization and easy to maintain. The design of such simple and low cost secure model is very important and an open research area.

Routing the packets in wireless sensor networks (WSN) is done by routing protocols. The routing procedure uses encryption, digital signature, and authentication. Further encryption and authentication limits the performance of nodes in WSNs. Encryption and authentication cannot prevent the malicious or misbehavior of the sensor nodes. After reviewing relevant sensor network models, we found that the trust-based model is a better model than the existing models. Since trust cannot be generated automatically, we use the verification of repeated data transfer in the successive node. The trust model detects the sinkholes, selective packet dropping, and malfunction of the node.

Once the trust is established, it cannot be taken for granted for the rest of the sensor lifetime without repeated reevaluations. The trust relationship changes continuously due to sensor failures and malfunctions. Therefore, the

ISBN: 978-1-61208-144-1

trust relationships among the neighboring nodes are very important to keep track of the uncertainties.

The remaining part of the paper discusses the recent developments, collaborative reputation activity, simulations using Sporas formula, reputation-based trust formulation, trust cluster approach and conclusion of results.

2. Recent Developments

If a user is given the work repeatedly and the user completes to the level of satisfaction, we say the user will be trusted. The same concept is used in credit cards, bank loans, and at work places. Sensor networks are not different when we consider the trust. The Figure 1 shows the scenario of a senor network. The nodes A, B, C are transferring the data to their successive node D, where D transfers to its next successive node E. The level of trust of a node D depends upon the percentage of packets successfully transferred to its next successive node E. The trust of node D depends upon the behavior of node D at a given time. The trust evaluation of node D also monitored through the neighboring nodes (node B and node C within communication distance) of A.

Trust values are derived in [2] by evaluating risk and reputation. In [3] the authors developed an algorithm to calculate trust using the complaints of another agent. Reputation-based framework using Bayesian formulation was developed by Generiwal et al. [4]. The proposed system uses community trustworthy behavior of the sensor nodes. The trust calculation in WSN using a bio-inspired algorithm (BTRM-WSN) based on ant colony systems was presented in [5]. The system uses similarity of how the ants' deposited pheromone helps to trace the path and quality of path by trusting the deposited pheromone.

Momani et Al. [6], explained the difference between trust, security, and reputation. Further, authors introduced the WSN security issues and innovative approaches to solve these problems. The authors concluded that the future research follows the innovative approach to model trust-based approach in WSN.

Task-based trust management, event-based trust management and an agent-based trust management was studied in [7-11]. In [7], a general approach for task-based trust management is used similar to economics to detect the malicious node. The event-based approach [8] uses several trust ratings to enforce the security in WSN. The agent-based trust models in [8-11] discuss the attacks on WSN, packet dropping, and local storage management using the trust policy. The models can further discusses the trust aggregation, Hello flood attack, and detect the malicious nodes.

Zhang et al. [12] presented a trust-based approach to distinguish illegal nodes from legal nodes. They claim that their approach detects insider attacks and uses trust evaluation model. The trust management model in [13] uses the Bayesian probabilistic approach. The current model calculates the trust factor by using the current trust factor plus the second hand information received from its neighboring nodes.

3. Collaborative Reputation activity

Reputation builds the trust in a specific domain. Reputation-based trust was discussed in [14] and defined as the amount of trust influenced by a person or node in a specific domain. Like with human relationships, reputation values associated with a node may change over a time. Therefore, it is advised to update the node ratings using current ratings. This procedure helps to calculate better trust factor. The reason for changing the trust rates overtime is that the node may get corrupt due to malicious activity.

Assume that each node entering in the sensor network has a minimum reputation value, i.e., initially every node transfers packets correctly. The node value will be updated after a set time period. A threshold value (trust value) will be set to decide either the node will continue in the communication path or discord at the end of each set period. In a sensor network, a new node can join the existing set of sensors or a malicious sensor will be discarded from communicating path (Similar to an electronic market, a new user may join in the group or untrusted member may discontinue). The reputation value of a current node should not fall below the newly joined node. A node can rate the neighboring node more than once, but the current rating will be taken. A higher rated node will have smaller change after each rating (unless the node is corrupted).

The sample rating of a neighboring node is done depending upon the trust. Trust of each node depends upon the opinion of other nodes, particularly neighboring nodes. Trust of a node is continuous updating through rating. The current reputation of a node (trust level) is updated using the following Sporas formula [15].

$$R_i = R_{i-1} + \frac{1}{\theta}.\phi(R_{i-1}).(W_i - R_{i-1}) \qquad (1)$$

$$\phi(R_{i-1}) = 1 - \frac{1}{1 + e^{-(R_{i-1}-D)/\sigma}} \qquad (2)$$

where:

θ - effective number of ratings taken into account $(\theta > 1)$. The change in rating should not be very large.

ϕ - helps to slow down the incremental change

W_i - represents the rating given by the node i

D - range or maximum reputation value

σ - the acceleration factor to keep the ϕ above certain value (> threshold).

If the node is compromised, the rating will be smaller and $(W_i - R_{i-1})$ become negative. Therefore the current

ISBN: 978-1-61208-144-1

reputation slowly crosses below threshold and node declared as malicious.

In the Figure 1, the opinion poll on node D will be done by nodes A, B, C, and P, because these nodes communicates the packets through D to the base station. The node E cannot poll on D, since it receives the packets from D. Assume that there is a node Z between E and base station, then E can poll on D, because if Z becomes malicious then E may need to transfer the data through D to reach the base station. In the opinion poll, the node A may poll 70% and node B may poll 80%. Sporas formula updates the rating (helps to build the trust) on node D using the rates polled by the connected nodes.

4. Simulations Using Sporas Formula

Suppose the node A makes 50 transactions and each time the node A give its own rating depends upon the number of packets transmits by the node D. Figure 2 provides the ratings on node D obtained by node A. Similarly, the ratings at B and C were found. The random values selected to calculate the ratings are as below:

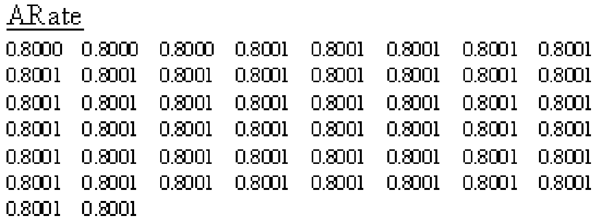
θ>10; $\quad$ 0.5<W_i<1;

0.6<D<0.99 $\quad$ 0<σ<0.5;

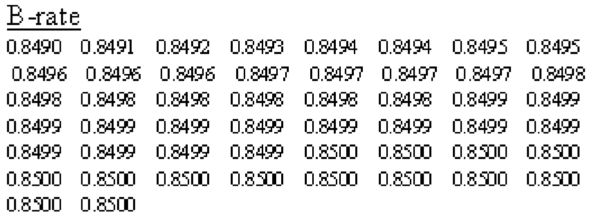
0.5<R_{i-1}<1;

Table I provides the ratings after 50 samples. Each time the rate was updated with the current value. At 50th time (current ratings at A, B, and C are 0.8001, 0.85, and 0.90. Suppose the threshold value is set at 0.85, then the node A requests the neighbor nodes C and D about the trust of node D. The nodes B and C give their trust value 0.85 and 0.9. The node A cannot discard the node D from its communication path, but it uses the reputation based trust calculation as given in the next section and will come to a conclusion.

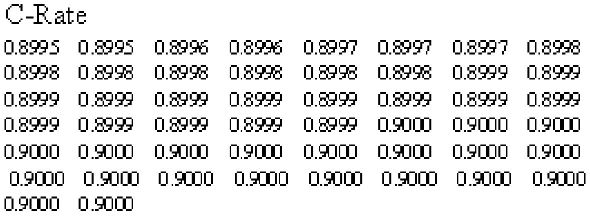
Table I: Ratings of node D at Nodes A, B, and C



```
ARate
0.8000  0.8000  0.8000  0.8001  0.8001  0.8001  0.8001  0.8001
0.8001  0.8001  0.8001  0.8001  0.8001  0.8001  0.8001  0.8001
0.8001  0.8001  0.8001  0.8001  0.8001  0.8001  0.8001  0.8001
0.8001  0.8001  0.8001  0.8001  0.8001  0.8001  0.8001  0.8001
0.8001  0.8001  0.8001  0.8001  0.8001  0.8001  0.8001  0.8001
0.8001  0.8001  0.8001  0.8001  0.8001  0.8001  0.8001  0.8001
0.8001  0.8001
```



```
B-rate
0.8490  0.8491  0.8492  0.8493  0.8494  0.8494  0.8495  0.8495
 0.8496  0.8496  0.8496  0.8497  0.8497  0.8497  0.8497  0.8498
0.8498  0.8498  0.8498  0.8498  0.8498  0.8498  0.8499  0.8499
0.8499  0.8499  0.8499  0.8499  0.8499  0.8499  0.8499  0.8499
0.8499  0.8499  0.8499  0.8499  0.8500  0.8500  0.8500  0.8500
0.8500  0.8500  0.8500  0.8500  0.8500  0.8500  0.8500  0.8500
0.8500  0.8500
```



```
C-Rate
0.8995  0.8995  0.8996  0.8996  0.8997  0.8997  0.8997  0.8998
0.8998  0.8998  0.8998  0.8998  0.8998  0.8998  0.8999  0.8999
0.8999  0.8999  0.8999  0.8999  0.8999  0.8999  0.8999  0.8999
0.8999  0.8999  0.8999  0.8999  0.8999  0.9000  0.9000  0.9000
0.9000  0.9000  0.9000  0.9000  0.9000  0.9000  0.9000  0.9000
 0.9000  0.9000  0.9000  0.9000  0.9000  0.9000  0.9000  0.9000
0.9000  0.9000
```

5. Reputation-based Trust Formulation

The Sporas formula updates the node reputation to current status. The reputation status value of each node is stored by its neighboring nodes. Therefore, each node maintains a table and stores the reputation status of its neighboring node or nodes. If any node gets malicious, then the connected nodes change the reputation value in their table. If the value of a node drops below the threshold, then the node will be declared as malicious. The declared malicious node will be disconnected from the network.

The reputation of a node is calculated average of two methods. First, reputation value is calculated through Sporas formula using equation (1) basing on opinion poll. Second, reputation of a neighboring node is calculated using the ratio of the number of packets sent to the node (S_m) to the number of packets forwarded (F_n) by the node.

$$R_{nm} = \frac{F_n}{S_m} \tag{3}$$

The ratio $F_n / S_m \le 1$ and $n / m \le 1$ must be true all time. Unlike in Sporas formula, whenever a node is added to the network, it is given a reputation value equals to 1, means the reputation value is 100%. The reputation R must be calculated in fixed intervals. After few intervals, the average reputation value will be generated (includes the initial value). The average reputation value R_{av} should not fall below the threshold T. Therefore, it follows that $R_{av} > T$, otherwise the node is treated as malicious. Consider an arbitrary variable x that has a maximum value 1 and minimum 0. The current reputation value of a node is calculated as:

$$R_r = ((1-x).R_p + x.R_c) \le 1 \tag{4}$$

where, R_r is the calculated reputation calculated, R_p is the previous reputation value, and R_c is the current reputation value. The value of x will be above 0.5 and closely the threshold value (0.95). The new updates must be small enough to be comparable with Sporas formula. The equation (4) will be compared to the equation (1), where the reputation in both cases provides the current trust status of the node. The average of these two values will provide best possible trust value R. Therefore,

ISBN: 978-1-61208-144-1

$$R = (R_i + R_r)/2 \le 1 \qquad (5)$$

If the node A observes that the node D is suspicious, Figure 1, then it broadcasts to its neighbor nodes C and B. The node C and node B calculates the trust value of node D using the broadcasted value and determines the node A's claim of suspiciousness. The node B calculates the trust of node D as below:

$$T_{ND} = R.R_{BA} + (1 - R_{BA}).R_{BD} \qquad (6)$$

where

T_{ND} is the new trust value of D at B

R is the trust value received from A

R_{BA} is the trust value of B on A

R_{BD} is the trust value of B on D

If A broadcasts that D is suspicious, B should not believe immediately. It should use the trust on A and trust on D and calculate the combined trust. If the calculated value is below the threshold then B believes that D is suspicious otherwise it broadcasts that D is not suspicious. Similarly the node C calculates its own trust on D.

Discussion of Results

We assume that the current trust value of node D is known by nodes A, B, and C. Suppose the current trust values of node D at nodes A, B, and C are 0.8, 0.85, and 0.92 respectively. Let *x*=0.8, and let R_r, R, and T_{ND} be given by equations (5) and (6). The value of R_r, R, and T_{ND} with respect to node C and node B decides that either node D will be trusted or not.

It is a known fact that sensors have limited resources including battery, computational, and communication resources. Therefore, it is suggested to use either Sporas formula or reputation-based trust model. It is further suggested using average of both formulas depending upon the sensitivity of the problem. If the data sensitivity is low and data need to be transferred securely then use either one of the formulas.

6. SIMULATIONS ON REPUTATION-BASED TRUST CALCULATION

The simulation uses the Dynamic Source Routing (DSR) protocol of wireless sensor network. The simulation is written in the Java language. The idea of the simulation is based on the popular simulation software NS-2 (Network Simulator 2). We created a 500 x 500 field and randomly distributed 20, 50 and 100 nodes in the network. The simulation detected all misbehaving (nodes which did not forward data properly) nodes such as a sink hole and selective forwarding nodes. We calculated the trust ratio using promiscuous mode overhearing the packets forwarded by the successive node. The node swaps from promiscuous mode to normal mode as soon as it overhears the packet to save battery life. If it does not overhear packet for a given time frame, it automatically comes back to normal mode. The trust ratio was calculated using the number of packets received by successive node and transferred from it.

The sample simulations are given in Figure 3. We assumed node 2, node 3, and node 4 are neighbors of node 1. These four nodes transfer the data from the same node and calculate the reputation value. If the reputation value of node 1 drops below the threshold, then node 1 verifies the data from its neighboring nodes (nodes 2 – 4). For example, node 2 is a neighbor of node 1 and sent 50 packets and the successive node forwards 50 packets. That is 90% success. Similarly, the calculations follow from other nodes. Each node calculates its trustworthiness using equation (6) and send to node 1. The node 1 decides the trustworthiness of its successive after receiving data from its neighboring nodes.

7. TRUST CLUSTER APPROACH

Trust clusters are very useful in the collaborative approach. The nodes within communication distance forms a cluster, and conduct collaborative activity in calculating the trust of any successive node. For example, if n_j is a neighbor of node n_i then we represent the neighbors $(n_i, n_j) = true$. Suppose, if n_i has more than one neighbor then we write

$$(n_i, n_j)_{\bar{\lambda} j} = true. \qquad (7)$$

$\bar{\lambda} j$ are the j nodes which are within the communication distance of node i. That is, i has the collaborative relation to the $\bar{\lambda} j$ nodes. Similarly, we form the neighboring nodes to each node in the network. Suppose $\zeta_{i,j}$ is the trust factor of each of j node close i^{th} node, then most dependable node in the neighborhood of a node i is the highest reputation value calculated through equation (5).

It is always necessary to keep track of the most trusted nodes within the communication distance and most inferior nodes within the communication distance. The inferior nodes will be eliminated to calculate the trust factor and if the trust factor of any inferior node is below the threshold, then all neighbors must discord the suspected node from the network. The inferior node is denoted as

$$\varsigma_{\inf} = (n_i, n_j)_{\lambda_j < threshold} \qquad (8)$$

Therefore, the node i must depend upon the trusted neighbor nodes for the future path selection.

8. CONCLUSIONS

Sensors are organized in an open environment and injecting of bad nodes to corrupt the transmission is possible. Therefore, a secure transmission model is required to avoid the transmission through corrupted node. The traditional protocols use exchange and distribute the keys for trust evidence. The resource limitations in sensor networks obstruct the use of traditional tools including cryptographic models and intrusion detection packages for secure communications. Encryption, intrusion detection models, and authentication techniques limits the performance of nodes in WSNs. Encryption and authentication cannot prevent the malicious or misbehavior of the sensor nodes. After reviewing relevant sensor network models, we believed that proposed collaborative trust-based model will overcome the shortcomings in the current models.

In the proposed collaborative model, each node is each node is updated through ratings. The ratings are provided by the nodes transfer the packets through that node. The update of node ratings is done through Sporas formula. If the node rate is below the threshold, the previous node uses the cooperative effort through neighboring nodes. By using the cooperative and collaborative effort, we eliminate the suspicious node from the communication path. Preliminary results are presented using the rating of a node through Sporas formula.

The future work includes the agent-based trust model. Each cluster has an agent and agent relieves the computations of the nodes. All decisions will be taken at the agent.

ACKNOWLEDGEMENT

The research work was supported by the ONR with award No. N00014-08-1-0856. The first author wishes to express appreciation to Dr. Connie Walton, Grambling State University and Dr. S. S. Iyengar, LSU Baton Rouge for their continuous support.

REFERENCES

[1] Reddy, Y. and Selmic, R., “Secure Packet Transfer in Wireless Sensor Networks – A Trust-based Approach”, IARIA- ICN 2011, January 23-28, 2011 - St. Maarten, pp. 218-223.

[2] Liang, Z. and Shi, W., “PET: A Personalized Trust Model with Reputation and Risk Evaluation for P2P Resource Sharing”, 38 Hawaii Int. conf. on Systems Sciences, 2005, pp. 201-210.

[3] Aberer, A. and Despotovic, Z., “Managing trust in a Peer-2-Peer information system”, 10th International Conference on Information and Knowledge management, 2001, pp. 310-317.

[4] Generowal, S. and Srivastava, M., “Reputation-based Framework for high Integrity Sensor Networks”, 2nd ACM workshop on Security of ad hoc and sensor networks, 2004, pp. 1-36.

[5] Marmol, F. and Perez, M., “Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique”, Networking and Electronic Commerce Research Conference (NAEC), 2008, pp. 1-16.

[6] Momani, M. and Challa, S., “Survey of Trust Models in different Network Domains”, International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Vol.1, No.3, September 2010, pp. 1-19.

[7] Chen, H., “Task-based Trust Management for Wireless Sensor Networks”, International Journal of Security and its applications, vol 3, 2009 (last accessed: May 24, 2011), URL: http://earticle.net/article.aspx?sn=105974)

[8] Chen, H., Wu, H., Hu, J. and Gao, C., “Event-based Trust Framework Model in Wireless Sensor Networks”, IEEE International Conference on Networking, Agriculture, and Storage, 2008, pp. 359-364.

[9] Chen, H., Wu, H., Hu, J. and Gao, C., “Agent-Based Trust Management Model for Wireless Sensor Networks”, 2008 International Conference on Multimedia and Ubiquitous Engineering, 2008, pp. 150-154.

[10] Boukerche, A. and Xu, L., “An agent-based trust and reputation management scheme for wireless sensor networks”, Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE, 28 Nov.-2 Dec. 2005, pp.1857-1861.

[11] Chen, H., Wu, H., Zhou, X. and Gao, C., “Agent-based Trust Model in Wireless Sensor Networks”, Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007, pp. 119-124.

[12] Zhang, W., Das, S. K., and Liu, Y., “A Trust Based Framework for Secure Data Aggregation in Wireless Sensor Networks”, 3rd annual IEEE communications on sensor and ad hoc communications and networks (SECON 06), 2006, pp. 60-69.

[13] Momani, M. and Challa, S., "Probabilistic Modelling and Recursive Bayesian Estimation of Trust in Wireless Sensor Networks," submitted to *Ad Hoc Networks*, 2007, pp. 381-403.

[14] Marsh, S. P., “Formalising Trust as a Computational Concept,” PhD Thesis, University of Stirling, 1994.

[15] Zacharia, G., Moukas, A., and Maes, P., “Collaborative Reputation Mechanisms for Electronic Marketplaces”, Decision Support Systems, Volume 29, Issue 4, December 2000, pp. 7-29.

[16] Josang, A. and Ismail, R., “The Beta Reputation System”, 15th Bled Electronic Commerce Conference, 2002, pp. 1-14.

Graphs

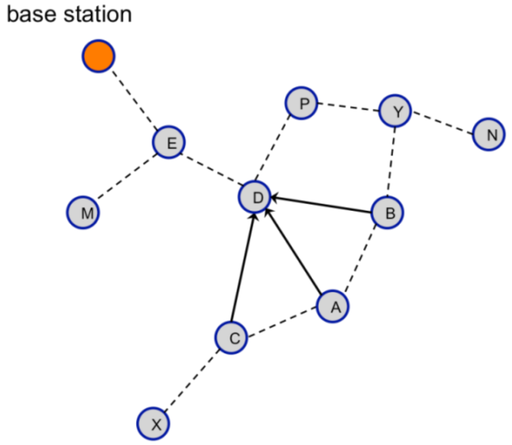


Figure 1. Wireless sensor network communication topology.

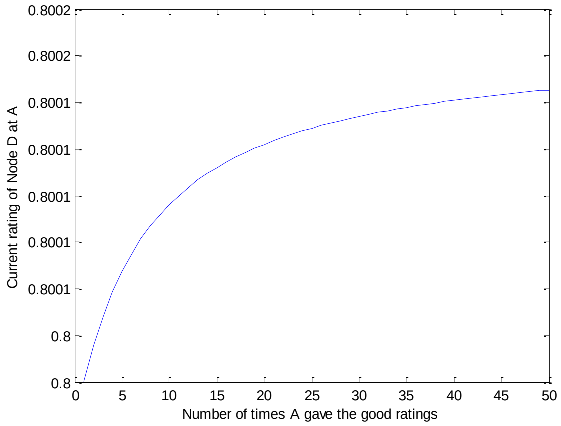
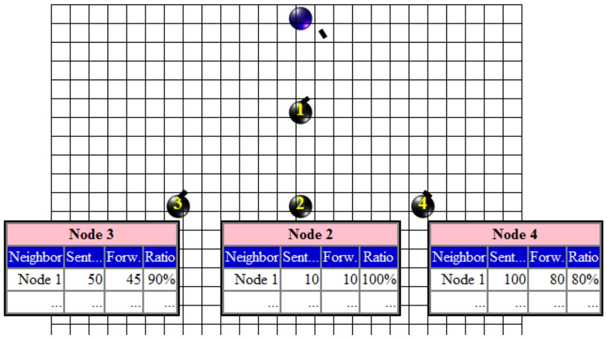


Figure 2. The rating of node A on node D.



Node 3			
Neighbor	Sent...	Forw.	Ratio
Node 1	50	45	90%
...	...	...	...

Node 2			
Neighbor	Sent...	Forw.	Ratio
Node 1	10	10	100%
...	...	...	...

Node 4			
Neighbor	Sent...	Forw.	Ratio
Node 1	100	80	80%
...	...	...	...

Figure 3: Simulation Of Wireless Sensor Networks using DSR Protocol

ISBN: 978-1-61208-144-1

Agent-based Trust Calculation in Wireless Sensor Networks

Yenumula B. Reddy
Grambling State University
Grambling, LA 71245, USA
ybreddy@gram.edu

Rastko Selmic
Louisiana Tech University
Ruston, LA 71270, USA
rselmic@latech.edu

***Abstract* - Cooperation in wireless sensor networks to detect the malicious node without any infrastructure is a recent trend in research. The current models need more storage, computation, security tools, and communication requirements. They fail in wireless sensor networks due to limitation of resources. Trust-based approach does not need high-end resource requirement. The proposed agent-based approach eliminates the computations in the sensor nodes with appropriate trust factor. The proposed approach uses an agent-based collaborative concept to ensure the trust in the successive node in the path. The proposed agent-based framework uses reputation of neighboring nodes as part of trust calculation in its successive node. The simulations were presented to calculate the trust of a node.**

Keywords: agent-based approach, packet transfer, wireless sensor networks, protocols, trust-based approach, resource.

I. INTRODUCTION

Sensors are small in size, limited computational power, and capabilities. Wireless sensor networks are based on these small form-factor nodes transmitting the collected information to the base station. Since safe transmission of information is important, the path of transmission must be trustworthy. Therefore, each node must trust the successive node in the path. If any node in the path is suspicious, the decision node must calculate the alternative path.

There are varieties of methods to calculate the trust of a successive node. The methods include the reputation-based trust management, event-based trust management, collaborative trust management, and agent-based trust management. In reputation-based trust management, the node stores the number of packets transfer from the node and calculate the success rate of packets transferred from its successive node. In the event-based trust management system, the trust rate is calculated at particular or specific time events or periodically. In collaborative models, the business models are used to calculate the trust similar to product trust management. In agent-based trust management systems, an agent node is introduced to store the packet transfer information from a cluster of nodes within communication distance. The agent-based systems relieve the most of the processing time of nodes and the nodes concentrate on transfer of information. Trust-based systems will help to detect the malicious nodes and eliminate them from the communication path.

A trusted node must transmit the minimum acceptable number of packets. The minimum acceptable number is called threshold. The threshold is used to rate the node. The ratings will be updated and maintained using Sporas formula [7] or Molina’s fuzzy reputation model [5] or proposed agent-based model. The proposed model reduces the overheads on sensor nodes and helps to improve the life time expectancy and efficiency.

Figure 1 show the WSN with nodes, neighbor nodes, and an agent to collect and process trust information. The agent’s responsibility is to collect the node ratings update the trust of each node within communication distance of successive node in the path. The agent also provides the level of trust and recommends alternative path if the trust is below the threshold value.

The remaining part of the paper discusses the related work, reputation based trust, agent-based trust calculation. The reputation based trust model uses Sporas formula and Molina’s fuzzy model and comparison of these models to update the rates. Finally, the paper presents concluding remarks and future research.

II. RELATED WORK

Trust management is not a new concept in the electronic market. Reputation and trust are the basics of product sales. Establishing trust on a product manufacture industry and reputation of a product is the source of sales. Similarly, establishing trust on a node transferring the packets and reputation of the node is very important to keep the sensor node on data transfer path. Trust calculation and update the node ratings uses reputation-based trust calculation [1, 4], event-based trust management [3], and agent-based trust management [7-9]. Repeated games help to detect the trustworthiness of a node in the path [1].

Ganeriwal et al. [2] discussed the reputation-based framework for high integrity sensor networks. The model evaluates the trustworthiness of the nodes and various type of misbehavior of nodes in the network. The model uses the Bayesian formulation and updates the trust with direct and indirect trust calculations.

Trust is not consistent. It varies from time to time and event to event. In sensor networks, a series of events happens. Data collection, data routing, location report, identifying neighbor, reorganizing the network, and time synchronization are very common. In event-based systems [2], the behavior of sensors and collection of trust rating from neighbor nodes is done through agents. The agent decides the trustworthiness of sensor and path reestablishment.

The agent-based system [7-9] uses various methods of sensor node ratings and calculation of trust of nodes. In agent-based models, an agent is created with a set of nodes within the communication distance. The agent is responsible to calculate the trust and reputation of the nodes using various formulas.

Collaborative reputation in an electronic market [3] uses the Sporas formula to calculate the ratings of a node on Web. The ratings will conclude the trust in WSN. Bio-inspired technique based on ant colony system. The most worthy path is detected by using the pheromone traces deposited by ants.

Momani et al. [10] proposed the secure data aggregation scheme to detect the inside attack (within networks) and trustworthiness of a node in the WSN. Further, trust establishment in ad hoc network using distributed environment was studied in [12].

Contribution: Trust ratings with Sporas formula and fuzzy reputations of Molina's formula were derived and compared. The two methods used to calculate the trust of a node. It is concluded that the learning rate and most recent trust rate helps in detecting the malicious node quickly. Further, the agent in each cluster minimizes the computational overhead of the nodes. The simulations were presented to illustrate the theoretical analysis.

III. TRUST AND REPUTATION

The reputation-based models use the rate of a number of packets received to transfer by a node [1]. The event-based models calculate the trust on the rate of transfer of packets at any particular event [2]. Further, business (collaborative) models are used to calculate the trust of a node depending upon the rating by neighboring nodes [3]. All models were used to calculate the trust and detect the malicious node, so that they can avoid the malicious node from the data transfer path. These calculations show that trust is calculated on the behavior of a node in the data transfer path.

Molina et al. used the fuzzy reputation to calculate the trust of a node [5, 6]. The trust depends upon the reputation of a node R_{i-1} at the time $i-1$, current rating C_i and remembrance weight ω. The maximum value of remembrance is 1. Therefore, $0 \le \omega \le 1$. The current reputation is calculated as [5]:

$$R_i = \frac{R_{i-1}.\omega + C_i.(2-\omega)}{2} \quad (1)$$

If $\omega = 0$ then $R_i = C_i$. If the node does not remember the previous reputation, then current rating is the reputation value. It shows that a new node entering into network does not have previous value. If $\omega = 1$ then the new reputation is equal to average of previous reputation and current rating. The maximum value of ω provides the excellent reputation and more trustworthy. Therefore, the equation (1) becomes

$$R_i = \frac{R_{i-1} + C_i.}{2} \quad (2)$$

The equation (2) shows that if a node is added with the best possible rating, it should not be given more than half of reputation. The reputation must be established.

The reputation of a node is updated with current ratings. The current ratings are obtained using the following Sporas formula [7].

$$R_i = R_{i-1} + \frac{1}{\theta}\phi(R_{i-1})(C_i - R_{i-1}) \quad (3)$$

$$\phi(R_{i-1}) = 1 - \frac{1}{1+e^{-(R_{i-1}-D)/\sigma}} \quad (4)$$

where:

θ - effective number of ratings taken into account $(\theta > 1)$. The change in rating should not be very large.

ϕ - helps to slow down the incremental change

C_i - represents the rating given by the node i

D - range or maximum reputation value

σ - the acceleration factor to keep the ϕ above certain value (> threshold).

If the node compromises, the rating will be smaller and $(C_i - R_{i-1})$ become negative. Therefore the current reputation slowly crosses below threshold and node declared as malicious.

The equations (1) and (3) calculate the new reputation of a node. Substituting equation (1) in (3), we obtain.

$$\frac{R_{i-1}.\omega + C_i.(2-\omega)}{2} = R_{i-1} + \frac{1}{\theta}\phi(R_{i-1})(C_i - R_{i-1}) \quad (5)$$

Assume the remembrance weight ω =1 or ω =0 the equation (5) simplifies

$$C_i = R_{i-1} \quad (6)$$

The equation (6) shows that if the remembrance ω =1 or ω =0 the ratings given by a node i is equal to reputation of the node. That is, a long term excellent reputation node and recent added good node assumes to be trustworthy.

Further, in equation (1), if the remembrance ω=1, then current reputation is average of previous reputation and current ratings. Figure 2a shows the relation between reputation of a node and current reputation. In normal conditions, the current reputation is proportional to previous reputation.

Figure 2b is drawn for the remembering weights $\omega = 0,\ 0.7,\ 1.0$. Once the system get updated continuously, the node rate constantly increases (stabilizes). If the reputation is random (reputation may be low or high) and ratings are increasing or decreasing, the node is not trustworthy. The node drops the packets randomly. The Figure 2c and Figure 2d shows that if the nodes are dropping packets randomly, the increasing reputation is better than decreasing reputation.

In the agent based systems, it is recommended to use the Sporas formula to update the ratings, so that the fuzzy reputation formula of equation (1) provides better results. The reliability of the nodes in WSN is temporary. The continuous update of ratings is required in the WSN.

IV. AGENT-BASED APPROACH

Agent-based trust approach is similar to cluster-based approach or watchdog approach [5, 7, 9]. The cluster forms with the nodes that are within communicating distance. Each cluster has an agent to collect the reputation of nodes. The reputation of a node includes two factors.

- Trust of each node in the cluster transmitting the packets through same node and must be within communicating distance.
- Trust of a node (constant and less than 1) to its neighboring node(s).

The agent keeps the above information of each node within communicating distance and calculates the trust of a node in the transmitting path. The trust value decides the trust of node in the communication path. Therefore, the trust depends upon the direct observations of a node plus the indirect observations received from its neighboring nodes. The reputation of a node is calculated in two ways.

Case 1: From the Figure 1, the reputation of a node D at node A is a sum of the observations of node A, node C with respect node A, and node B with respect to node A. The reputation of node D at node A is given by

$$R_{A,D} = \alpha.R_{A,D} + \beta.R_{C,D} + \gamma.R_{B,D} \quad (7)$$

and $$\alpha + \beta + \gamma = 1 \quad (8)$$

where

$R_{A,D}$ reputation of node D at node A

$R_{C,D}$ reputation of node D at node C

$R_{B,D}$ reputation of node D at node B

The nodes C and B are neighbors of node A. The direct reputations are at decision node and indirect reputations are from its neighboring nodes. Initially, the constant factor at decision node carries higher value than other nodes. The values of β and γ are based on the trust of node A with respect to nodes C and B. Figure 3a shows that the higher value of alpha lower the confidence of a node that was put in trust test. If the value of β and γ are larger, then the indirect observations provide better results. That is, the neighbor nodes receive more confidence on the successive node with respective to the testing node (node A is a testing node in the current case).

Therefore, it is better to adjust the alpha value at lower level (<0.5). Figure 3b shows the collaborative trust calculation at Node A as trust value decreases. Collaborative effort helps and confirms the trust status. In the current problem (Figure 3a and 3b), it is clearly shown that, the node A to D has communication problem and D is not a malicious node. Furthermore, node A can confirm from node B and node C the confidence or reputation of node D using their original trust values which are stored at the agent.

In agent-based systems, the agent has the trust and reputation values of all nodes. The agent also has the level of belief on its neighbors. The level of belief is the multiplication factor (β or γ) that helps to calculate its belief factor on a specific node. The trust at any node is calculated using the equation (7). Further, the agent-based system eliminates the computations required at each node and saves the energy of nodes. Saving the energy increases the life of sensor nodes.

Case 2: The trust of node D with respect to node A $(R_{A,D})$ is calculated using the trust of node D at B with respect to node A and trust of node D at C with respect to node A.

(a) Trust of node D at node B with respect to node A $(R_{B_A,D})$ is the sum of the trust of node B on node D and trust of node B on node A (R_{BA}) :

$$R_{B_A,D} = R_{A,D}.R_{BA} + (1 - R_{BA})R_{B,D} \qquad (9)$$

(b) Trust of node D at node C with respect to node A $(R_{C_A,D})$ is the sum of the trust of node C on node D and trust of node B on node A (R_{CA}) :

$$R_{C_A,D} = R_{A,D}.R_{CA} + (1 - R_{CA})R_{C,D} \qquad (10)$$

Find the average of trust of node A on D, trust of node B on node D with respect A, and trust of node C on node D with respect A.

$$R_{A,D} = (R_{A,D} + R_{B_A,D} + R_{C_A,D}) / 3 \qquad (11)$$

Figure 4a shows the slow decrease of trust calculated through equations (9) to (11). The confidence factor helps to confirm the successive node status. The Figure 4a is drawn with higher reputation of neighbor nodes and trust of node A on node D is decreasing. Figure 4b is drawn for higher reputation of node D at node A (above the threshold value) and lower reputation of nodes B and C on D. The results show that the lower reputation of node D at neighboring nodes effects the decision at node A.

The equations (7) and (11) approximately produce the same result. The results show that if the node D is malicious and temporarily produces better reputation at A, the collaborative effort will give warning to drop the node from the communication path.

V. CONCLUSIONS

Trust-based packet transfer has been taken significant importance in recent years. The secure transfer of information with low cost is still a debatable problem in WSN. In this paper, we first presented the fuzzy rating models and Sporas formula for node rating. An agent-based approach was introduced to calculate the trust using the collaborative approach. The ratings of a node and its neighbors with respective to the node help for better decision on trust calculation of successive node in the path. A similar approach was used to lower the burden of computational work on the node. Lowering the computational work at node increases the life of sensor node.

The future research includes the event-based trust calculation. The event-based trust is recently introduced, and very little work was done in this line. Event-based trust models depend upon the specific events in the surroundings of a sensor node. It will be easier to detect the malicious node in the communication path using the data of specific events in the surroundings of a node.

ACKNOWLEDGEMENT

The research work was supported by the ONR with award No. N00014-08-1-0856. The first author wishes to express appreciation to Dr. Connie Walton, Grambling State University and Dr. S. S. Iyengar, LSU Baton Rouge for their continuous support.

REFERENCES

[1] Y. B. Reddy and Rastko Selmic., “Secure Packet Transfer in Wireless Sensor Networks – A Trust-based Approach”, IARIA- ICN 2011, January 23-28, 2011 - St. Maarten, pp. 218-223.

[2] Chen, H., Wu, H., Hu, J., and Gao, C., “Event-based Trust Framework Model in Wireless Sensor Networks”, International Conference on Networking, Architecture, and Storage, 2008, pp. 359-364.

[3] Zacharia, G., Moukas, A, and Mae, P., “Collaborative Reputation Mechanisms for Electronic Marketplaces”, Decision Support Systems, Vol. 29, Issue 4, December 2000, pp. 1-7.

[4] Ganeriwal, S., and Srivastava, M. B., “Reputation-based Framework for High Integrity Sensor Networks”, Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks (SASN ’04), 2004, pp. 66-77.

[5] Carbo, J., Molina, J.M., and Davila, J., “Trust Management through Fuzzy Reputation”,

International Journal of Cooperative Information Systems", Vol. 12, Issue 1, 2003, pp. 135-155.

[6] Carbo, J., Molina, J.M., and Davila, J., "Comparing Predictions of Sporas vs. a Fuzzy Reputation System", 3rd International Conference on Fuzzy Sets and Fuzzy Systems, 2002 (last accessed on May 24, 2011: www.wseas.us/e-library/conferences/switzerland2002/papers/456.pdf

[7] Chen, H., Wu, H., Hu, J., and Gao, C., "Agent-based Trust Management Model for Wireless Sensor Networks", International Conference on Multimedia and Ubiquitous Engineering, 2008(last accessed: May 24, 2011), URL: http://earticle.net/article.aspx?sn=105974)

[8] Chen, H., Wu, H., Hu, J., and Gao, C., "Agent-based Trust Model in Wireless Sensor Networks., "Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing", 2007, pp.119-124.

[9] Boukerche, A., and Li, X., "An Agent-based Trust and Reputation Management Scheme for Wireless Sensor Networks", IEEE GLOBECOMM, 2005.2nd ACM workshop on Security of ad hoc and sensor networks (SASN '04), October 2004, pp 66-77,.

[10] Momani, F. G., and Perez, G. M., "Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique", NAEC 2008, pp. 1-16.

[11] Momani, M. and Challa, S., "Probabilistic Modelling and Recursive Bayesian Estimation of Trust in Wireless Sensor Networks," submitted to Ad Hoc Networks, 2007, pp. 381-403.

[12] Aivaloglou, E., Gritzalis, S., and Skianis, C., "Trust Establishment in ad hoc and Sensor Networks", Lecture notes in computer science, 2006, vol. 4347, pp. 179-194.

Graphs

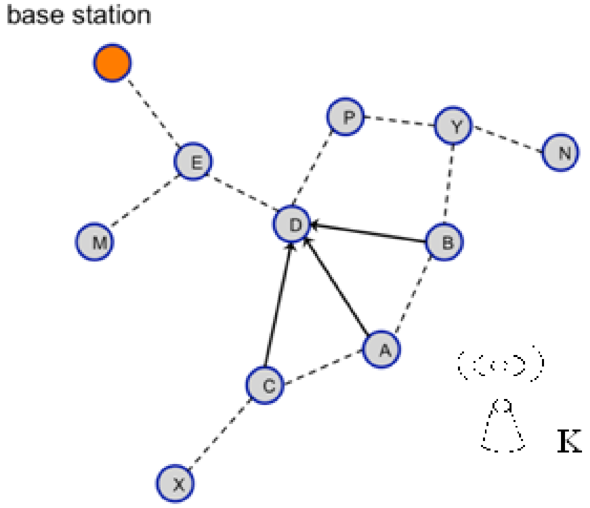


Figure 1: Wireless sensor network communication topology.

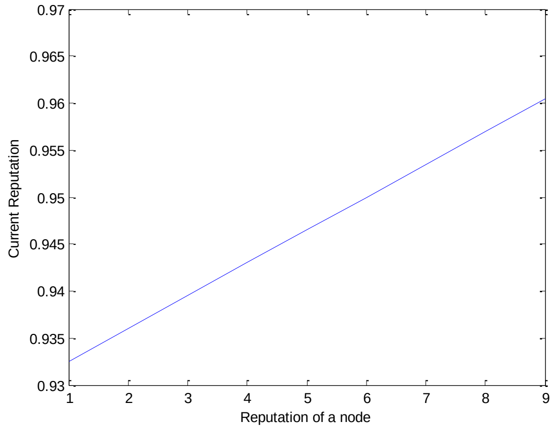


Figure 2a: Relation between the reputation of a node and current reputation

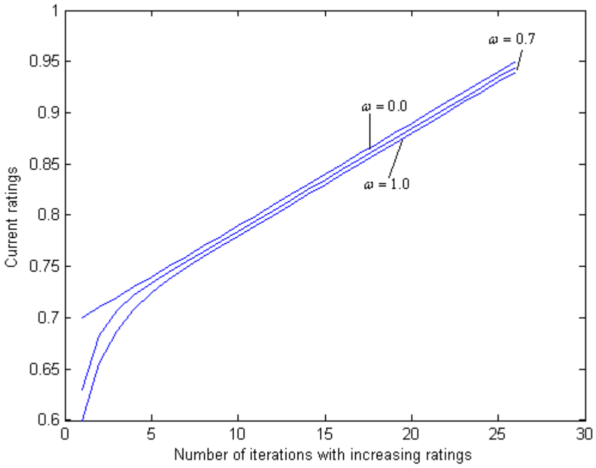


Figure 2b: Relation between the reputation of a node and current reputation

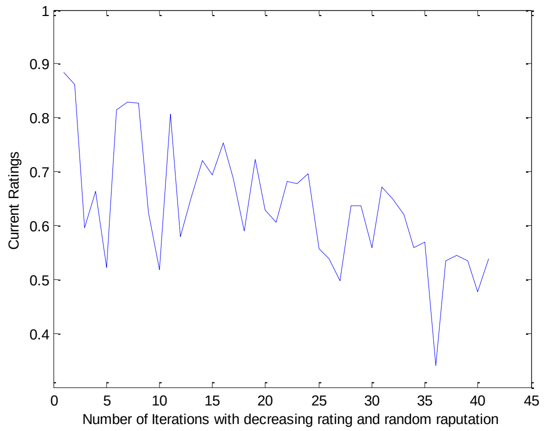


Figure 2c: Relation between the reputation of a node and current reputation

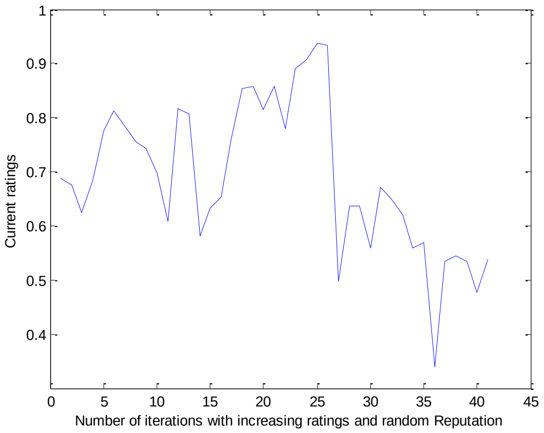


Figure 2d: Relation between the reputation of a node and current reputation

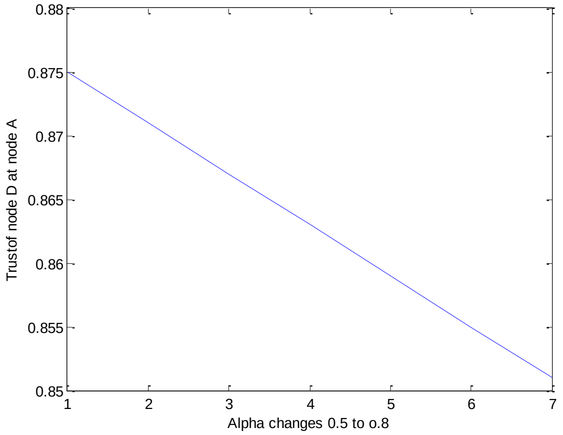


Figure 3a: Trust of node D at A with collaborative effort

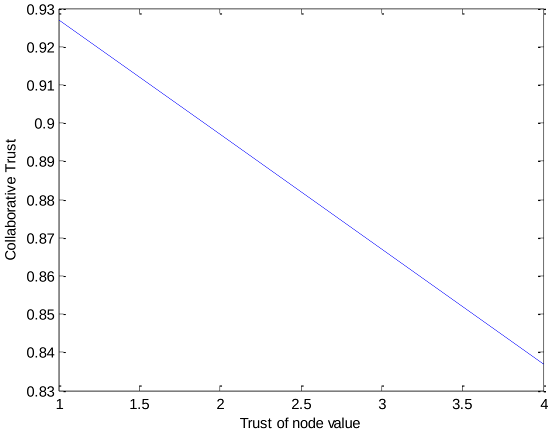


Figure 3b: Trust of node D at A with collaborative effort

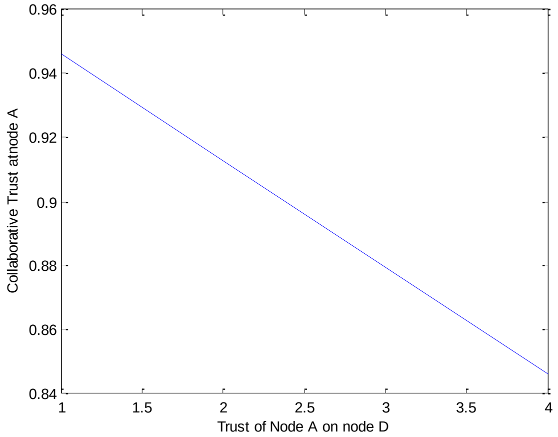


Figure 4a: Trust of node D at A with collaborative effort for Case 2.

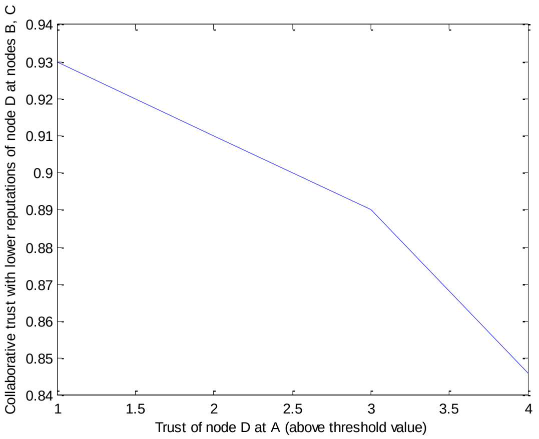


Figure 4b: Trust of node D at A with collaborative effort with lower confidence at nodes B and C (for Case 2).

ISBN: 978-1-61208-144-1

A Pairing-Free ID-based One-Pass Authenticated Key Establishment Protocol for Wireless Sensor Networks

Rehana Yasmin, Eike Ritter
School of Computer Science, University of Birmingham
Birmingham, B15 2TT, UK
Email: {R.Yasmin,E.Ritter}@cs.bham.ac.uk

Guilin Wang
School of Computer Science, University of Wollongong
Wollongong, NSW 2522, Australia
Email: guilin@uow.edu.au

***Abstract*—Due to resource constraints and unique features of wireless sensor networks (WSNs), designing a key establishment protocol is much harder for WSNs than for traditional wired and wireless counterparts. In this paper, we propose a new efficient and secure ID-based one-pass authenticated key establishment protocol between an outside user and a sensor node. The proposed protocol does not require sensor nodes to compute any expensive pairing function. Moreover, it imposes very light computational and communication overheads and also provides scalability. We analyze security and efficiency of the proposed protocol by comparing firstly the session key establishment protocols for WSNs and secondly the existing ID-based one-pass key establishment protocols. The comparison shows that the proposed protocol is the most secure and efficient one for WSNs applications providing both security features of user authentication and session key establishment.**

***Keywords*-Wireless Sensor Networks; Security; ID-based One-Pass Key Establishment; User Authentication;**

I. Introduction

Recent advances in embedded technologies, as well as wireless communications, have broadened the prospects for many applications of wireless sensor networks (WSNs), for instance, environmental monitoring, ocean reading and many military applications [1]. However, the vulnerability of wireless communication and the ad-hoc nature of deployment open the door for a wide variety of malicious attacks, making security a key concern for these applications. On the other hand, the resource constrained nature of sensor nodes, i.e., limited *power*, *computing* and *storage* resources, poses a need for highly efficient security solutions. This restriction has significantly impacted the field of application security. For such applications, the efficiency of a security scheme is as important as its security. Any security scheme which is computationally expensive, no matter how secure it is, does not suit resource constrained sensor nodes.

To protect the communication in WSNs, one security requirement is the ability to encrypt and decrypt confidential data entailing the establishment of a session key. An authenticated **session key establishment** protocol provides the communicating parties with a secret and authentic shared session key which is used for the encryption and decryption of data. The session key establishment protocol is particularly important for those applications of WSNs which frequently exchange confidential data through insecure channels, for instance, environmental monitoring and ocean reading. In these applications, the data collected by the sensor nodes is useful for many research and business purposes. Different research organizations and businesses pay money to the deployment agencies of large scale sensor networks and obtain data from them. Thus, the data collected within the network is valuable and confidential in these applications. Since the data is available only to “authorized” users who have paid for the data, **user authentication** is another security requirement for such applications. These authorized users, after successful authentication, issue queries to the sensor nodes to access data of their interest. Therefore, a secure mechanism becomes highly desirable in these applications that allows sensor nodes to establish a session key with the users (to encrypt and decrypt confidential data) while facilitating all the necessary authentications (to know who their counterparts are). On the other hand, designing a secure and efficient security protocol for resource constrained sensor nodes is a challenging task.

In this paper, we propose an efficient and secure ID-based one-pass session key establishment protocol between an outside user and a sensor node which combines user authentication with session key establishment. ID-based cryptography [2] replaces a user’s public key with his unique public identifier (ID), such as email address. The corresponding private key is generated by a private key generator (PKG), a trusted third party. ID-based cryptography removes the need for certificate transmission and verification to obtain the public key, and hence reduces the transmission and the processing costs of the security schemes.

A. One-Pass Key Establishment

High computation and communication cost of secure two-pass key establishment protocols makes them expensive for low-power WSN applications (where low computation and communication cost is critical), for instance, the authenticated DiffieHellman protocol named as Station-to-Station protocol [3]. To satisfy the resource constraints of sensor nodes, a session key establishment protocol with high security and a minimum amount of computation and number of passes is required. A secure one-pass key establishment protocol is an attractive alternative for them. In a one-pass key establishment protocol only one message transmission is required for the establishment of key, i.e., only the sender (initiator) of the protocol generates an ephemeral private key and transmits its public part, called the ephemeral public key, to the receiver (responder). Both parties then compute a shared session key using their own private keys and ephemeral keys. In one-pass key establishment protocols, the reduced number of exchanged messages lessens the transmission and processing costs because only one message is transmitted and processed.

Besides the reduced cost, another advantage of a one-pass key establishment protocol is its use for off-line communications, explained as follows: The sender computes its shared session key, encrypts the message *m* (any confidential message) using the computed session key and sends both the ephemeral public key and the ciphertext of *m* to the receiver. The receiver can compute the same shared key any time using the sender's ephemeral public key and decrypt the message. The receiver needs not to be necessarily on-line. This feature is particularly useful for applications where only one entity is on-line, for instance, email. This feature can also be utilized in a WSN environment where the only message sent by the user for key establishment can be combined with the encrypted user query to provide query privacy. The details are discussed in Section III.D.1.

Contribution. The main contribution of this paper is a new secure and efficient ID-based one-pass key establishment protocol for WSNs. To the best of our knowledge, this is the first ID-based one-pass key establishment protocol which does not require any pairing computation. Lack of pairing operation makes our scheme computationally efficient and, hence, suitable for resource constrained sensor nodes. Scalability is another attractive feature of the proposed protocol which is required for WSN environment. Other than performance, the provable security is also an aspect of the proposed protocol. Although the details of formal security analysis including security model and security proof are omitted in this paper due to space limitations, they will be a part of the extended version of this paper.

Organization. Section II presents an overview of the related work. Section III describes the proposed scheme in detail. Section IV and Section V give the security analysis and the performance evaluation of the proposed protocol, respectively. Finally, a brief conclusion is given in Section VI.

II. Related Work

A. Session Key Establishment in Wireless Sensor Networks

This section briefly reviews the work related to the session key establishment in WSNs. A public key cryptography based hybrid authenticated key establishment protocol between a sensor node and a security manager is proposed by Huang et al. [4]. Their protocol exploits the difference in capabilities between the sensor nodes and the security manager. Like an outside user, a security manager is a powerful device (compared to a sensor node) which establishes a session key with a sensor node for subsequent use. In the beginning of the protocol, both parties exchange their certificates signed by a certification authority to extract the public keys of each other. However, the knowledge of the corresponding private keys is only proved after the complete run of the protocol on both sides. An adversary can exploit this fact and repeat this protocol with the sensor node by replaying a valid certificate, resulting into Denial of Service (DoS) attack. Before a sensor node detects the replayed certificate, it would have performed expensive computations and communications wasting its resources, particularly battery power. Later on Tian et al. [5] detected another serious security attack against Huang et al.'s protocol. They showed in [5] that a security manager (user in our case) easily learns the long-term private key of a sensor node after having one normal run of the protocol with the sensor node.

Kim et al. [6] propose an ID-based key establishment protocol from pairing-based cryptography which aims to reduce the communication cost of [4]. Being ID-based, their protocol replaces the public keys by the IDs, eliminating the need of exchange of certificates. This protocol reduces the communication cost but increases the overall computation cost of the protocol due to the expensive pairing computation. Like [4], this protocol also experiences a delayed user authentication (again by the proof of private key knowledge) on the sensor node's side, causing a DoS attack. An attempt to reduce the computation cost of [6] is made by Zhang et al. in [7]. They propose another version using pairing-based cryptography. Compared with Kim et al.'s protocol, their contribution is to scale down the number of point multiplication operations on a sensor node under the same communication complexity as in [6]. However, their protocol does not authenticate the security manager at all which enables any one to establish a session key with the sensor node. Yasmin et al. [8] propose an authentication framework describing user authentication and session key establishment for WSNs using ID-based cryptography. However, they did not provide any concrete scheme for the establishment of session key between the user and the sensor node. Our proposed protocol can be integrated into their authentication framework to provide a concrete scheme for user authentication and session key establishment.

B. ID-based Key Establishment

This section lists the work related to the ID-based key establishment. In recent years, a few ID-based one-pass key establishment protocols [9], [10], [11], [12] have been designed for traditional networks. However, none of these schemes is efficient as all of these require pairing computations. Extensive use of pairings makes these schemes quite slow and computationally expensive, particularly for resource constrained sensor nodes consuming considerable resources on them. Other related work includes the pairing-free ID-based two-pass authenticated key establishment schemes, for instance, [13], [14], [15] using the same ID-based setup as used in our scheme. However, as mentioned earlier, the secure two-pass key establishment schemes consume more resources on sensor nodes in terms of computation and communication overheads than one-pass schemes.

III. The Proposed Session Key Establishment Protocol

In this section, we present our proposed ID-based one-pass authenticated key establishment protocol by introducing the four phases: *System Initialization*, *Private Key Generation*, *User Registration* and *Key Establishment*. The first two phases are performed once, before the deployment of the sensor network. In an ID-based cryptosystem, a private key generator (PKG) computes the private keys corresponding to *ID*s. In WSNs the base station, a resourceful device, is considered as trustworthy. In our scheme, the base station plays the role of *PKG* and computes the private keys for sensor nodes and users.

A. System Initialization

In this phase, the *Setup* algorithm runs on the base station (before deployment) and generates the system parameters, including master public key (*mpk*), and the corresponding master secret key (*msk*) by using a security parameter k. This algorithm performs the following steps:

(a) Specify q, p, E/F_p, P and $\mathbb{G}$ where
- q is a large prime number and p is the field size,
- E/F_p is an elliptic curve E over a finite field F_p,
- P is a base point of order q on the curve E and
- $\mathbb{G}$ is a cyclic group of order q under the point addition "+" generated by P.

(b) For *msk* $s \in_R \mathbb{Z}_q^*$, compute *mpk* as $P_{PKG} = sP$.

(c) Choose one hash function H: $\{0,1\}^* \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$.

(d) Choose one key derivation function χ: $\mathbb{G} \rightarrow \{0,1\}^k$.

(e) Output system parameters $\{q, p, E/F_p, P, \mathbb{G}, P_{PKG}, H, \chi\}$ and keep s secret.

B. Private Key Generation

In this phase, the *Extract* algorithm runs on the base station (before deployment) and computes the private keys of all sensor nodes corresponding to their *ID*s. This algorithm takes *msk* and a sensor node's *ID* as input and generates a private key corresponding to that *ID* using the well known Schnorr signature. For a sensor node I with identity ID_i, this algorithm performs the following steps:

(a) For $r_i \in_R \mathbb{Z}_q^*$, compute $R_i = r_iP$ and $c_i = H(ID_i, R_i)$.

(b) Compute private key as $s_i = c_is + r_i$.

(c) Output (s_i, R_i) where s_i is secret while R_i is public.

Here the private key s_i is the Schnorr signature on the ID of the node signed with the private key of the PKG. *ID*s, corresponding private keys and system parameters are stored on sensor nodes before deployment. Hence, every sensor node i stores $\{ID_i, s_i, R_i\}$ and system parameters.

C. User Registration

This phase is repeated every time when a new user is registered with the system. In this phase, the *Extract* algorithm runs on the base station and computes the private key for a user U corresponding to his identity ID_u in the same way as computed for sensor nodes in the *Private Key Generation* phase. The base station, who runs this algorithm, sends the private key to the user via a secure channel. Hence, every user U gets $\{ID_u, s_u, R_u\}$ and system parameters.

D. Key Establishment: One-Pass Authenticated Session Key Establishment

Whenever a user wants to access data from sensor nodes, he establishes a session key with the sensor node in his range after successfully authenticating himself to the sensor node. Whether the user query is processed by a single sensor node or a set of sensor nodes is related to the topic of *query processing in wireless sensor networks* and is not addressed in our paper. We now describe our ID-based one-pass session key establishment protocol between a user U and a sensor node I. Fig. 1 describes the steps of the protocol.

(a) The user U chooses at random $t \in \mathbb{Z}_q^*$ as ephemeral key and computes $y = ts_u$ and $L = yP$. U signs the ephemeral public key L together with ID_u, ID_i and TS and sends $[L, ID_u, ID_i, TS, Sig_{s_u}(L, ID_u, ID_i, TS)]$ to the sensor node I in his range. Here TS is the current time stamp to avoid a replay attack and $Sig_{s_u}(L, ID_u, ID_i, TS)$ is a signature signed by U using his private key s_u. Computing y from L is the so-called *Elliptic Curve Discrete Logarithm* (ECDL) problem, which is intractable.

(b) The sensor node I first checks the time stamp TS to avoid the verification of a replayed message. If this is a fresh message, I verifies the signature $Sig_{s_u}(L, ID_u, ID_i, TS)$. Successful signature verification implies the message is actually sent by the user U and is fresh. Hence, I accepts the message, otherwise the protocol is terminated at this stage. Next the sensor node I computes the shared secret $K_{i,u}$ as

$$K_{i,u} = s_iL \; (=s_its_uP)$$

and deletes L.

(c) The user U computes the same shared secret $K_{u,i}$ as

$$S_i \; = c_iP_{PKG} + R_i \text{ where } c_i = H(ID_i, R_i)$$
$$K_{u,i} = yS_i \; (=ts_us_iP)$$

U then deletes L, t and y.

The both parties then compute the shared session key as $SK = \chi(K_{u,i}) = \chi(K_{i,u}) = \chi(ts_us_iP)$, where χ is the key derivation function.

User U		**Sensor Node** I
$t \in_R \mathbb{Z}_q^*$		
$y = ts_u$		
$L = yP$	$\xrightarrow{L,ID_u,ID_i,TS,Sig_{s_u}(L,ID_u,ID_i,TS)}$	
$c_i = H(ID_i,R_i)$		
$S_i = c_i P_{PKG} + R_i$		$Ver(L,ID_u,ID_i,TS)$
$K_{u,i} = yS_i$		$K_{i,u} = s_i L$
$\left\{ \begin{array}{l} = ys_iP \\ = ts_us_iP \end{array} \right\}$		$\left\{ \begin{array}{l} = s_iyP \\ = s_its_uP \end{array} \right\}$
$SK_u = \chi(K_{u,i})$		$SK_i = \chi(K_{i,u})$
	$SK = SK_u = SK_i = \chi(ts_us_iP)$	
		$E = (SK_i \oplus ID_u \oplus ID_i)$
$E = Dec_{SK_u}(E')$	$\xleftarrow{E'}$	$E' = Enc_{SK_i}(E)$
$E \overset{?}{=} (SK_u \oplus ID_u \oplus ID_i)$		

Figure 1. Authenticated One-Pass Session Key Establishment Protocol with Key Confirmation

However, there is no guarantee that at the end of the secure run of the protocol both parties compute the key. Indeed, in any key establishment protocol, the sender of the last message cannot make sure whether or not its last message is received by the other party. The user may successfully finish the protocol with a key output. Although the adversary is not able to learn the computed key, the sensor node might not receive the user's message and consequently might not be able to compute the key. The assurance against this scenario is achieved via an *authenticated key establishment protocol with key confirmation (AKC)*. This is usually achieved by adding a key confirmation message to the authenticated key establishment protocol after the key has been established. Hence, after both parties establish the session key, the *Key Establishment* algorithm proceeds as follows:

(d) After key computation, the sensor node I performs the following steps:
   i) Computes the XOR of its computed key SK_i with ID_u and ID_i as follows: $E = (SK_i \oplus ID_u \oplus ID_i)$.
   ii) Encrypts E with SK_i using a secure symmetric encryption algorithm, i.e., $E' = Enc_{SK_i}(E)$ and sends E' to U.

(e) After U receives E', he performs the following steps:
   i) Decrypts E' using his computed key SK_u to obtain E, i.e., $E = Dec_{SK_u}(E')$.
   ii) Checks whether $E \overset{?}{=} (SK_u \oplus ID_u \oplus ID_i)$.

Successful verification implies that both parties have computed the shared session key. As user does not expect to receive any message from the sensor node to compute the key, he does not need to send a key confirmation message to the sensor node.

1) Authentication, Key Establishment and Query Privacy: To obtain sensor nodes data, the user first authenticates himself to his nearby sensor node, establishes a session key with it and then sends his query to it. The sensor node, after successful user authentication and session key establishment, processes the received user query, encrypts the query results and sends them back to the user. For privacy reasons, the user query needs to be encrypted in some situations [16] since users may not be willing to disclose their areas of interests. Due to the one-pass key establishment, query privacy can also be provided by the proposed protocol as follows: the user computes his shared session key, encrypts his query using computed session key and sends his signed ephemeral public key to the sensor node together with his encrypted query in a single message. The sensor node first authenticates the user by verifying the signature. If the signature verification fails, the protocol terminates here. Otherwise, the sensor node computes the same shared session key, decrypts the user query, processes it and sends the encrypted query results back to the user. Thus, only a single message is exchanged for authentication, key establishment and encrypted query transmission achieving transmission efficiency.

2) ID-based Signature: To sign the ephemeral public key, any secure ID-based signature (IBS) scheme with the same ID-based parameters can be used, for instance, the secure ***BNN-IBS*** [17] scheme proposed by Bellare et al., whose security is proved under the discrete logarithm problem. There are also some other secure variants of ***BNN-IBS***, e.g., ***vBNN-IBS*** [18] and ***SLL-IBS*** [19] which can be used. ***vBNN-IBS*** has already been used in WSNs to provide broadcast authentication.

3) Distributing The Public Information ID_i and R_i: One possible question might be how a user can obtain ID_i and R_i, the public information of a sensor node I. As the user is equipped with a resourceful device, it can store the ID_i and R_i pairs of the sensor nodes in user's range. In 160-bit ECC settings, the size of the ID_i and R_i pair is about

25 bytes. For say 5000 sensor nodes in user's range, the total storage required will be about 125KB. This is an acceptable storage overhead on a resourceful user device to provide security with efficiency on resource constrained sensor nodes. User can also obtain ID_i and R_i pair from the base station via any other means e.g., Internet, before making a query to I. Note the difference that here ID_i and R_i are two identity elements of I and not the public keys as in the traditional public key crypto system where public keys are verified using the signed certificates. Here, if some one tries to use fake ID_i and R_i pair, he would not be able to generate corresponding private key s_i which is generated using *msk*. Generating such a valid triplet without *msk* would be equivalent to forging Schnorr signature. The ID-based two-pass key establishment schemes mentioned in Section II.B and the ID-based signature schemes mentioned in Section III.D.2 all use the same ID-based setup. This setup allows to construct efficient pairing-free ID-based schemes while handling the problem of public keys/certificates.

IV. Security Analysis

The security of our protocol is formally analyzed using the reductionist proof technique under the standard Computational Diffie-Hellman (CDH) assumption. The CDH assumption assumes that for a large security parameter k (e.g., $k \geq 160$) it is intractable to compute abP given $\langle P, aP, bP \rangle$, where P is a random generator of $\mathbb{G}$ and a, b are uniformly selected at random from $\mathbb{Z}_q$. By assuming that the CDH assumption holds in $\mathbb{G}$ we show that the proposed protocol is secure in the ID-eCk model [10]. Due to space limitation, the security model and rigorous proof have to be omitted in this paper. The detailed security analysis including security proof will be a part of the extended version of this paper. In this section, we informally discuss five security attributes appertaining to the proposed protocol.

Authentication. The proposed protocol provides the required authentication. There is only one message exchanged and that is sent by the user. Authentication of that single message is achieved by the verification of signature signed by the user. It is infeasible for an adversary to sign a message on behalf of a user without knowing user's private key. Successful signature verification by the sensor node I proves the fact that the ephemeral public key is actually sent by a legitimate user U. On the other side, $S_i (= s_i P)$ computed from I's public information assures the user that the session key is, in fact, established with I. Only the sensor node I with the valid corresponding private key s_i can compute the same session key. Authentication avoids the chances of the adversary mounting a man-in-the-middle attack.

Key Confidentiality. After the successful key establishment between a sensor node and a user, the only information available to the adversary is the public parameters and the ephemeral public key $L (= ts_u P)$. However, he cannot compute the user U's private key s_u and/or ephemeral private key t from L since we assume there is no polynomial time algorithm to solve the ECDL problem. Furthermore, he cannot compute the shared secret $ts_u s_i P$ because it requires the knowledge of private keys of both the sensor node and the user. Hence, the key is computable only by the user U and the sensor node I.

Key Compromise. The random value for ephemeral private key t is separately generated for each session. Therefore, the established session key is computationally different for different sessions. A session key established between a compromised sensor node and a user would not enable an adversary to compute or learn any other session key established between any other legitimate sensor node and a user. Furthermore, it would not enable an adversary to learn the user's private key s_u from L due to the intractability of ECDL problem. In fact, the proposed protocol guarantees that the communication between an uncompromised sensor node and a user cannot be exposed, irrespective of the number of other nodes that are compromised.

Key Confirmation. In the proposed protocol, the key confirmation message E' provides the explicit key confirmation. The sensor node computes E' and sends it to the user so that the user can be assured that the sensor node has received the user's ephemeral public key and successfully computed the session key. However, the user does not expect to receive any message from the sensor node for key establishment, as he can compute the same session key by himself. Hence, the user does not need to send a key confirmation message back to the sensor node.

Replay Attack. In a replay attack, an adversary replays the previous successful user request to either establish a session key with the sensor node or to waste sensor node resources by the request verification. In the proposed protocol, because of user's signed message, the adversary will not be able to authenticate successfully and establish a key. Furthermore, the time stamp TS provides freshness. The sensor node checks time stamp before the signature verification to avoid the verification of a replayed request message. Depending on the transmission delay imposed by the communication channel between the user and the sensor node, the sensor node sets a time threshold leaving a potential attacker little time to mount a replay attack.

V. Performance Comparison

In this section, we evaluate the performance of our proposed protocol in two ways: firstly, by comparing it with the existing session key establishment protocols for WSNs in Tables I and II, and secondly, by comparing it with the other ID-based one-pass session key establishment protocols in Table III. The factors used to evaluate the performance

Table I
COMPUTATION COST COMPARISON WITH THE EXISTING SESSION KEY ESTABLISHMENT PROTOCOLS FOR WSNS

	Key Establishment Cost		**User Authentication Cost**		**Time (s)**
	User	*Sensor Node*	*Sensor Node*		*Sensor Node*
Huang et al. [4]	$4M+3H$	$3M+3H$	Signed certificate ver. (***ECDSA***)	$2M$	1.60
Kim et al. [6]	$2P+1M+1E+2H$	$3M+1E+2H$	Implicit ver.	NA	2.24
Zhang et al. [7]	$2P+1M+4H$	$2M+1E+3H$	Does not support	NA	1.92
Our scheme	$3M+1H$	$1M$	Signature ver. (***vBNN-IBS***)	$3M$	1.28

are the number of complex cryptographic operations including pairing, point multiplication and exponentiation operations (computation overhead), total number of messages exchanged in each protocol run (communication overhead) and the memory requirements (storage overhead). Since the sensor nodes are more resource-constrained than the users devices, we pay more attention to the efficiency of the protocol on the sensor node side than on the user side.

For 80-bit security, in an efficient and optimized implementation on a standard MICA2 sensor node, one pairing computation takes 1.90s [20] and one point multiplication takes 0.32s [21]. Note that if the basic operation in $\mathbb{G}$ is denoted multiplicatively $(*)$ instead of additively $(+)$, the point multiplication in $\mathbb{G}$ is then called exponentiation correspondingly and thus takes 0.32s. However, the exponentiation in the target group $\mathbb{G}_T$ (in the settings of pairing [22]) takes more time than exponentiation (or point multiplication) in $\mathbb{G}$ because of the fact that it computes arithmetic in $\mathbb{G}_T$ which is operated in a field much bigger than the field in which $\mathbb{G}$ is defined. In usual implementations of pairing, 1 exponentiation in $\mathbb{G}_T$ costs about equal to 4 exponentiations in a multiplicative group [22] or 4 point multiplications in an additive group. The overheads of hash operation and arithmetic operations in $\mathbb{Z}_q^*$ are very small compared to the above mentioned expensive cryptographic operations. Thus, we only consider the expensive cryptographic operations for performance analysis. In all tables, P denotes one pairing computation, H denotes one hash evaluation, M denotes one point multiplication or exponentiation in $\mathbb{G}$ and E denotes one exponentiation in target group $\mathbb{G}_T$.

A. Session Key Establishment for Wireless Sensor Networks

This section compares the proposed protocol with the existing session key establishment protocols for WSNs. Tables I and II show the comparison results.

1) Computation Overhead: In WSNs scenario, it is highly desirable for a security protocol to have low computational overhead on resource constrained sensor nodes. In Huang et al.'s key establishment protocol [4], the computation overhead on a sensor node is the verification of a signed certificate to extract user's public key and the computations of 3 point multiplications to compute session key. The user authentication, however, is achieved via key confirmation messages. For comparison purpose, we assume that the certificate verification requires the verification of an ***ECDSA*** signature. The ***ECDSA*** signature is considered more efficient for sensor nodes than RSA signature because of shorter key and signature sizes. ***ECDSA*** requires 2 point multiplications as expensive operations to verify a signature. Hence, the total computation overhead of Huang et al's protocol is 5 point multiplications. Kim et al.'s protocol [6] requires sensor nodes to compute 3 point multiplications and 1 exponentiation in $\mathbb{G}_T$. Zhang et al.'s protocol [7] brings down the computation cost of [6] by one point multiplication without providing user authentication. Our proposed protocol requires a sensor node to compute only 1 point multiplication to compute the session key and one signature verification to authenticate the user. For comparison with [4] and [6], we assume that the secure and efficient ID-based signature scheme ***vBNN-IBS*** [18] is used in our protocol for user authentication which requires 3 point multiplications for signature verification. It is clear from Table I that the overall computational load of the proposed protocol is still lower than the computational loads of both [4] and [6] and the key computation cost is lower than the key computation cost of [7]. At the same time, the proposed protocol has stronger security, as we shall discuss in Section V.A.6.

2) Time Consumption: We now compare the estimated total computation time taken by a sensor node to authenticate a user and derive a session key. The results of this time analysis are also given in Table I. Huang et al.'s protocol [4] requires a sensor node to compute 5 point multiplications and therefore, takes about 1.60s on it. Kim et al.'s protocol [6], on the other hand, computes 3 point multiplications and 1 exponentiation in $\mathbb{G}_T$. Considering the fact that the exponentiation in $\mathbb{G}_T$ costs four times than one point multiplication, the estimated computation time is about 2.24s for their protocol. Zhang et al.'s protocol [7] requires a sensor node to compute 2 point multiplications and 1 exponentiation in $\mathbb{G}_T$ for key computation (this protocol does not provide user authentication) and consumes about 1.92s on a sensor node. Considering the ID-based signature scheme ***vBNN-IBS*** [18], the total estimated computation time for the proposed protocol is about 1.28s for 4 point multiplications. This implies that compared with the protocols proposed by Huang et al., Kim et al. and Zhang et al., our protocol reduces the total computation time for key establishment and user authentication on a sensor node by 20%, 33%, and 43%, respectively, without mentioning that Huang et al.'s and Zhang et al.'s protocols are quite weak in security

Table II
COMMUNICATION COST COMPARISON WITH THE EXISTING SESSION KEY ESTABLISHMENT PROTOCOLS FOR WSNS

	Messages Exchanged	
	Key Establishment	*Key Confirmation*
Huang et al. [4]	4	2
Kim et al. [6]	3	1
Zhang et al. [7]	3	NA (Does not support)
Our scheme	1	1

(Refer to Section V.A.6). In addition, note that our protocol also improves the performance of a user by 25%, 82%, and 75% over Huang et al.'s, Kim et al.'s and Zhang et al.'s solutions, respectively. As improving the efficiency of the user side is not our focus in this paper, we do not discuss this issue in detail.

3) Communication Overhead: To achieve network resource efficiency and minimum latency, the number of message exchanges between the sensor node and the user should be as small as possible. Huang et al.'s protocol [4] and Kim et al.'s protocol [6] exchange 6 and 4 messages, respectively, for key establishment and user authentication. The key confirmation messages are compulsory to provide user authentication in their protocols. Zhang et al.'s protocol [7] exchanges 3 messages for the key establishment. The proposed protocol exchanges only 1 message for both key establishment and user authentication. Hence, the proposed protocol causes very low communication overhead than the other three protocols for WSNs as shown in Table II.

4) Storage Overhead: The storage overhead of the proposed protocol is similar to the other protocols which is not very high. The proposed protocol does not require sensor nodes to store any user credentials (IDs, public keys, certificates etc.) for the verification of a user's legitimacy and so provides storage efficiency. The only storage overhead is the sensor node's ID, corresponding ID-based key and the system parameters.

5) Scalability: Since the overheads of the proposed protocol do not increase with the network size, it supports large scale deployment of WSNs. New sensor nodes and outside users can be added to the WSN easily at any time. Preloaded with ID, secret key and public parameters, the new sensor node can establish a key with any legitimate user after user authentication. The new users simply need to register themselves to the base station and get their private keys and system parameters. ID-based cryptography relieves sensor nodes from storing any users specific information to authenticate them, and consequently eliminates the restriction on the number of outside users.

6) Analysis - Performance Versus Security: As discussed earlier, Huang et al.'s protocol [4] is not secure since user can learn a sensor node's private key after one run of the protocol with that node. This is a severe security attack against a key establishment protocol which cannot be tolerated, no matter how efficient a protocol is. Another drawback is the DoS attack caused by the delayed user authentication. On the other hand, Zhang et al.'s protocol [7] does not support user authentication at all allowing any adversary to establish a session key and obtain sensor nodes data. Hence, these two protocols lack the required security. Kim et al.'s protocol [6] also suffers from the DoS attack caused by the delayed user authentication wasting sensor node's resources. The proposed protocol authenticates a user at the first step by the verification of a signed user's ephemeral public key and the time stamp. Furthermore, it is not possible for any participant or any adversary to learn any participant's private key.

However, an adversary can cause a sensor node to verify a fake signature in the proposed protocol wasting its resources. To see how devastating this attack is as compared to the DoS attack in Kim et al.'s protocol, we assume the secure and efficient ID-based signature scheme ***vBNN-IBS*** [18] for signature generation in our protocol. To detect a fake user request sent by an adversary, a sensor node will perform 3 point multiplications in the proposed protocol and 3 point multiplications and 1 exponentiation in $\mathbb{G}_T$ in Kim et al.'s protocol. Before a user is authenticated, 4 messages will be exchanged in Kim et al.'s protocol while only 1 message will be exchanged in the proposed protocol as after receiving the first message from the user the sensor node can find out the fake request and terminate the protocol. Thus, DoS attack in the proposed protocol is much less devastating than in Kim et al's protocol saving both the communication and the computation costs. Hence, the proposed protocol provides better performance versus security than the existing session key establishment protocols for the WSNs.

B. ID-based One-Pass Session Key Establishment

In this section, by comparing our protocol with the existing ID-based one-pass session key establishment protocols, we show that the significant efficiency improvement achieved by the proposed protocol is its very low computation overhead. Note that the existing protocols are not originally claimed for WSNs. What we are discussing here is that these protocols do not suit WSNs due to their low performances. Table III compares our protocol with the existing protocols by listing the key establishment costs for both sides of each protocol. Compared to the existing protocols, the proposed protocol is computationally efficient on the

Table III
COMPARISON WITH THE EXISTING ID-BASED ONE-PASS KEY ESTABLISHMENT PROTOCOLS

	Key Establishment Cost		Time (s)
	User	*Sensor Node*	*Sensor Node*
Benit et al. [9]	$1P+2M+1H$	$1P+1H$	1.90
Okamoto et al. (*II*) [11]	$1P+3M+2H$	$1P+1M+2H$	2.22
Wang [12]	$1P+3E+2H$	$1P+2E+2H$	4.46
Gorantla et al. [10]	$1P+2M+1H$	$1P+1M+1H$	2.22
Our scheme	$3M+1H$	$1M$	0.32

sensor node's side requiring only one point multiplication but no pairing computation. One pairing computation on a standard MICA2 sensor node takes 1.90s versus 0.32s for a point multiplication on the same node and therefore, consumes resources equal to 6 point multiplications. Due to the lack of pairing computations on both sides, our proposed protocol provides much better performance than the existing protocols. Table III also shows the estimated time that a sensor node consumes if the existing protocols are applied in WSNs. It is clear from Table III that our protocol is almost 6 times faster than Benit et al.'s protocol [9], which is the best existing ID-based one-pass key establishment protocol in terms of efficiency on the sensor node side. Moreover, if we also count the user authentication (signature verification) cost mentioned in Section V.A.2, the proposed protocol still outperforms all the existing protocols with the total time of 1.28s. Note that not all the existing protocols include user authentication, for instance, Benit et al's protocol. Hence, the proposed protocol is the first most suitable ID-based one-pass session key establishment protocol for WSNs.

VI. Conclusion

In this paper, we propose a new secure and efficient ID-based one-pass key establishment protocol for WSNs. To the best of our knowledge, this is the first ID-based one-pass authenticated key establishment protocol without pairing. Lack of pairing computation makes it much more efficient for sensor nodes than the existing ID-based one-pass key establishment protocols. At the same time, it enjoys all the desirable security properties for session key establishment protocols. The security and efficiency analysis shows that the proposed protocol performs better than the existing ID-based one-pass key establishment protocols and the key establishment protocols for WSNs.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[2] A. Shamir, "Identity-based Cryptosystems and Signature Schemes," in *Advances in Cryptology - CRYPTO 1984*. Springer-Verlag, 1985, pp. 47–53.

[3] W. Diffie, P. C. Van Oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," *Design Codes and Cryptograhpy*, vol. 2, no. 2, pp. 107–125, 1992.

[4] Q. Huang, J. Cukier, H. Kobayashi, B. Liu, and J. Zhang, "Fast Authenticated Key Establishment Protocols for Self-Organizing Sensor Networks," in *Proc. WSNA'03*. ACM, 2003, pp. 141–150.

[5] X. Tian, D. Wong, and R. Zhu, "Analysis and improvement of an authenticated key exchange protocol for sensor networks," *Communications Letters*, vol. 9, no. 11, pp. 970 – 972, 2005.

[6] Y. Kim, H. Lee, J. Park, L. Yang, and D. Lee, "Key establishment scheme for sensor networks with low communication cost," in *Autonomic and Trusted Computing*, ser. LNCS, vol. 4610. Springer Berlin / Heidelberg, 2007, pp. 441–448.

[7] L.-P. Zhang and Y. Wang, "An ID-Based Key Agreement Protocol for Wireless Sensor Networks," in *Proc. Int. Conf. Information Science and Engineering*. IEEE Computer Society, 2009, pp. 2542–2545.

[8] R. Yasmin, E. Ritter, and G. Wang, "An Authentication Framework for Wireless Sensor Networks using Identity-Based Signatures," in *Proc. Int. Conf. CIT'10*. IEEE Computer Society, 2010, pp. 882–889.

[9] W. Benits Jr and R. Terada, "An IBE Scheme to Exchange Authenticated Secret Keys," Cryptology ePrint Archive, Report 2004/071, 2004, http://eprint.iacr.org/.

[10] M. C. Gorantla, C. Boyd, and J. M. González Nieto, "ID-based One-pass Authenticated Key Establishment," in *Proc. AISC*. Australian Computer Society,, 2008, pp. 39–46.

[11] T. Okamoto, R. Tso, and E. Okamoto, "One-Way and Two-Party Authenticated ID-Based Key Agreement Protocols Using Pairing," in *Proc. MDAI' 05*, ser. LNCS, vol. 3558. Springer-Verlag, 2005, pp. 122–133.

[12] Y. Wang, "Efficient Identity-Based and Authenticated Key Agreement Protocol," Cryptology ePrint Archive, Report 2005/108, 2005, http://eprint.iacr.org/.

[13] X. Cao, W. Kou, and X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Information Sciences*, vol. 180, no. 15, pp. 2895–2903, 2010.

[14] Y.-M. Tseng, "An Efficient Two-Party Identity-Based Key Exchange Protocol," *Informatica*, vol. 18, no. 1, pp. 125–136, 2007.

[15] R. Zhu, G. Yang, and D. Wong, "An Efficient Identity-Based Key Exchange Protocol with KGS Forward Secrecy for Low-Power Devices," in *WINE'05*, ser. LNCS, vol. 3828. Springer Berlin/Heidelberg, 2005, pp. 500–509.

[16] B. Carbunar, Y. Yu, L. Shi, M. Pearce, and V. Vasudevan, "Query privacy in wireless sensor networks," in *Proc. SECON'07*, 2007, pp. 203–212.

[17] M. Bellare, C. Namprempre, and G. Neven, "Security Proofs for Identity-Based Identification and Signature Schemes," in *Proc. EUROCRYPT'04*, ser. LNCS, vol. 3027. Springer, 2004, pp. 268–286.

[18] X. Cao, W. Kou, L. Dang, and B. Zhao, "IMBAS: Identity-based Multi-user Broadcast Authentication in Wireless Sensor Networks," *Computer Communications*, vol. 31, no. 4, pp. 659–667, 2008.

[19] D. Galindo and F. D. Garcia, "A Schnorr-Like Lightweight Identity-Based Signature Scheme," in *Proc. AFRICACRYPT'09*, vol. 5580. Springer, 2009, pp. 135–148.

[20] L. B. Oliveira, D. F. Aranha, C. P. L. Gouvêa, M. Scott, D. F. Câmara, J. López, and R. Dahab, "TinyPBC: Pairings for Authenticated Identity-Based Non-Interactive Key Distribution in Sensor Networks," *Computer Communications*, vol. 34, no. 3, pp. 485–493, 2011.

[21] D. F. Aranha, R. Dahab, J. López, and L. B. Oliveira, "Efficient Implementation of Elliptic Curve Cryptography in Wireless Sensors," *Advances in Mathematics of Communications*, vol. 4, no. 2, pp. 169–187, 2010.

[22] L. Chen, P. Morrissey, and N. P. Smart, "Pairings in Trusted Computing," in *Proc. Pairing'08*, ser. LNCS, vol. 5209. Springer Berlin/Heidelberg, 2008, pp. 1–17.

Secret Key Sharing and Rateless Coding for Practical Secure Wireless Transmission

Wei Liu†1, Chunjie Duan‡, Yige Wang‡, Toshiaki Koike-Akino‡, Ramesh Annavajjala‡, and Jinyun Zhang‡
† Department of EECS, Syracuse University, Syracuse, NY 13244, USA
‡ Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA
Email: wliu28@syr.edu, {duan, yigewang, koike, annavajjala, zhang}@merl.com

***Abstract*—We discuss a secure wireless communication scheme, focusing on designing two major components: the key generation and the coding scheme. To achieve high key matching rate, we propose a feed-forward and feed-back quantization. The proposed scheme offers 1 dB improvement over the best known schemes. We also propose a universal quantization scheme with feed-forward/feed-back and show that its performance is the same as, or better than the other schemes which require prior distribution information. For rate-adaptive coding, we propose the use of rateless codes. Our evaluations show that the rateless code can offer significant performance gain over a low-density parity-check (LDPC) code. Moreover, we implement a soft input rateless decoder which offers additional gains. The overall security performance of our design based on these two components significantly outperforms existing designs.**

Keywords - Secret key generation; Wireless communications; Quantization; Rateless codes; Rate compatible codes.

I. Introduction

A security system is only as good as its weakest part, frequently, the Key Management System (KMS), which consists of the key management, key derivation, storage and distribution [20]. Unfortunately, designing a good KMS is an extremely hard problem and not all designers agree on how to construct it. This problem is exacerbated with the current trend of moving into the “Internet of Things” and ”cyber physical systems”, where large scale, complex, ad-hoc, often infrastructure-less wireless sensor networks and the broadcast nature of the channels makes the problem much harder to tackle.

Most commonly used key management techniques are based on public key cryptography and requires a Public key Infrastructure (PKI). PKI is generally not suitable for ad-hoc systems where a) infrastructure is not guaranteed; b) key derivation functions are computation intensive and therefore cannot be carried out in low power and low cost devices such as wireless sensor nodes; and c) designing a key storage scheme that survives frequent system and node reboot is also challenging.

Physical-Layer Security (PLS) has been proposed [23, 24] with the hope to address this problem. It is envisioned that by establishing security at the physical or link layer, a network wide KMS is no longer needed, or can be greatly simplified and we can potentially remove the requirement of having a PKI in the network, that would result in a simple and cost/power efficient yet secure network.

Most of the PLS schemes proposed so far fall into two categories: a) generating secret keys from correlated sources based on wireless channel reciprocity, and then applying traditional cryptography and b) adapting transmission rate to the information-theoretic channel capacity to achieve a positive secrecy capacity.

In the former category, where Alice and Bob try to generate a pair of secret keys based on correlated observations X^n and Y^n respectively, as illustrated in Fig. 1. Assume that Alice and Bob can communicate through an unauthenticated public channel which might be observed by an eavesdropper, Eve, who might have side information Z^n which can be correlated with X^n and Y^n. The information-theoretical study of this problem was provided by Maurer [1] and Ahlswede and Csiszár [2], where the secrecy capacity defined as the maximum key generation rate is given for the special case when Eve has no side information Z^n, and lower bounds and upper bounds on the secrecy capacity are provided for general cases. While the theoretical aspects of this problem are well understood, there is a growing interest in designing practical secret key generation algorithms to approach the secrecy capacity of the key generation rate.

One way to design such algorithms is to exploit the inherent randomness in the wireless channel between two nodes as the source for extracting secret key sequences [3–12]. The security of these schemes relies on the *reciprocity principle* of the radio wave propagation which states that the multipath properties of the radio channel such as channel gains, phase shifts and delays at any point in time are identical in both direction of the communication link [8]. In addition, these properties are intrinsically *spatially specific* in a multipath radio environment due to the scatter effects. An eavesdropper at a third location more than a few wavelengths away from the two legitimated users will observe a different and uncorrelated radio channel [13]. As a result, the two legitimated users can generate a secret key based on the shared common randomness which is unavailable to the eavesdropper. Among these schemes, channel gain information [4, 5, 7–12] is the most commonly used. Most of these existing algorithms on key generation from channel gain measurements consist three steps: quantization, information reconciliation [14], and privacy amplification [15]. The purpose of quantization is to convert the real channel measurements into binary bit strings. Information reconciliation is aimed at generating an identical random sequence between the two legitimated users by communicating through the public channel and privacy amplification is used to extract a perfect secure key from the identical random sequence agreed in the information reconciliation stage. As the foundation of the whole process,

[1] This work is done during the author’s internship at MERL

designing a good quantization algorithm is crucial. In this paper, we focus on the quantizer design. The purpose is to generate two binary sequences at two legitimated users with bit mismatch rate as small as possible.

Traditional quantization schemes such as equiprobable quantization [4] and level crossing quantization [7] have the limitation of high bit mismatch probability or low key generation rate. Moreover, these quantization schemes do not take advantage of the public channel. Recently, Wallace *et al.* [16] proposed a new quantization algorithm utilizing a one-way communication over the public channel and achieved a significant improvement in the bit mismatch probability. The over-quantization algorithm proposed in [12] used a similar idea to achieve higher key generation rate within 1.1 bits from the secret key capacity.

Schemes in the latter category are all based on the principle that data transmitted at a rate exceeding the channel capacity cannot be decoded by the receiving end. Many papers derive the secrecy capacity for various channels including wire-tap channels [21], broadcast channels [22], multiple access channels [26], relay broadcast channels [24], fading channels [23], *etc*. Fundamentally, to achieve a positive secrecy capacity, it is required that the channel capacity of the eavesdroppers be lower than that of the intended receivers. This limits the practicality of these types of schemes, as it is nearly impossible to guarantee such a condition in wireless environment. Some attempts have been made to address this. For instance, Pinto et al. showed in [27] that if physical security can be guaranteed (e.g., a small area around the intended receiver is secured from eavesdroppers), secrecy capacity is improved. Advanced transmission schemes are also proposed to improve the secrecy capacity, e.g., the secrecy capacity improvement with MIMO precoding is analyzed in [25]. Beamforming and artificial jamming noise injection are shown to improve the secrecy capacity [28], with the assumption that certain knowledge of the eavesdropper's channel is available to the transmitter.

To the best of our knowledge, however, there is yet a practical way of guarantees that legitimate receivers have favorable channels, and the research in this area suffers from being limited to mostly just theoretical analysis.

In this paper, we propose a secure transmission scheme by combining both ideas. The nodes first carry out key generation using the quantization schemes that offers high bit matching rate (BMR), and then carry out secure transmissions by scrambling data to create an artificial channel that is favorable to the legitimate node pairs, and then apply rate adaptive coding with a rate tightly coupled to instantaneous capacity.

The first focus of the design is the key generation process, in particular, quantization schemes that allow an independently generated keys pair to have high BMR. Motivated by the theoretical analysis in [1, 2], we propose a quantization algorithm by exploiting the public channel through two-way communications; more specifically, feed-forward and feed-back quantization. For the case that the channel measurements are Gaussian distributed, the proposed algorithm improves around $1\,\text{dB}$ at high signal-to-noise ratio (SNR) over the best known scheme [16]. Moreover, we propose a universal quantization scheme with feed-forward and feed-back for the general case when the two end nodes do not know the prior distribution. It is demonstrated through simulations that the proposed universal scheme works asymptotically close to the case with known prior information.

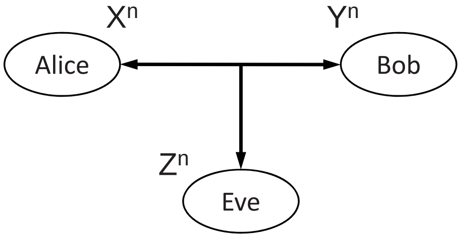


Fig. 1. Secret key sharing between Alice and Bob.

The second focus of the design is the transmission scheme, where we propose applying channel scrambling and rateless coding to achieve the maximum achievable secrecy capacity.

The key management of PLS scheme proposed in this paper is simple, as the link keys are generated locally between two communicating nodes and are used locally. A node only need to store and update the keys to its immediate (one hop) neighbors. PLS schemes such as the one discussed in this paper can be seamlessly integrated with existing security mechanisms in the upper layer to enhance the overall security level of wireless systems.

The rest of the paper is organized as follows. Section II gives an overview of the transmission scheme and the node design. In Section III, we introduce the system model for key generation. Section IV introduces the new quantization algorithm which uses the feed-forward and feed-back scheme. The universal quantization scheme is detailed in Section V. Coding with rateless codes is discussed in Section VI. We conclude this paper in Section VII.

II. Secure Transmission with Mismatching Keys

The transmitter and the receiver under consideration have the structure shown in Fig. 2. Both legitimate nodes, Alice and Bob, implement a *Key Generator* block. Each key generator produces a secret key string, i.e., K_A and K_B. The transmitter encodes a message M with an inner code and then scrambles the coded message C with its secret key, K_A. The scrambled data S will be transmitted directly, or alternatively, coded with an outer code before being transmitted over the wireless channel.

Correspondingly, a decoder implemented in the receiver to decode the outer code is necessary. The receiver descrambles the output of the outer decoder R_B with the secret key generated by its own key generator, K_B, and feed the descrambled message C_B to its inner code decoder. The receiver's inner decoder decodes the message. We also assume Eve has full knowledge of the transmission scheme and is equipped with the same functional block as in a legitimate receiver. Eve may be able to produce a key, K_E, descramble and decode the receiver message.

In the proposed receiver, rate adaptive codes are used as the inner code to maximize the secrecy capacity. We focus on the performance of rateless codes in this paper. During

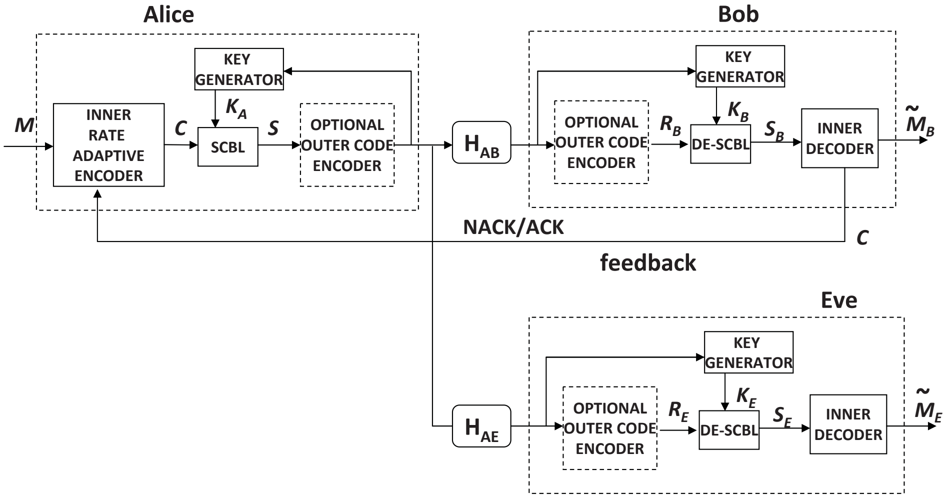


Fig. 2. Transmitter and receiver structures.

the transmission of a message M, the transmitter's encoder continues sending out the coded stream S until the receiver has successfully decoded the message, $\tilde{M}$. A feedback channel is needed for the receiver to send an acknowledgement (ACK) or a negative ACK (NACK) back to the transmitter. The mechanism of rateless codes guarantees that the rate of the transmission matches the instantaneous capacity.

The process of key generation needs to be carried out before any secure data transmissions. Alice and Bob first perform key generations. Both nodes transmit sounding signals alternatively to enable the other side to perform channel estimations. Key pairs need to be updated periodically but not necessary at every data transmission. The update frequency is determined by the overhead. In most cases, the data packets are preceded by preambles so that the receiver can estimate the channel for coherent detections. Therefore, this allows the key generation/update be integrated into the transmission period and further reduces the overhead.

The overall secrecy capacity of such a scheme is determined by two factors:

1) The difference of the key matching rates
2) The closeness of the coding rate

The former item dictates the *maximum* secrecy capacity while the latter item affects the *achievable* secrecy capacity. They both need to be optimized such that overall security can be maximized. The following sections describe how they can be optimized separately.

III. Secret Key Generation

A. Secret Key Generation Based on Channel Measurements

Alice and Bob want to share a secret key by measuring the channel response between them. With a channel reciprocity, the channel estimates of Alice and Bob are highly correlated. We can model Alice and Bob's observations, X^n and Y^n, as independent and identically distributed (*i.i.d.*) n repetitions of dependent random variables, X and Y, from a joint distribution $f(X, Y)$. Based on those correlated observations X^n and Y^n, Alice and Bob generate a secret key by communicating over a public channel, while the messages transmitted through the public channel may be observed by eavesdroppers. We assume that the eavesdropper, Eve, can only observe the channel without any message modifications, i.e., a passive eavesdropper. The messages transmitted through the channel (possibly two way) are denoted as $\mathbf{V}$.

In this paper, we focus on the case when X and Y are jointly Gaussian; more specifically,

$$X = G + W_A, \qquad Y = G + W_B, \tag{1}$$

where $G \sim \mathcal{N}(0, P)$, $W_A \sim \mathcal{N}(0, N_A)$ and $W_B \sim \mathcal{N}(0, N_B)$ denote the channel response, the estimation error at Alice and the estimation error at Bob, respectively. For simplicity, we consider the case of $N_A = N_B = N$.

IV. Quantization with Feed-forward and Feed-back

In this section, we discuss a quantization-based key generation algorithm. We use a core quantization module based on a scalar equiprobable quantizer because of its lightweight and its property of maximum entropy.

A. Algorithm Description

Our secure key sharing algorithm consists of the following five steps.

1) Initialization phase: Alice and Bob agree on the quantization level L and the feed-forwarding level m. Both Alice and Bob use the same equiprobable scalar quantizer, which is designed for a zero-mean and unit-variance Gaussian random variable. For the L-level quantizer, the quantization boundary q_i to indicate L intervals, $(q_0, q_1], (q_1, q_2], \cdots, (q_{L-1}, q_L)$, is chosen such that [4]

$$Q(q_i) \triangleq \int_{q_i}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{x^2}{2}\Big) \mathrm{d}x = \frac{L-i}{L}, \tag{2}$$

for any $i \in \mathbb{Z}_L$, where $Q(\cdot)$ denotes the Gaussian tail function and $\mathbb{Z}_L \triangleq \{0, 1, \cdots, L-1\}$ denotes an integer set. Here, we set $q_0 = -\infty$ and $q_L = \infty$. Gray coding is used for mapping the quantizer indices to bits.

For generating feed-forwarding information, each quantization interval $(q_{i-1}, q_i]$ is further split into m sub-intervals, $(t_{i-1,0}, t_{i-1,1}], (t_{i-1,1}, t_{i-1,2}], \cdots, (t_{i-1,m-1}, t_{i-1,m}]$, where $t_{i-1,0} = q_{i-1}$ and $t_{i-1,m} = q_i$, such that each sub-interval $(t_{i-1,k}, t_{i-1,k+1}]$ for $k \in \mathbb{Z}_m$ has an identical probability of

$$Q(t_{i-1,k}) - Q(t_{i-1,k+1}) = \frac{1}{mL}. \tag{3}$$

For each sub-interval, we index them by $0, 1, \cdots, m-1$ in an ascending order.

2) Channel sounding phase: Alice and Bob sequentially send known training signals to each other in order to measure the channel between Alice and Bob. Given each channel estimate X_i and Y_i for any $i \in \mathbb{N}_n$ (where n denotes the total number of measurement times), Alice and Bob individually quantize them into $\log_2(L)$-bit indices $K_A(i)$ and $K_B(i)$, using the L-level equiprobable scalar quantizer. Note that the quantization is done with a power normalization of $\beta = \frac{1}{\sqrt{P+N}}$ to have unity variance for quantizing data. The quantized data at $i \in \mathbb{N}_n$ is then given as

$$\begin{aligned} K_A(i) &= \big\{j : \beta x_i \in (q_j, q_{j+1}]\big\}, \\ K_B(i) &= \big\{j : \beta y_i \in (q_j, q_{j+1}]\big\}, \end{aligned} \tag{4}$$

where $\mathbb{N}_n \triangleq \{1, 2, \ldots, n\}$ is a positive integer set. After n observations, Alice and Bob obtain $n \log_2(L)$-bit quantized information, $\mathbf{K}_A = [K_A(1), K_A(2), \cdots, K_A(n)]$ and $\mathbf{K}_B = [K_B(1), K_B(2), \cdots, K_B(n)]$, respectively.

3) Feed-forward phase: Alice generates a $\log_2(m)$-bit feed-forwarding data $V_a(i)$ from X_i such that $V_a(i)$ is the sub-interval index of the interval $K_A(i)$, more specifically, we can write $V_a(i)$ for any $i \in \mathbb{N}_n$ as follows:

$$V_a(i) = \big\{j : \beta x_i \in (t_{K_A(i),j}, t_{K_A(i),j+1}]\big\}. \tag{5}$$

Through the public channel, Alice then sends an $n \log_2(m)$-bit message, $\mathbf{V}_a = [V_a(1), V_a(2), \cdots, V_a(n)]$ towards Bob where each sub-interval index $V_a(i)$ can be expressed by its binary natural code representation with $\log_2(m)$ bits. We propose one additional step termed feed-back phase to further improve the bit mismatch rate.

4) Feed-back phase: Upon receiving the feed-forward information $\mathbf{V}_a$ from Alice and his own observation Y^n, Bob employs a maximum *a posteriori* probability (MAP) estimation of $\mathbf{K}_A$. In the MAP estimation, for each $Y_i = y_i$ and $V_{ai} = v_{ai}$ (for any $i \in \mathbb{N}_n$), Bob searches for the index j_i such that

$$j_i = \operatorname*{argmax}_{j \in \mathbb{Z}_L} \Pr\big(\beta X_i \in (q_j, q_{j+1}] | Y_i = y_i, V_{ai} = v_{ai}\big). \tag{6}$$

With the MAP estimate j_i and the original quantized data $K_B(i)$, Bob generates feed-back information $V_b(i)$ which is set to be one if $j_i \neq K_B(i)$ and zero otherwise, for each $i \in \mathbb{N}_n$. The feed-back message $\mathbf{V}_b = [V_b(1), V_b(2), \cdots, V_b(n)]$ of n bits is sent to Alice through the public channel.

5) Key generation phase: Based on the feed-back message $\mathbf{V}_b$, Alice skips the corresponding $K_A(i)$ if $V_b(i) = 1$, for $i \in \mathbb{N}_n$, and sets the remaining as her secret key $\mathbf{K}_A$. Similarly, for each $i \in \mathbb{N}_n$, Bob also skips the corresponding indices $K_B(i)$ if $V_b(i) = 1$ and produce his secret key using the remaining bits.

B. Discussions

1) MAP estimation: We call the algorithm described above *MAP with feed-back* key generation algorithm. For the case that X^n and Y^n are jointly Gaussian as in (1), by noticing that $\Pr(V_{ai} = v_{ai} | Y_i = y_i)$ is a constant for given y_i and v_{ai}, we have

$$\begin{aligned} &\Pr(\beta X_i \in (q_{j-1}, q_j] | Y_i = y_i, V_a(i) = v_{ai}) \\ &\propto \Pr(\beta X_i \in (q_{j-1}, q_j], V_a(i) = v_{ai} | Y_i = y_i) \\ &= \Pr(\beta X_i \in (t_{j-1,v_{ai}}, t_{j-1,v_{ai}+1}] | Y_i = y_i) \\ &= Q\left(\frac{\frac{1}{\beta} t_{j-1,v_{ai}} - \mu_i}{\sigma_i}\right) - Q\left(\frac{\frac{1}{\beta} t_{j-1,v_{ai}+1} - \mu_i}{\sigma_i}\right), \end{aligned} \tag{7}$$

where $\mu_i = \rho y_i$, $\rho = \sqrt{\frac{P}{P+N}}$ and $\sigma_i = \sqrt{(P+N)(1-\rho^2)}$. The last equality comes from the fact that $(X_i | Y_i = y_i) \sim \mathcal{N}(\mu_i, \sigma_i^2)$.

Hence, the MAP estimation in (6) is equivalent to find the index $j \in \mathbb{Z}_L$ such that interval $(t_{j,v_{ai}}, t_{j,v_{ai}+1}]$ is the closest one to $\mu_i = \rho y_i$ in the sense of Euclidean distance. Or equivalently, Bob will decide $j_i = j \in \mathbb{Z}_L$ if $y_i \in (\bar{q}_j, \bar{q}_{j+1}] = \frac{1}{\rho}(\tilde{q}_j, \tilde{q}_{j+1}]$, where

$$\tilde{q}_j = \begin{cases} \frac{1}{2}(t_{j-1,v_{ai}+1} + t_{j,v_{ai}}), & 1 \leq j \leq L-2, \\ -\infty, & j = 0, \\ \infty, & j = L-1. \end{cases} \tag{8}$$

We remark here that the process of finding new quantization regions for Bob according to feed-forward bits is slightly different from the one described in [16]. In [16], a middle point as in (8) is chosen as the new quantization region while the optimal one should multiply a coefficient of $\frac{1}{\rho}$.

The procedure of finding j_i is illustrated in Fig. 3 for $L = 4$ and $m = 2$. Initial quantization codebook for both Alice and Bob is $(-\infty, q_1], (q_1, q_2], (q_2, q_3], (q_3, \infty)$. The observations of Alice and Bob are X and Y, respectively. In this example, Alice will quantize her observation into a two-bit information $K_a = [0, 1]$ and send a feed-forwarding data $V_a = 0$ towards Bob. By observing $V_a = 0$, Bob calculates his MAP codebook $(-\infty, \bar{q}_1], (\bar{q}_1, \bar{q}_2], (\bar{q}_2, \bar{q}_3], (\bar{q}_3, \infty)$ using (8), and finds $j_i = 1$ to generate the feed-back information.

2) Security: Transmitting the messages $\mathbf{V}_a$ and $\mathbf{V}_b$ through the public channel does not result in release of information about the secret key. This is because $V_a(i)$ is equiprobable, namely $\Pr(V_a(i) = v_{ai}) = 1/L$, and $V_b(i)$ indicates only the positions of the skipped bits.

3) Simulation results: Fig. 4 shows the performance of the algorithm described above (with 10^6 runs). As we can see from this figure, interaction between two nodes significantly improves the key agreement performance in bit mismatch rate at high SNR (higher than 10 dB), where SNR is defined as P/N. The introduction of feed-back phase brings approximately 1 dB gain over the scheme without feed-back. To compare the performance with the result by Wallace *et al.* [16], we also plot MAP without feed-back scheme in which the feed-back step is dropped in the algorithm. The 1-bit feed-forwarding in [16], though slightly different from the MAP without feed-back scheme as discussed above, achieves almost the same performance as the optimal MAP scheme. It is shown by

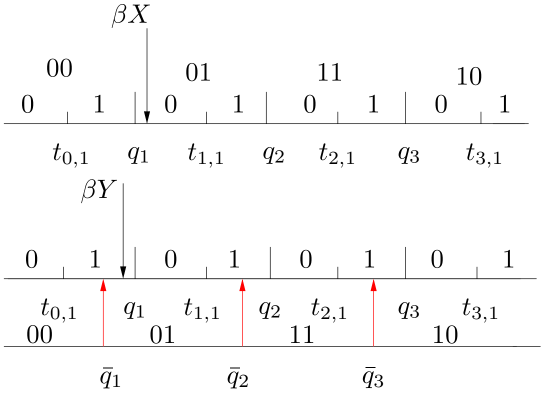

βX
00
01
11
10
0 1 0 1 0 1 0 1
$t_{0,1}$ q_1 $t_{1,1}$ q_2 $t_{2,1}$ q_3 $t_{3,1}$
βY
0 1 0 1 0 1 0 1
$t_{0,1}$ q_1 $t_{1,1}$ q_2 $t_{2,1}$ q_3 $t_{3,1}$
00 01 11 10
$\bar{q}_1$ $\bar{q}_2$ $\bar{q}_3$

Fig. 3. MAP with feed-back quantizer.

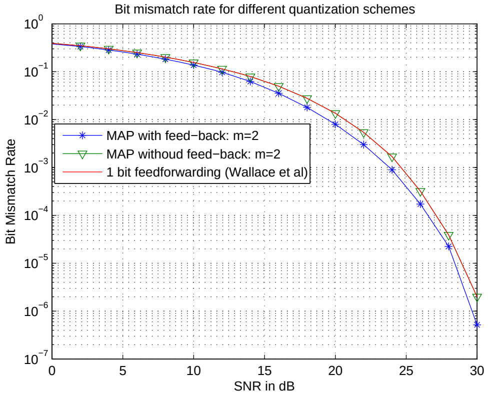


Fig. 4. Bit mismatch rate for 2-bits quantization.

simulation that the average number of skipped bits is relatively small (less than 15%).

V. UNIVERSAL KEY GENERATION ALGORITHM

The algorithm described in the previous section assumes prior knowledge of the joint distribution of X^n and Y^n is available to the nodes. However, in most of the practical applications, we might not have such information. In this section, we deal with this issue by introducing a universal key generation scheme in which no prior knowledge about the underlying joint distribution is required. The core quantization module is a universal equiprobable quantizer described below.

A universal equiprobable scalar quantizer for an arbitrarily distributed random sequence X^n is a set of intervals $[q_0, q_1), [q_1, q_2), \cdots, [q_{L-1}, q_L]$, and the boundary q_i (for $i \in \mathbb{Z}_L$) can be calculated as follows. Let us sort $X^n = [X_1, X_2, \cdots, X_n]$ into an ascending order $[X_{(1)}, X_{(2)}, \cdots, X_{(n)}]$ and dividing it into L intervals with each interval containing equal number of X_i's, and then pick the boundary points in an ascending order as $q_0, q_1, \cdots, q_L$, that is, $q_0 = X_{(1)}, q_1 = X_{(\lfloor n/L \rfloor)}, \cdots, q_L = X_{(n)}$.

A. Algorithm Description

The universal key generation algorithm is described in the following steps. We suppose that Alice and Bob pre-agree on the quantization level L and the feed-forward data level m.

1) Channel sounding phase: For each X_i and Y_i ($i \in \mathbb{N}_n$), Alice and Bob quantize their estimated channel information into $\log_2(L)$-bit data $K_A(i)$ and $K_B(i)$ using the universal equiprobable scalar quantizer described above. Gray coding is used for mapping the quantizer indices to bits. Notice that, Alice and Bob's quantization codebook might not be the same. After n channel measurements, Alice and Bob obtain an $n \log_2(L)$-bit data $\mathbf{K}_A = [K_A(1), K_A(2), \cdots, K_A(n)]$ and $\mathbf{K}_B = [K_B(1), K_B(2), \cdots, K_B(n)]$.

2) Feed-forwarding phase: Alice generates feed-fowarding data by using the universal equiprobable quantizer to indicate m-level sub-intervals $[q_{i-1}, t_{i-1,1}), [t_{i-1,1}, t_{i-1,2}), \cdots, [t_{i-1,m-1}, q_i)$. Let us index the sub-intervals by $0, 1, \cdots, m-1$ for each sub-interval in an ascending order. Alice then sends an $n \log_2(m)$-bit message, $\mathbf{V}_a = [V_a(1), V_a(2), \cdots, V_a(n)]$ towards Bob through the public channel.

3) Feed-back phase: For each $i \in \mathbb{N}_n$, using the feed-forward data $V_a(i) = v_{ai}$, Bob decides his estimation of Alice's index by $j_i = j \in \mathbb{Z}_L$ if $y_i \in (\bar{q}_j, \bar{q}_{j+1})$, where

$$\bar{q}_j = \begin{cases} \text{median}[t_{j-1,v_{ai}+1}, t_{j,v_{ai}}), & 1 \le j \le L-2, \\ -\infty, & j = 0, \\ \infty, & j = L-1. \end{cases} \tag{9}$$

Here, the median operation of an ordered set $(t_{j-1,v_{ai}+1}, t_{j,v_{ai}})$ as in (9) can be implemented by looking into the ordered sequences $[X_{(1)}, X_{(2)}, \cdots, X_{(n)}]$ and picking the middle point such that it divides interval $(t_{j-1,v_{ai}+1}, t_{j,v_{ai}})$ into two subregions with equal number of sequences. Notice that this is a little different from (8) where a Euclidean middle point of two boundaries of $(t_{j-1,v_{ai}+1}, t_{j,v_{ai}})$ is chosen. Bob then sets $V_b(i) = 1$ if $j_i \neq K_B(i)$ and $V_b(i) = 0$ otherwise, for each $i \in \mathbb{N}_n$, and sends an n-bit feed-back message $\mathbf{V}_b = [V_b(1), V_b(2), \cdots, V_b(n)]$ to Alice through the public channel.

4) Key generation phase: Using the feed-back message $\mathbf{V}_b$, Alice skips the corresponding $K_A(i)$ if $V_b(i) = 1$ (for each $i \in \mathbb{N}_n$), and sets the remaining as her secret key. As for Bob, for each $i \in \mathbb{N}_n$, he skips the indices $K_B(i)$ if $V_b(i) = 1$ and sets it as his secret key.

B. Simulation results

Fig. 5 plots the performance of the universal key generation algorithm described above with 1-bit (i.e., $m = 2$) feed-forwarding (with 10^5 runs). Here, we assume the underlying joint distribution of X and Y is given by (1). To compare the performance, we also present the result of MAP with feed-back algorithm with $m = 2$ as in the previous section. As we can see from the figure, the universal scheme works almost as well as the MAP with feed-back scheme. Fig. 6 shows the performance of the universal key generation algorithm when the underlying source is uniformly distributed and the estimation noise is assumed Gaussian. For this case, the universal scheme outperforms the MAP with feed-back algorithm designed for Gaussian priors as in Section III.

VI. SECURE CODING WITH RATELESS CODES

If a positive secrecy capacity can be established, it is necessary that the transmitter shall transmit at a rate R that

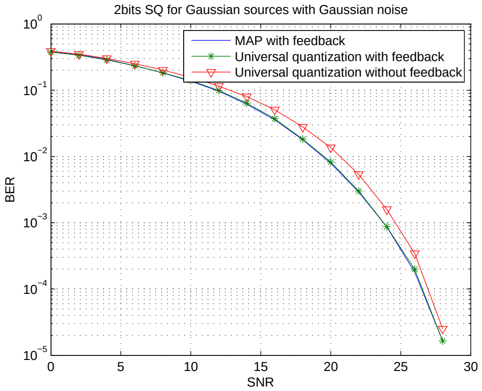


Fig. 5. BER for 2-bits universal quantization for Gaussian source.

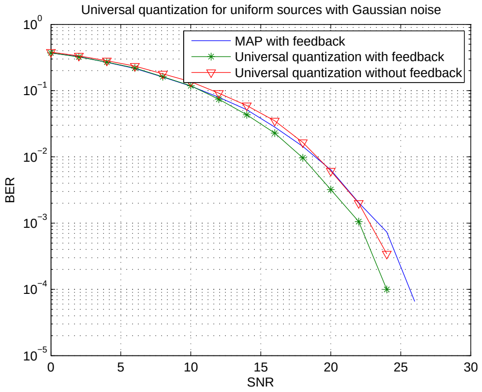


Fig. 6. BER for 2-bits universal quantization for uniform source.

equals the C_{AB} to achieve the allowed maximum secrecy capacity. Unfortunately, it is difficult to estimate accurately C_{AB}, and moreover, the instantaneous capacity $C_{AB}(t)$ varies.

Automatic Repeat Request (ARQ) is generally used to deal with instantaneous channel capacity variation. However, it is inefficient as information from previous transmission is thrown away. Hybrid ARQ (HARQ) using rate-compatible codes combines the original packet and the retransmitted packet(s) and is slightly more efficient. The issue with typical rate compatible code is the large granularity of the effective rates, For instance, a convolutional code may be punctured to obtain rates of $R = 7/8$, $R = 3/4$ and $R = 1/2$. Certain degree of secrecy capacity is sacrificed because of the coarse granularity.

Rateless codes, on the other hand, allows the transmitter to gradually reduce the rate. Combining it with HARQ, we can theoretically match the transmission rate to the exact instantaneous capacity as the number of bits transmitted can be arbitrarily small and we can adjust the rate at a much finer granularity.

A. Channel Scrambling for Positive Secrecy Capacity

To obtain positive secrecy capacity, the transmitter simply scrambles the coded data stream with the key it produced using the scheme discussed above. The received stream is descrambled with the receiver's key. This is similar to a one-time pad encryption. Our scheme, however, does NOT require keys to be exactly the same.

Note that in the case of a perfectly matched key pairs, the output after descrambling is the original received message. Mismatching bits in the keys will result in erroneous bits and the scrambling-descrambling process is equivalent to a binary symmetric channel (BSC), and the key mismatch rate is the crossover rate of the BSC channel. Because the eavesdropper's channel has zero, or low correlation with the channel between Alice and Bob, the crossover rate is much higher, and hence a much worse channel for the eavesdropper.

The artificial BSC channels created by scrambling-descrambling, can be combined with the wireless channels between nodes, or treated separated if we insert an outer channel code. For clarity, we consider the wireless channel noise free in the following discussion.

B. Overview of Rateless Codes

Rateless codes, also known as Fountain codes, are a class of codes with no fixed coding rate. The transmitter can generate a potentially limitless sequence of encoding symbols from a given set of information symbols. The receiver starts to decode after receiving a sufficient number of symbols from the channel. If it fails decoding, it will collect more symbols and start decoding again. The process is repeated until decoding succeeds. The transmitter keeps sending more symbols until receiving an ACK of successful decoding from the receiver. The number of symbols required for successful decoding depends on the quality of the channel. Rateless codes are especially useful for those cases where channel statistics are not known and fixed-rate codes do not work well.

The first class of practical rateless codes, Luby Transform (LT) codes [17], is invented in 1998. LT codes can be represented by Tanner graph and decoded using the belief propagation algorithm, like low-density parity-check (LDPC) codes. Given k information symbols, an encoding symbol of LT codes can be generated by first picking a degree d at random according to some distribution. Then select d distinct information symbols uniformly and XOR them to form an encoding symbol. The transmitter and the receiver should share the same random seed so that the decoder could construct the same code graph as the encoder. Luby shows the Robust Soliton distribution has excellent performance on erasure channel, but the disadvantage is that the decoding complexity is high, i.e., $O(k \ln k)$.

One problem with LT codes is that they exhibit an error floor phenomenon. This can be fixed by Raptor codes proposed by Shokrollahi [18], which combine LT codes with outer LDPC codes. Raptor codes show no noticeable error floors, however their rate is slightly bounded away from capacity.

C. Code Parameters and Decoding Scheme

In this paper, we focus on LT codes with the following degree distribution [18].

$$\begin{aligned}\mu(x) &= 0.007969x + 0.493570x^2 + 0.166220x^3 + 0.072646x^4 \\ &\quad + 0.082558x^5 + 0.056058x^8 + 0.037229x^9 \\ &\quad + 0.055590x^{19} + 0.025023x^{65} + 0.0003135x^{66}. \qquad (10)\end{aligned}$$

The main advantage of this distribution is that its decoding complexity grows only as $O(k)$. However this leads to a small fraction of information symbols that are not involved in any encoding symbols. Then the bit error rate does not go to zero even when k goes to infinity. Nevertheless in practice, LT codes with the degree distribution in (10) still work well.

Decoding LT codes is similar to LDPC codes. Since information symbols are not transmitted, at the decoder side, they do not have any observation, thereby no *a priori* information. Different from information symbols, encoding symbols have *a priori* information, which is denoted by log likelihood ratios (LLRs) [29]:

$$L = \ln \frac{\Pr(s=1|r_c, r_k)}{\Pr(s=0|r_c, r_k)}, \tag{11}$$

where r_c is the observation from the channel and r_k is the observation from the key. An *a priori* information is used for the initialization of decoding. Let the LLRs of received data from the channel be L_c and the LLRs of the key be L_k. Then, we have [29]

$$L = 2\tanh^{-1}\left(\tanh\left(\frac{L_c}{2}\right)\tanh\left(\frac{L_k}{2}\right)\right). \tag{12}$$

If the reliability of key is unknown, then K_b can be treated as error free, i.e., L_k is infinity. For this case, (12) can be rewritten as

$$L = (1-2K_b)L_c, \tag{13}$$

where K_b is the corresponding bit in the key string of the receiver.

D. Simulation Results

We consider a time-variant channel and compare the performance of a fixed-rate LDPC code and an LT code with the degree distribution in (10). In this simulation, we assume the channel between Alice and Bob is a binary symmetric channel (BSC) with crossover probability p. To consider the time-variant property in wireless channels, we assume that p is uniformly distributed over an interval of $[0.01, 0.06]$, and is fixed for every 100 blocks while keep changing during the entire simulation (10^5 blocks).

In each block, 1000 bits are transmitted using either an LT code with or a fixed-rate LDPC code, whose degree distribution is optimized according to [19]. To make fair comparison, the parameters of the time-variant channel are the same for both codes. This can be achieved by saving the values of p in the first round of simulation and use it directly in the second round.

If decoding fails, the scheme using LT codes can request more bits from the transmitter to achieve an arbitrary small probability of decoding error. Hence, a reasonable procedure is to set a retransmission threshold n_t at the decoder side. If the total number of received bits exceeds this threshold and there are still bit errors, the decoder will report a block error.

Fig. 7 shows the simulation result of the two codes, wherein the y-axis is the probability of block error and the x-axis is the inverse of the code rate. For LDPC codes, we set a rate of 0.65, 0.6 or 0.55, with an optimization degree [19]. For LT codes, the rate is calculated using the average code-length in the entire 10^5 blocks. Note that, if a block error is reported, n_t is chosen as the code-length in that block and a block error is collected. In this figure, the red curve corresponds to the case where the LLRs of the key are known while the blue one corresponds to the case without key reliability information. As we can see from the figure, the LT code achieves better performance. The probability of block error converges to zero much faster than the fixed-rate LDPC code. Moreover, key reliability information helps decoding and hence achieves better performance. The same observation can be made in another simulation where p is assumed to be uniformly distributed in $[0.06, 0.11]$ as shown in Fig. 8.

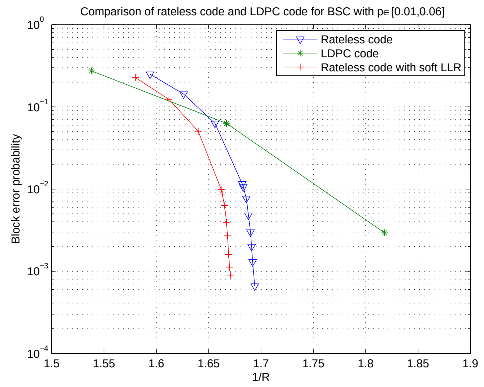


Fig. 7. Comparison of LT code and LDPC code for $p \in [0.01, 0.06]$.

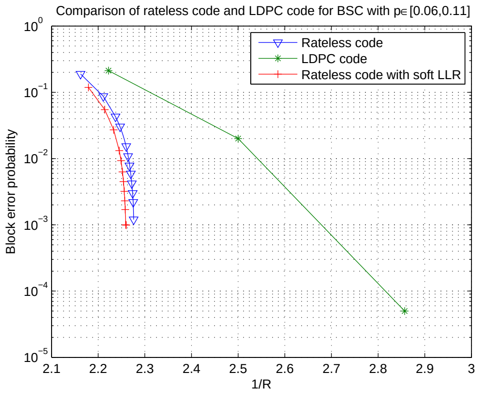


Fig. 8. Comparison of LT code and LDPC code for $p \in [0.06, 0.11]$.

VII. Conclusion

We described a transceiver design which can improve physical-layer security for practical wireless communications. The design combines the channel reciprocity based key generation and rate-adaptive coding. The design does not require a perfectly matched key pair. By scrambling the transmitted data and de-scrambling the received data with the independently generated keys, we guarantee the legitimate receiver to have a preferable channel compared to eavesdroppers, therefore a positive secrecy capacity.

To improve the overall security, we optimized the key generation and secure transmissions separately. The secrecy capacity is maximized by minimizing key mismatch rate between two legitimate nodes. We introduced the notion of feed-forward and feed-back techniques into our quantizer design and proposed an MAP estimator with feed-back quantization scheme that achieves 1 dB improvement over the best known scheme in the high SNR regimes. Moreover, for the cases that the two legitimated nodes do not have any prior information about the underling statistical distribution, we proposed a universal quantization scheme with feed-forward and feed-back information. It has been verified by simulations that the proposed universal scheme can achieve the same performance as algorithms which require prior information.

We then investigated the scheme of applying rateless codes for secure transmissions. Rateless codes can potentially overcome the uncertainty from the key generation as well as channel variation. The performance of the proposed scheme is compared with a fixed-rate LDPC code. It is demonstrated in the simulation that the proposed scheme achieves a significant improvement over LDPC codes.

REFERENCES

[1] U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 733–742, 1993.

[2] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography–part I: secret sharing," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1121—1132, 1993.

[3] A. A. Hassan, W. E. Stark, J. E. Hershey, and S. Chennakeshu, "Cryptographic key agreement for mobile radio," *Elsevier Digital Signal Processing Magazine*, vol. 6, pp. 207–212, 1996.

[4] C. Ye, A. Reznik, and Y. Shah, "Extracting secrecy from jointly Gaussian random variables," *IEEE Int'l Symp. Inf. Theory*, pp. 2593–2597, July 2006.

[5] R. Wilson, D. Tse, and R. A. Scholtz, "Channel identification: secret sharing using reciprocity in UWB channels," *IEEE Trans. Information Forensics and Security*, vol. 2, no. 3, pp. 364–375, Sept. 2007.

[6] A. Sayeed and A. Perrig, "Secure wireless communications: secret keys through multipath," *IEEE Int'l Conf. Acoustic, Speech & Signal Processing (ICASSP)*, pp. 3013–3016, Apr. 2008.

[7] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," *ACM SigMobile Int'l Conf. Mobile Computing and Networking (Mobicom)*, Sept. 2008.

[8] S. Jana, S. P. Nandha, M. Clark, S. K. Kasera, N. Patwari, and S. Krishnamurty, "On the effectiveness of secret key extraction using wireless signal strength in real environments," *ACM SigMobile Int'l Conf. Mobile Computing and Networking (Mobicom)*, Sept. 2009.

[9] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels," *IEEE Trans. Antennas Propagation*, no. 53, pp. 3776–3784, Nov. 2005.

[10] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks," *ACM Conf. Computer and Communications Security*, pp. 401–410, Nov. 2007.

[11] M. A. Tope and J. C. McEachen, "Unconditionally secure communications over fading channels," *Military Communications Conference (MILCOM)*, pp. 54–58, Oct. 2001.

[12] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N. B. Mandayam, "Information-theoretically secret key generation for fading wireless channels," *IEEE Trans. Information Forensics and Security*, vol. 5, no. 2, pp. 240–254, June 2010.

[13] G. D. Durgin, "*Space-Time Wireless Channels*," Prentice Hall, Upper Saddle River, NJ, 2002.

[14] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," *Advances in Cryptology–EUROCRYPT*, pp. 410–423, 1994.

[15] C. H. Bennett, G. Brassard, C. Crepeau, and U. Maurer, "Generalized privacy amplification," *IEEE Trans. Inf. Theory*, vol. 41, no. 6, pp. 1915–1923, Nov. 1995.

[16] J. W. Wallace, C. Chen, and M. A. Jensen, "Key generation exploiting MIMO channel evolution: algorithms and theoretical limits," *European Conf. Antennas and Propagation (EuCAP)*, Berlin, Germany, March 2009.

[17] M. Luby, "LT Codes," *43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002.

[18] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52 , no. 6, pp. 2551–2567, 2006.

[19] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.

[20] R. Moskowitz, "Key management protocols-value, cost, and future proofing", *doc.: 15-10-0877-00-0hip-Turorial-KeyManagementProtocols*, Nov. 8, 2010. Last access Nov., 2010.

[21] A. D. Wyner " The wire-tap channel", *The Bell System Technical Journal*, vol. 54, pp. 1355-1387, Oct. 1975.

[22] I. Csiszár and J. Körner, "Broadcast channels with confidential messages", *IEEE Trans. Inform. Theory*, IT-24, no. 3, pp. 339-348, May 1978.

[23] Y. Liang, H. V. Poor, and S. Shamai (Shitz). "Secure communication over fading channels", *IEEE Transactions on Information Theory*, Special Issue on Information Theoretic Security, 54(6), 2470-2492, June 2008.

[24] Y. Liang and G. Kramer. "Rate regions for relay broadcast channels", *IEEE Transactions on Information Theory*, Special Issue on Models, Theory and Codes for Relaying and Cooperation in Communication Networks, 53(10), 3517-3535, Oct. 2007.

[25] T. Koike-Akino, A. F. Molisch, C. Duan, Z. Tao, and P. Orlik, "Capacity, MSE and secrecy analysis of linear block precoding for distributed antenna systems in multi-user frequency-selective fading channels", *IEEE Transactions on Communications.*, Mar. 2011.

[26] R. Liu, I. Maric, R. D. Yates, and P. Spasojevic, "The discrete memoryless multiple access channel with confidential messages", *IEEE ISIT2006*, Seattle, July 2006.

[27] Pedro C. Pinto, Joao Barros, and Moe Z. Win, "Secure Communication in Stochastic Wireless Networks", *http://arxiv.org/abs/1001.3697*, Jan. 2010. Last access date Oct., 2010.

[28] S. Goel and R. Negi, "Guaranteeing Secrecy using Artificial Noise", IEEE Transactions on Wireless Communications, vol. 7, no. 6, pp. 2180-2189, June 2008

[29] John G. Proakis, "Digital Communications", McGraw-Hill Science/Engineering/Math; 4th edition, 2000

End – to – End Secure Data Delivery in Wireless Sensor Networks

Alexandros Zaharis
University Of Thessaly
Volos, Greece
alzahari@inf.uth.gr

Leonidas Perlepes
University Of Thessaly
Volos, Greece
leperlep@inf.uth.gr

George Stamoulis
University Of Thessaly
Volos, Greece
georges@inf.uth.gr

Panagiotis Kikiras
University Of Thessaly
Volos, Greece
kikirasp@inf.uth.gr

***Abstract* — Typical sensor nodes are resource constrained devices containing user level applications, operating system components, and device drivers in a single address space, with no form of memory protection. A malicious user could easily capture a node and tamper the applications running on it, in order to perform different types of attacks. In this paper, we propose a 3-Tier Security Framework composed by physical security schemes, cryptography of communication channels and live forensics protection techniques that allows for secure WSN deployments. Each of the abovementioned techniques maximizes the security levels leading to a tamper proof sensor node. Even if the physical protection of the nodes is bypassed which is common in wireless sensor attacks, the attacker must surpass 2 more security tiers in order to perform a valid attack on the underlying network. By applying the proposed security framework, secure communication between nodes is guaranteed, identified captured nodes are silenced and their destructive effect on the rest of the network infrastructure is minimized due to the early measures applied. Our main concern is to propose a framework that balances its attributes between robustness, as long as security is concerned and cost effective implementation as far as resources (energy consumption) are concerned.**

Keywords - Security Framework; Sand-boxing; live forensics; cryptography; wireless sensor networks.

I. INTRODUCTION

Wireless Sensor Networks (WSN) are emerging as an important tier in the IT ecosystem where active research involving hardware and system design, networking, distributed algorithms, data management and security, is blended to deal with a unique environment with distinctive characteristics and demands. The main function of a sensor network is the utilization of tiny sensing devices which are capable of sensing various types of incidents/parameters and communicating those with other devices. Sensor networks sensing can be applied for many applications such as target tracking, surveillance, environmental monitoring, etc. [30].

Due to the unattended environment on which wireless sensors operate and the resource constrained nature of these devices in the manner of computational capabilities, memory size and available energy, it is a major challenge to employ efficient security schemes coming from the computers or ad hoc wireless networks domain [2][30].

In this paper, the critical security issues in wireless sensor networks are addressed, various types of threats and attacks against them are explored in order an efficient multi-tier security framework to be proposed.

Furthermore, an evaluation of the combination of cryptography of the communication channel is presented along with valid sand-boxing techniques for providing protection in energy constrained embedded sensor nodes

The three layers of the proposed framework are:

a) Physical Sensor Protection
b) Sand-Boxing
c) Crypto-Communication

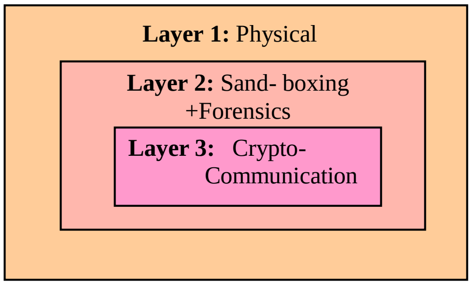

Layer 1: Physical
Layer 2: Sand- boxing +Forensics
Layer 3: Crypto- Communication

Figure 1: Multi layer Security Framework

Framework's primary goal is the effective blending of common security techniques such as physical security or cryptography with more modern ones like sand-boxing [1][2][26][27].

The proposed protection framework is thoroughly presented along with real life use examples that prove its robustness and effectiveness against the most popular WSN security attacks. The overall concept of combining live forensics along with "sand-boxing" techniques and other commonly used security schemes as cryptography in a single framework is, to our knowledge, a unique and out of the box security attempt that can lead to an impenetrable multi-tier security framework.

The remainder of this paper is organized as follows: Section II provides a review of similar security techniques and frameworks. In Section III, we briefly explain the components on which the framework is based on. Section IV describes in details the proposed framework. In Section V, the framework's efficiency against different types on attacks in explained. Finally, Section VI some concluded remarks are presented.

II. Similar Work

Attacks on the sensor network can be classified as:

a) Physical attacks on sensor devices, e.g., destroying, analyzing, and/or reprogramming sensors.
b) Service disruption attacks on routing, localization
c) Data attacks, e.g., traffic capture, spoofing.
d) Resource-consumption and denial-of-service (DoS) attacks.

One of the serious attacks to the sensor networks deployed in an unattended environment is physical tampering with sensors. An adversary can easily capture, reverse-engineer the sensor, and deploy (multiple clones of) manipulated sensors. The compromised sensors will then be exploited by the adversary to mount actual attacks which will facilitate the subversion of the entire network.

Traditionally, the tamper-proofing of programs relies on tamper-resistant hardware [1][2]. However, hardware-based protection will likely fail to provide acceptable security and efficiency on its own because 1) strong tamper-resistance is 'expensive' to be implemented in resource-limited sensor devices and 2) the tamper-resistant hardware itself is not always absolutely safe due to various tampering techniques [1][3][4] such as reverse-engineering on chips, microprobing, glitch and power analysis, and cipher instruction search attacks.

Existing approaches to generating tamper-resistant programs without hardware support can be classified as:

a) Code obfuscation that transforms the executable code to make analysis/modification difficult [5][6][7][8].
b) Result checking that examines the validity of intermediate results produced by the program [9][10][11].
c) Self-decrypting programs that store the encrypted executables and decrypt them before execution [12][13].
d) Self-checking that embeds, in programs, codes for hash computation as well as correct hash values to be invoked to verify the integrity of the program under execution[12][14][15].
e) Software based Attestation to remotely verify the integrity of sensor software [20].

However, most of the above mentioned approaches will more likely fail on sensor networks where a program runs on slow, less-capable microcontrollers.

Software attestation is a challenge-response protocol where a verifier (e.g., base station) sends an attestation command to the attester (the node being attested) asking for certain state information as the evidence of its software integrity. Such state can be computed correctly only if the attester's system meets certain integrity requirement. After receiving the response, the verifier compares it with the known good state to check if the software at the attester has been corrupted. If a sensor node fails to give the correct answer, actions can be taken to revoke this node from the network. Several software attestation schemes have been proposed to attest the static memory regions of the software [17][18][19][20].

Physical hardening of the sensor is the first obstacle an attacker must overcome in order to tamper a wireless sensor. The effectiveness of the physical security on sensors is usually low and a WSN based only on physical security cannot be considered as secure. In our approach physically securing a sensor is the layer of defense mostly used to prevent less determined attackers. Our second defense scheme strips the major functions of a live forensics check on an average system in order to match with the limited resources of a sensor, leading to the important conclusion of whether a sensor is compromised.

The live forensics security layer proposed in this paper verifies the integrity of the program residing in each sensor through a process that has been specifically designed to:

a) Prevent altering / manipulation / reprogramming of the sensor
b) Be purely software-based.
c) Work on sensor devices with severe resource limitations
d) The verification of the different parameters tested does not add large overhead to the communication.
e) Prevent eavesdropping attacks on the communication channel.

III. Live Forensics Framework

As the need for decentralized security emerges in large public wireless sensor networks; new application level security mechanisms aim at providing application developers with appropriate abstractions for designing the security aspects of the target software.

In computer security, a sandbox is a security mechanism for separating running programs. It is often used to execute untested code, or untrusted programs from unverified third-parties, suppliers and/or untrusted users.

It typically provides a tightly-controlled set of resources for guest programs to run in, such as scratch space on disk and memory. In this sense, sandboxes are a specific example of virtualization.

Zaharis et al. [26] proposed a protocol based on sandboxing technique. According to this approach, they divide isolates in two categories:

- The Security-Dedicated Isolates ("SDI")
- The Work-Dedicated Isolates ("WDI")

An Isolate Verification Server plays a key role on verifying the genuine WDIs from the malicious ones while performing all the computational and energy consuming and needy tasks.

ISBN: 978-1-61208-144-1

The verification of genuine WDIs is based on 1) RAM dumbing and 2) Hashing techniques.

This framework uses a secure RAM dumping technique specially designed for sensors. This technique provides the framework with safer intrusion recognition while complying with the classic digital forensic techniques.

The Hashing technique that is used by the framework is based on the Randomized Hashing Function [16]. This technique is used on the Work-Dedicated isolates in order to acquire highly secure tamper-proofing on sensor-resident programs. The hashing function plays a key role in the effectiveness of the proposed architecture as it is robust technique, used frequently in computer security and digital forensics due to its precision in detecting altered code.

Our goal is to improve this mechanism by enriching it with cryptographic procedures, in order to provide a secure end-to-end data delivery framework.

A. *Cryptography*

The Tiny Encryption Algorithm (TEA) is a cryptographic algorithm designed to minimize memory footprint and maximize speed. It is a Feistel type cipher that uses operations from mixed (orthogonal) algebraic groups [31]. Despite its robustness minor extensions have been published in order to present safer encryption results. In this research, we determine the weaknesses and identify the robustness of TEA, XTEA and XXTEA algorithms in wireless sensor networks and implement them in secure framework to harden security during communication [27][28][29].

The conditions must be met in order the algorithm to bet truly "inseparable" are:

- The distribution of keys must have been to all nodes in a secure manner.
- Each message uses a secure, unique key.
- The key generation has become with a truly random cryptographic way

In order to generate a set of unique, truly random keys, we use the Random Number Generator service designed and operated by the University of Trinity [25]. RANDOM.ORG's source of entropy is atmospheric noise. This noise is obtained by tuning a radio to a radio frequency that no one is using. It is then played into a workstation where a program converts it to an 8-bit mono signal at a frequency of 8 KHz. Then the first seven bits are discarded and the remaining bits are gathered together. This stream of bits has very high entropy.

A possible attack on the generator is to broadcast on the frequencies that the RANDOM.ORG radios use in order to affect the generator. However, radio frequency attacks of this type would be difficult for a variety of reasons. First, the frequencies that the radios use are not published, so an attacker would have to broadcast across all frequencies of all bands used for FM and AM broadcasting. Second, this is not an attack that can be launched from anywhere in the world, only reasonably close to the generator. RANDOM.ORG currently has radio receivers in several different countries, which would make it difficult to coordinate this type of attack. Third, if an attacker actually did succeed at broadcasting highly regular signals (e.g., perfect sine waves) at exactly the right frequencies from the right locations, then the RANDOM.ORG real-time statistics would pick up the drop in quality very rapidly , which would raise an alert [25].

IV. Network Architecture

Our sensor network consists of an Isolate Verification Server (IVS) an Isolate Verification Database (IVDB) and numerous sensors which consist of an SDI and one or more WDIs. The Security-Dedicated Isolate (SDI) is the one executed on start up and conducts the forensics check of the second isolate. The 'SDI" is the one responsible for the communication with the Isolate Verification Server (IVS)

The Role of the Isolate Verification Server is:

- To communicate with the SDI of every sensor in its vicinity.
- To update/manage its local IVDB.
- To act as a trusted authentication third party.

For scalability, we let cluster-heads in a cluster-based hierarchical architecture serve as IVSs. This allows each IVS to maintain a local IVDB that stores SDI_IDs of the sensors belonging to its own cluster.

It is undesirable to equip only one IVS in a network as it becomes a single point of failure and the performance bottleneck and so use multiple IVSs can be deployed over the entire network. We assume that there exists a mechanism for a sensor to learn how to discover, and reach, an IVS.

The proposed architecture leads to a decentralized model of sensor protection where its cluster head / IVS is responsible for its sensors. Of course, all this information must be gathered in a master IVS with the total IVDB of the whole WSN.

A. *Sandboxing In Action*

In order to achieve the maximum tampering protection of our sensors, sand-boxing is applied to achieve safety against malicious code execution. While more than one Security-Dedicated Isolates can run on a sensor in our proposed Framework we will use one per sensor.

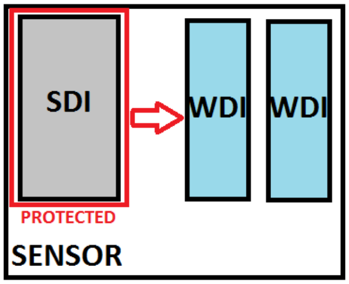


Figure 2: The isolates on a sensor

On the other hand, more than one Work-Dedicated Isolates can run on a sensor performing different tasks. The verification process performed on a single WDI applies for more than one instance with the same results. Failure to verify one of the WDIs leads to locking and blacklisting of the sensors.

1) *Security dedicated isolate*

The Security-Dedicated Isolate is actually a mini forensics tool case specially created to perform live forensics in a sensor, on demand or periodically in order to specify if the sensor is compromised and react depending on the result.

The Security-Dedicated Isolate has a unique id/key for every sensor, the “SDI_ID” that is used in order to communicate with the Isolate Verification Server.

On the first supervised boot the SDI is the first to execute and perform a mini mapping and state validation of the sensor. These results are stored on the Isolate Verification Database (‘IVDB’) which resides on the Isolate Verification Server (‘IVS’). As in every Digital Forensics case these data are going to be used as a proof of the sensors authenticity on the field.

From now on all data transmitted by the SDI are going to be compared to those stored on the ‘IVDB’ depending on the “SDI_ID”.

The tasks the SDI is responsible for are:

a) Communicating safely with IVS.

b) Checking the Work-Dedicated Isolates.

c) Applying countermeasures upon intrusion detection.

2) *Work dedicated isolate*

The Work-Dedicated Isolate performs the everyday tasks of a typical sensor. Due to the sand-boxing technology, more than one WDI can be executed simultaneously on a sensor, performing different tasks. Execution of non verifiable WDIs will lead to the activation of countermeasures by the SDI.

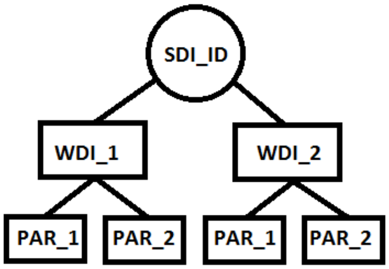


Figure 3: More than one WDI and their check parameters

The fingerprinting of the performance of every isolate on different parameters is stored on Isolate Verification Database along with the SDI_ID of the sensor on which the WDIs belong.

The fingerprinting parameters of a WDI can depend on:

a) The hash value of the isolate.

b) The RAM dump of the isolate.

B. *State-Transition Diagram*

Each sensor device is associated with one of four states:

a) “LOCKED”

b) “VERIFYING”

c) “ACTIVATED”

d) “COUNTERMEASSURES”

When a sensor starts its execution, it is in the LOCKED state. Upon deployment a sensor device will remain in LOCKED state until it securely authenticates with IVS. No other tasks can be performed until it is authenticated.

After a valid authentication, it makes a transition to the VERIFYING state by executing the SDI verification checks. The stripped results are transmitted back to the IVS where: If the verification fails, it returns to the LOCKED state, causing the network to deny this sensor’s access to the network. Otherwise, it transitions to the ACTIVATED state, in which the WDIs code is normally executed. Periodic re-verification by the SDI during ACTIVATED state can lead to LOCKED state or COUNTERMEASSURES state. COUNTERMEASSURES is the state in which a sensor is already accepted on the network and then compromised. In order to avoid denial of service attacks on which the attacker can lead all sensors to LOCK state, the COUNTERMEASSURES state can be used. In this state the compromised sensor tries to identify the type of attack on which it has been subjected through a different type of live forensics process. All other nodes ignore the compromised node through an alarm message send by the IVS. Finally it returns to the LOCKED state.

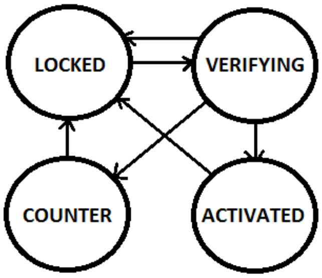


Figure 4: State Diagram

C. *Authentication Protocol*

The proposed protocol is consisted by three phases where certain actions must take place. These phases are divided into actions prior to deployment, during the “initialization” phase and, while in regular operation.

1) *Pre-deployment phase*

During the phase prior to deployment a set of random keys is generated by the base station. This set is stored to the tamper resistant storage area and it is the same for each of the network’s nodes. This set will act as the key repository from where the nodes and the base station will choose their encryption keys during the operational life of the network.

The generation of the keys prior to deployment allows for significant gains in the energy consumed by the nodes, due to the fact that in order to compute a strong cryptographically key, a number of complex mathematical operations and a sequence of iterations are required, which are energy consuming and computational demanding operations.

2) *Initialization phase*

After the deployment of the network the following actions are taking place:

a) Initiation: This step starts the authentication protocol between the IVS and the sensor by transmitting the SDI_ID. The sensor, after receiving the IVS_ID, asks for authentication.
If the authentication fails the protocol is terminated.

b) If authentication succeeds, SDI is executed.

c) The result of SDI is transmitted back to IVS. IVS checks the IVDB and validates the results. The received hash value and Ram dump are checked. If it passes the test, the IVS registers the sensor in the IVDB. Then, the IVS notifies the sensors SDI of the verification result.

d) Based on the verification result, the sensor is either activated or locked. The sensor state will be changed to either ACTIVATED or LOCKED, accordingly.

Step 1 ensures sensor security, i.e., a malicious device can neither passes the authentication procedure nor has its own code executed on the sensor as far as the IVS's authentication key is kept secret from the attacker.

3) *Regular operation*

After the initialization phase, the activated sensors can perform the data's collection, encryption and transmit ion to the base station.

All message transactions, described to the above phases, are encrypted using the XTEA cryptographic algorithm. Each message is encrypted with a key belonged to the set of random keys deployed during the Pre-Deployment Phase of the protocol.

D. *Verification Protocol*

The verification of a sensor is based on two widely used digital forensics techniques 1) Hashing (RHF) and 2) RAM dumping per WDI.

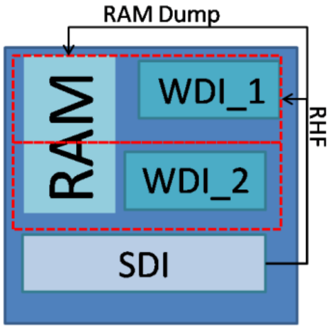


Figure 5: Fingerprinting a WDI

1) *Hashing*

Every Work-Dedicated Isolates has a unique Randomized Hashing Function (RHF)[16] which can be easily and with a minimum cost be calculated. Once calculated for every user it is stored on ISDB along with the SDI of every sensor creating the first fingerprint of the sensor.

Also thanks to the fact that sensors of the same network usually perform the same task can lead to a smaller number of different hash patterns stored on ISDB per WDI.

Each WDI can be classified as being 1) common to all sensors in the network, 2) common to a group of sensors with the same missions, or 3) unique to a specific sensor.

2) *RAM Dump*

When using this technique, our SDI reads arbitrary RAM contents from the different WDIs running on the sensor. Every process running on a system leaves specific, well distinguished footprint on the RAM. Our goal is to create hash like footprint of the memory and store it on ISDB along with the SDI of every sensor creating the second fingerprint of the sensor. In order to keep our framework in energy efficient levels specific parts of the RAM dump are checked concerning the execution of the WDIs.

These WDI –specific fingerprints are also hashed using the Randomized Hashing Function providing an extra protection parameter.

E. *Protocol Implementation*

In order to evaluate our protocol we have implemented it on Mica2 sensor nodes [23]. The MICA2 is a third generation mote module used for enabling low-power, wireless sensor networks. It consists of an ATMega128L CPU, 4kb of Ram, 128kb of program memory and 512kb of serial flash memory and a ChipCon CC1000 radio. The Crossbow MTS310 sensor board was used which provides temperature, and other sensor types.

The protocol is implemented in two parts; the first part corresponds to IVS code (Figure 6) and the second to sensor code (Figure 7).

IVS Code

```
on receive UserHashMsg:
        receive( idAddr , UserHashMsg );
        decrypt(UserHashMsg);
        if( check (UserHashMsg) == valid() ){
                VerifyMsg = Valid;
                encrypt(VerifyMsg);
                send( idAddr , VerifyMsg );
        }
        else{
                VerifyMsg = Invalid;
                encrypt(VerifyMsg);
                send( idAddr , VerifyMsg );
        }

on receive HashMsg:
        receive( idAddr , HashMsg );
        decrypt(HashMsg);
        if( IVDBcheck (HashMsg) == valid() ){
                VerifyMsg = Valid;
                encrypt(VerifyMsg);
                send( idAddr , VerifyMsg );
        }
        else{
                VerifyMsg = Invalid;
                encrypt(VerifyMsg);
                send( idAddr , VerifyMsg );
        }
```

Figure 6: IVS Code

The messages sequence diagram of the aforementioned code implementation can be seen in Figure 8.

```
Sensor Code
on boot:
            state = LOCKED;
            encrypt(UserHashMsg);
            send( broadcastAddr , UserHashMsg );

on receive VerifyMsg:
            receive( idAddr , VerifyMsg );
            decrypt(VerifyMsg);
            if( VerifyMsg==Valid && state==LOCKED){
                        state = VERYFING;
                        HashMsg = computeHashValue();
                        encrypt(HashMsg);
                        send( broadcastAddr , HashMsg  );
            }
            else if( VerifyMsg==Valid && state==VERYFING){
                        state = ACTIVATED;
                        start_data_process();
            }
            else{
                        state = LOCKED;
            }
```

Figure 7: Sensor Code

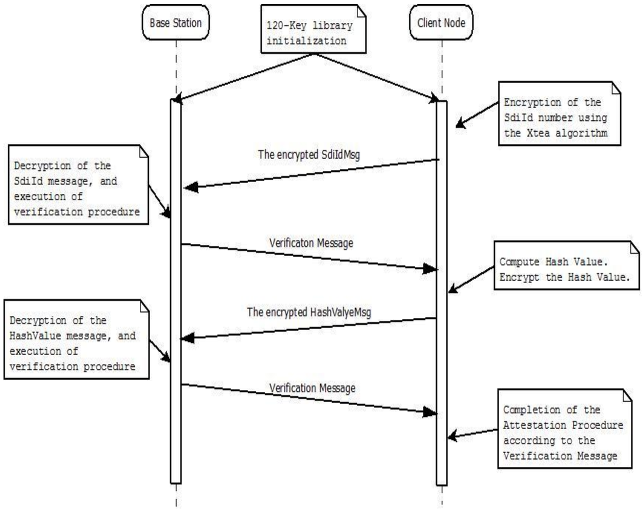

Base Station
120-Key library initialization
Client Node
Encryption of the SdiId number using the Xtea algorithm
The encrypted SdiIdMsg
Decryption of the SdiId message, and execution of verification procedure
Verificaton Message
Compute Hash Value. Encrypt the Hash Value.
The encrypted HashValyeMsg
Decryption of the HashValue message, and execution of verification procedure
Verification Message
Completion of the Attestation Procedure according to the Verification Message

Figure 8: Protocol Messages

a) During the Pre-deployment Phase, each node equipped with a set of secure keys. (120 keys.)

b) The client node encrypts the SDI_ID using the XTea algorithm.

c) The client node sends the 80-bit encrypted message to IVS

d) The IVS decrypts the message and checks the authenticity of the SDI_ID (this step was simulated with 1 sec delay during simulation).

e) IVS sends to the sensor the appropriate response (valid/invalid).

f) On valid response sensors turns from locked to verifying status, initiates the hashing and RAM-dumbing procedure (during the simulation we have used the SHA-1 algorithm (~13ms/hash) [24]. For the calculation of the hash – value the algorithm utilizes 512 bytes from the memory and produces a 160 bit hash.

g) The sensor encrypts the 160-bit hash values, producing a 208-bit message. The sensor transmits the 208bit message to the IVS for validation.

h) IVS verifies the validity of the sensor's hash value. (This step was simulated with 1 sec delay during simulation).

i) IVS validates or the sensor.

j) Sensor turns status into ACTIVATED or LOCKED in accordance with IVS message.

F. Energy Analysis

In order to measure protocol’s energy consumption we have implemented it and simulate its performance in Avrora Simulator. Avrora [22] is a set of simulation and analysis tools for programs written for the AVR microcontroller produced by Atmel and the Mica2 sensor nodes. Avrora contains a flexible framework for simulating and analyzing assembly programs, providing a clean Java API and infrastructure for experimentation, profiling, and analysis. Avrora uses the AOEN (Accurate Prediction of Power Consumption)[21] energy consumer model. AOEN uses empirical current consumption measurements (of hardware such as the radio transceiver, microcontroller and sensors) to calculate the overall power consumption. AOEN is based on the execution of real application and OS code and measurements of node current draw, this model enables accurate prediction of the actual energy consumption of nodes. Thus, it prevents erroneous assumptions on device and network lifetime. Such a detailed prediction allows the comparison of different low power and energy aware approaches in terms of energy efficiency and the estimation of the overall lifetime of a sensor network.

1) Energy cost of cryptography operations.

Table 1 compares the energy consumed by the different versions of TEA cryptography algorithm. The values represent the energy consumed by a node in order to execute the following procedure:

- Encryption and sending of a 64-bit packet
- Receiving and decryption of the 64-bit packet.

The SIMPLE algorithm represents the procedure of sending and receiving the raw packet, without the execution of any cryptographic command.

We do not present the cost of key generation. We assume that the key is created during the pre-deployment phase, as described on section IV.

TABLE 1. ENERGY COST OF TEA CRYPTOGRAPHY ALGORITHMS IN ORDER TO ENCRYPT-SEND/DECRYPT-RECEIVE 64BIT DATA.(MJOULE)

Algorithm	**Energy Cost**	
	Encryption – Send	***Receive - Decryption***
SIMPLE	6041.81μJoule	6002.94 μJoule
TEA	6067.82 μJoule	6030.77 μJoule
XTEA	6071.98 μJoule	6033.19 μJoule
XXTEA	6087.23 μJoule	6042.30 μJoule

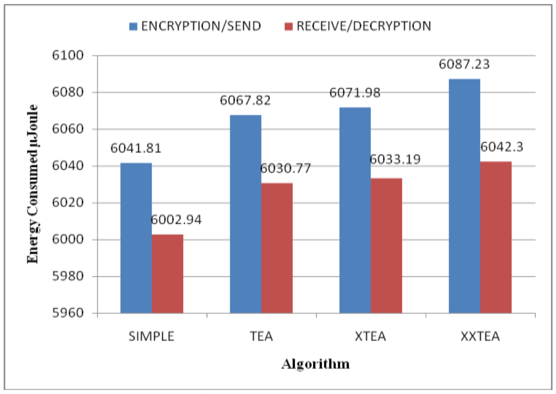


Figure 9: Energy cost of TEA cryptography algorithms. (µJoule)

2) *Energy Cost of Attestation Protocol.*

We analyze the energy usage of the Attestation's protocol handshake procedure. Table 2 compares the energy consumed by the 2 different version of the protocol. The main difference between these versions is the encrypted message transactions that are implemented in the second protocol. As described above, in the Encrypted Attestation Algorithm we the TEA cryptographic algorithm in order to provide an end-to-end secure data delivery protocol. For our analysis we chose to focus on XTEA version of TEA's family cryptographic algorithms.

TABLE 2. ATTESTATION'S PROTOCOL ENERGY COST (JOULE).

Protocol	**Energy Cost**		
	CPU Energy	***Radio Energy***	***Total Energy***
Attestation	0.02286 Joule	0.06034 Joule	0.08321 Joule
Encrypted Attestation	0.02311 Joule	0.06094 Joule	0.08406 Joule

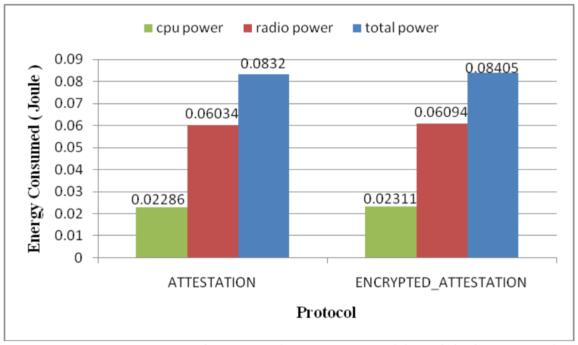


Figure 10: Energy cost of Attestation's protocol handshake procedure. (Joule)

V. SECURITY ANALYSIS

Examples of our proposed framework efficiency against different types on attacks will be displayed in this paragraph along with real life scenarios that prove its robustness.

A. *Physical Attacks*

Physical attacks that can impact the coverage of the WSN and in many cases make the WSN inoperable. Because of the widespread placement of the individual nodes in an often non-secure and unmonitored area, individual nodes are subject to capture. Physical hardening of the sensors against reverse-engineering on chips, microprobing, glitch and power analysis, and cipher instruction search attacks on the first layer of security of the proposed framework can lead to the needed results.

B. *Replay Attacks*

Replay attacks (i.e., intercepting a message and replacing it with an old message) cannot succeed as the proposed hash computation and verification are keyed operations that can be defeated as following: First, reporting a different SDI_ID will be detected by IVS when its uniqueness is checked and, moreover, the malicious sensor will not be able to pass the hash of RAM dump test unless it has the matching program which must be free of malicious codes and created an exact fingerprint. Second, modifying the Hash algorithm will cause inconsistency between two hash outputs and, hence, the verification will fail. Encryption of the communication channel makes it more difficult for an attacker to forge messages, as the messages have to be encrypted with the appropriate secret key.

C. *Forgery Attacks*

We will now show that it is impossible for the adversary to forge the hash value without the knowledge of all the specific parameters previously described, for each WDI. Consider the situation where the adversary reprograms the sensor with a malicious program and attempts to fake the verification process by nullifying the effect of the output of the Hash algorithm. This is impossible because the Hash algorithm is inherently a nonlinear function of program blocks. Thus it is impossible to create a malicious WDI that has the same RHF as the original. Encryption of the communication channel makes it impossible for an attacker to forge the communication between two nodes as unique keys are distributed to every node.

So, what can prevent an attacker from capturing and reverse-engineering a sensor, and using the same sand-boxing technique to keep a good copy of the sensor running in order to feed good answers to the challenge-response protocol initiated by the verifier?

The attacker will not be able to manipulate the sending data of the captured sensor by internal means as changes will be detected by the present SDI. Any attempt to copy only the WDIs will fail as no authenticated messages are going to be sent to the authentication authority leading to rendering the sensor useless. Any attempt to copy both images SDI and WDI in a different sensor or a more resource efficient device will lead in creating different Ram dumps, both in size and structure leading to non verification of the WDI and locking the sensor out of the network. The only way you can copy both SDI and WDI in a different device and get valid results is by copying them in the exact sensor model (hardware and software) thus leading to forensically sound proof results.

D. *Hardware Tampering Attacks*

If the malicious sensor has enough memory to maintain the original program blocks some of the previously stated attacks can succeed. However, as it has been previously defined upon initiation of our Framework a specific fingerprint of both the hash value of the WDIs and RAM dump has been stored.

Therefore, there is no room left in the sensor for the adversary to save and execute arbitrary code. The adversary may attach more memory to each sensor, but it will incur a considerable amount of hardware modification while the Ram dump check will identify the attack. Moving/Copying the isolates in different sensors, as far as hardware specs is concerned, or a personal computer will lead to rendering the sensor useless because of the RAM dump check inconsistencies.

E. *Encryption Algorithms*

TEA operates on 64-bit blocks and uses a 128-bit key. It has a Feistel structure with a suggested 64 rounds, typically implemented in pairs termed *cycles*. It has an extremely simple key schedule, mixing all of the key material in exactly the same way for each cycle.

TEA has a few weaknesses. Most notably, it suffers from equivalent keys—each key is equivalent to three others, which means that the effective key size is only 126 bits. TEA is also susceptible to a related-key attack which requires 223 chosen plaintexts under a related-key pair, with 232 time complexity.

Because of these weaknesses, we chose to focus on XTEA version of TEA's family cryptographic algorithms. XTEA is a 64-bit block Feistel network with a 128-bit key and a suggested 64 rounds. Several differences from TEA are apparent, including a somewhat more complex key-schedule and a rearrangement of the shifts, XORs, and additions.

Also, a third version **Corrected Block TEA** (often referred to as **XXTEA**) was designed, in order to correct weaknesses of the other previous two versions.

Our implementation is based on XTEA cryptographic algorithm, because it is more secure than the TEA and is less energy–harvesting than the XXTEA version, as described on table 1.

VI. Conclusion

In this paper, we have proposed a complete tamper-proofing framework based on physical security schemes, encryption, digital forensics and sand-boxing techniques which offer 1) prevention of manipulation, reverse-engineering, and reprogramming of sensors; 2) purely software based protection with/without tamper-resistant hardware; and 3) infrequent triggering of the verification.

Through securely executed isolates a verification of the Integrity of the program of each sensor device is performed successfully. For verification, it remotely calculates, 1) hash value of every WDI being executed, 2) RAM dumps and checks if the values match with those stored on IVDB depending on the SDI_ID. All communication is through encrypted channels.

Our security analysis has proven that the proposed framework effectively defeats different types of attacks while improving the state of the art in software based protection mechanisms, furthermore from the simulations conducted the protocol has proven to be low

References

[1] R. Anderson and M. Kuhn, “Tamper Resistance—A Cautionary Note,” Proc. Second USENIX Workshop Electronic Commerce, pp. 1-11, 1996.

[2] D.W. Carman, P.S. Kruus, and B.J. Matt, “Constraints and Approaches for Distributed Sensor Network Security,” NAI Labs Technical Report, vol. 00, no. 010, Sept. 2000.

[3] R. Anderson, “Why Cryptosystems Fail,” Comm. ACM, vol. 37, no. 11, Nov. 1994.

[4] S. Blythe, B. Fraboni, S. Lall, H. Ahmed, and U. Riu, “Layout Reconstruction of Complex Silicon Chips,” IEEE J. Solid-State Circuits, vol. 28, no. 2, pp. 138-145, Feb. 1993.

[5] C. Collberg, C. Thomborson, and D. Low, “Breaking Abstractions and Unstructuring Data Structures,” Proc. IEEE Int'l Conf. Computer Languages (ICCL '98), pp. 28-38, May 1998.

[6] C. Wang, J. Hill, J. Knight, and J. Davidson, “Software Tamper Resistance: Obstructing Static Analysis of Programs,” technical report, Dept. of Computer Science, Univ. of Virginia, 2000.

[7] C. Wang, J. Hill, J. Knight, and J. Davidson, “Protection of Software-Based Survivability Mechanisms,” Proc. Int'l Conf. Dependable Systems and Networks, pp. 193-202, July 2001.

[8] G. Wroblewski, “General Method of Program Code Obfuscation,” Proc. Int'l Conf. Software Eng. Research and Practice (SERP), June 2002.

[9] M. Blum and S. Kannan, “Designing Programs that Check Their Work,” J. ACM, vol. 42, no. 1, pp. 269-291, 1995.

[10] H. Wasserman and M. Blum, “Software Reliability via Run-Time Result-Checking,” J. ACM, vol. 44, no. 6, pp. 826-849, 1997.

[11] F. Ergun, S. Kannan, S.R. Kumar, R. Rubinfeld, and M. Vishwanathan, “Spot-Checkers,” Proc. ACM Symp. Theory of Computing (STOC '98), pp. 717-751,May 1998.

[12] D. Aucsmith, “Tamper Resistant Software: An Implementation,” Information Hiding, pp. 317-333, Springer-Verlag, 1996.

[13] C.S. Collberg and C. Thomborson, “Watermarking, Tamper-Proofing, and Obfuscation—Tools for Software Protection,” IEEE, Trans. Software Eng., vol. 28, no. 8, pp. 735-746, Aug. 2002.

[14] B. Horne, L. Matheson, C. Sheehan, and R.E. Tarjan, “Dynamic Self-Checking Techniques for Improved Tamper Resistance,” Proc. First ACM Workshop Digital Rights Management (DRM), pp. 141-159, 2002, 308 IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 4, NO. 3, MAY/JUNE 2005

[15] H. Chang and M.J. Atallah, “Protecting Software Code by Guards,” Proc. Second ACM Workshop Digital Rights Management (DRM), pp. 160-175, 2002.

[16] Taejoon Park, Kang G. Shin, “Soft Tamper-Proofing via Program Integrity Verification in Wireless Sensor Networks”, IEEE Transactions on mobile computing, VOL. 4, NO. 3, pp. 297-309, May/June 2005.

[17] Squawk Project, http://labs.oracle.com/projects/squawk/ (Accessed 4 April 2011)

[18] A. Seshadri, A. Perrig, L. van Doorn, and P. Khosla. SWATT: Software-based ATTestation for Embedded Devices. In IEEE Symposium on Security and Privacy, pp. 272-282, May 2004.

[19] M. Shaneck, K. Mahadevan, V. Kher, and Y. Kim. Remote software-based attestation for wireless sensors. In *Proceedings* of the 2nd European Workshop on Security and Privacy in Ad Hoc and Sensor Networks, pp. 27-41, 2005.

[20] Y. Yang, X.Wang, S. Zhu, and G. Cao. Distributed softwarebased attestation for node compromise detection in sensor networks. In Proceedings of the 26th IEEE International Symposium on Reliable Distributed Systems, pp. 219-230, 2007.]

[21] O. Landsiedel, K. Wehrle, and S. Gotz, “Accurate prediction of power consumption in sensor networks,” in Proc. 2nd IEEE Workshop on Embedded Networked Sensors (EmNetS-II). IEEE Computer Society, 2005, pp. 37–44.

[22] B. L. Titzer, D. K. Lee, and J. Palsberg, “Avrora: scalable sensor network simulation with precise timing,” in Proc. 4th Int’l Conf. Information Processing Sensor Networks (IPSN ’05), 2005, p. 67.

[23] http://www.xbow.com (Accessed 10 April 2011).

[24] NIST, “Digital hash standard,” Federal Information Processing Standards Publication 180-1, April 1995

[25] L. Foley, S. Wilson, "Analysis of an On-line Random Number Generator", Trinity College Dublin, http://www.random.org, (Accessed 8 April 2011).

[26] A. Zaharis, A. I. Martini, L. Perlepes, G. Stamoulis, and P. Kikiras. 2010. Live forensics framework for wireless sensor nodes using sandboxing. In *Proceedings of the 6th ACM workshop on QoS and security for wireless and mobile networks* (Q2SWinet '10), pp. 70-77

[27] D. Wheeler and R. Needham."TEA, a Tiny Encryption Algorithm." 1995. Springer-Verlag.

[28] S. Hong, D. Hong, Y. Ko, D. Chang, W. Lee, and S. Lee. "Differential cryptanalysis of TEA and XTEA." In *Proceedings of ICISC 2003*, pp. 402-417, 2003b.

[29] E. Yarrkov. Cryptanalysis of xxtea. Cryptology ePrint Archive, Report 2010/254, 2010. http://eprint.iacr.org/ (Accessed 27 May 2011).

[30] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. 2002. Wireless sensor networks: a survey. *Comput. Netw.* 38, 4 (March 2002), pp. 393-422. DOI=10.1016/S1389-1286(01)00302-4

[31] Feistel H. "Cryptography and Computer Privacy", Scientific American Vol. 228 No. 5, pp. 15-23, 1973

Time-domain Feature Extraction and Neural Network Identification of Structure Crack Based on Surface-mounted Active Sensing Network

Chunling Du, Jianqiang Mou, Landong Martua, Shudong Liu, Bingjin Chen,
Guopeng Cao, Wen Xiang Yock, Jingliang Zhang
*A*STAR, Data Storage Institute(DSI)*
5 Engineering Drive 1,
Singapore 117608
Email: DU_Chunling@dsi.a-star.edu.sg

Frank L. Lewis
Automation and Robotics
Research Institute,
University of Texas,
Arlington, TX76019, USA

***Abstract*—In this work, the condition of a metallic structure is classified based on the acquired sensor data from a surface-mounted piezoelectric sensor/actuator network. The structure under consideration is an aluminum plate with riveted holes and possible crack damage in these holes is investigated. The sensor/actuator network uses diagnostic signals injected to piezoelectric actuators and received sensor signals to detect the crack. The damage classification system consists of three major components: sensitive signal acquisition, principal feature extraction and damage classification. An appropriate sine wave burst is used as diagnostic signals for actuators to transmit to sensors in order to detect the integrity of the structure. The combination of time-domain S0 waves from all sensitive sensor signals is directly used as features to detect damage. Since the time sequence of the extracted S0 waves is selected as the feature and has a high dimension, principal component estimation is applied to reduce the data dimension before entering the neural network training. Finally, in structure condition classification, a LVQ (learning vector quantization) neural network is used to classify structure conditions as healthy or damaged. In this paper, a number of FEM (finite element modeling) simulation results of sensor signals are taken as inputs to the neural network for training, since it is found that the FEM results have a good agreement with the experimental testing results on real plates. The performance of the classification is then validated by using these testing results.**

***Keywords*-active sensing; damage classification; feature extraction; finite element modeling; Lamb wave; principal component analysis; structure health monitoring; sensor network.**

I. Introduction

Early detection of damages such as cracks in metallic structures due to cyclic loads and environmental corrosion is critical for preventing catastrophic failure and prolonging the life of aircraft structures. To reduce the cost of maintenance, structural health monitoring (SHM) has been proposed as an alternative approach to replace traditional time-consuming inspection for maintenance. The wave propagation method for structural health monitoring has been demonstrated to be effective in detecting debonding in composites and cracks in metallic structures. Therefore, in this work, wave propagation method based on surface-mounted piezoelectric sensor arrays is adopted to monitor crack growth at riveted holes.

In general, the structural health monitoring system consists of two major parts: hardware and software. The hardware includes the distributed sensor network and the data acquisition system. In this work, the active sensing network with actuators excited by known diagnosis signals is adopted. And the software part for diagnosis includes signal analysis for sensitive feature extraction and intelligent algorithms for damage interpretation and classification, which actually result in the physical condition of a given structure from the raw sensor measurements. Our research reported in this paper is concentrated on feature extraction and structure condition classification.

Neural network methods are widely used to solve classification problem, since when properly trained they easily map the extracted feature space to structure condition space. Designing an effective neural network has always been a challenging task. In [6], a three-layer propabilistic neural network is applied to classify the sensor data into several categories relative to the damage location in the circular plates using resonance frequency shifts of E/M (electro/mechanical) impedance as damage features. In [7], a backpropagation neural network is trained to categorize cracks according to their lengths using FE modeling data for scattering of ultrasound by the cracks emanating from rivet holes in a thin aluminum plate. An impedance-based damage detection combined with a backpropagarion neural network is developed in [8] to locate and identify the structure damages. In [9], with independent component analysis for vibration features, a multi-layer perceptron neural network, trained using an error backpropagation algorithm, is able to detect the undamaged and damaged states with very good accuracy and repeatability.

Lamb wave is much more sensitive to structure damages than other structural responses such as modal shape, natural frequency, etc., and the artificial neural network technique based on Lamb wave testing is able to lead to precise identification of the damages. In [10], the authors developed an identification technique for debonding in ad-

ISBN: 978-1-61208-144-1

hesively bonded joints using Lamb wave signals interpreted by neural network training. In [11], different crack lengths at four locations in a PVC sandwich panel were numerically simulated and the percentage shifts in structural energy were used to train a backpropagation neural network. Good identification precision was achieved for another numerically simulated damage cases, while poor precision was attained using measurement signals. In [12], the so-called DDFs (digital damage fingerprints) are extracted from the spectrographic characteristics of Lamb wave signals and serve as damage features. Various numerical simulation results are employed to train the NN (neural network) and then experimentally validated by identifying cylinder through-holes and delamination in the composite laminates. In most of these papers, the artificial NN employs the method of supervised feedforward backpropagation.

This paper presents a method based on neural network for the classification of an aluminum plate with and without crack damage at reveted holes. The time-domain sensor signals are directly used as damage features, and key features having reduced data size are extracted after principal component estimation. The estimated key features are then considered as the input at the neural network, which is trained according to the learning vector quantization method. The NN training uses FEM (finite element modeling) data of Lamb wave propagation in the active sensor network system, and the classification performance is evaluated through the validation using the experimental testing data.

II. Active sensing system for structure crack detection

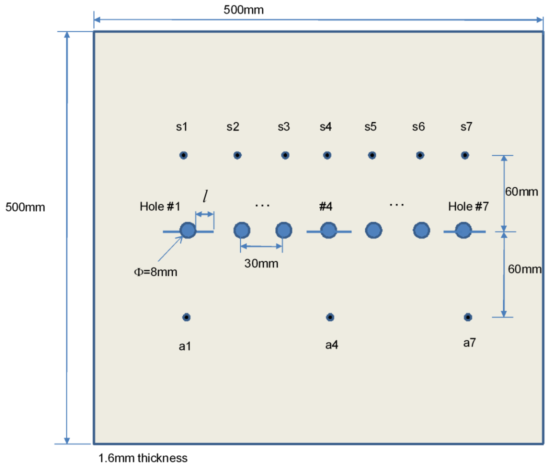


Figure 1. Active sensing system for an aluminum plate with riveted holes with cracks.

The aluminum plate with riveted holes illustrated in Fig. 1 is studied on crack detection using the active sensing system, which consists of sensors denoted as s1, s2, · · ·, s7, and actuators denoted as a1, a2, · · ·, a7. The cracks in the hole are indicated in Fig. 1 with length *l*. The actuators excited by the diagnosis signal, which is here the windowed sinewave burst as shown in Fig. 2, transmit the signal to the sensors. The received signal on the sensor contains the information about the integrity of the structure between the actuator and the sensor, and will be used to detect any crack occurrence by investigating any change in the received signal. Based on the analysis of the sensor signals, information can be retrieved concerning the extent of the damage and used to assess and classify the health condition of the structure.

In this work, we take actuator a4 as one case which is excited by the diagnosis signal in Fig. 2. Signal propagation is simulated with FEM method, and will be verified by testing on real plates.

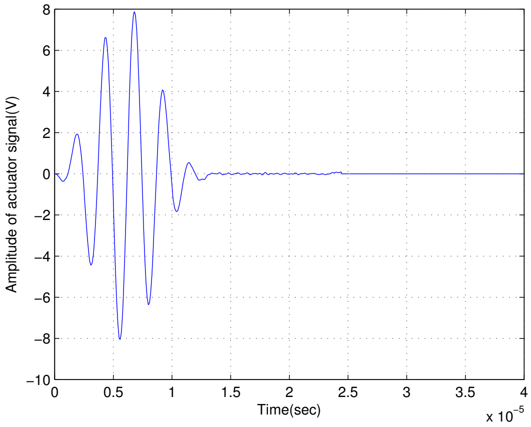


Figure 2. 400 kHz sinewave burst injected to the actuator in simulation.

A. FEM simulation

The FEM result of sensor signals for the pristine plate is shown in Fig. 3. The fundamental symmetric (S0) mode is widely used to detect surface crack in metallic structures due to its sensitivity to crack growth [1] and thus in this work it is utilized for crack detection. As indicated in Fig. 3, S0 wave is obviously fetched by the sensors.

When the plate has a crack with *l*=6mm at the hole #4, the signals of sensors 1 to 4 are shown in Fig. 12. It is obviously noticed that the S0 wave amplitude of sensor 4 is reduced compared with that of the pristine plate. In this case, because sensors s5, s6 and s7 are symmetric to sensors s3, s2 and s1, their signals are respectively the same as those of sensors s3, s2 and s1 and thus omitted here.

With different crack lengths, the maximum amplitude of S0 wave of each sensor signal is plotted in Fig. 5. As expected, sensor s4 is most sensitive to crack at hole #4, which is proved from Fig. 5 as the curve corresponding to s4 drops most significantly even for a small crack *l*=2mm.

ISBN: 978-1-61208-144-1

Other sensors s2 and s3 are sensitive to bigger cracks, while s1 signal does not change much and thus it is not able to detect the crack at hole #4.

Simulation results of sensor signals for cracks located respectively at holes # 3, 2 and 1 have been obtained. The S0 wave maximum amplitude versus different crack lengths is displayed in Fig.s 6-8. In Fig. 6, s3 and s2 are most sensitive to cracks, and s1 is also possibly useful to detect crack. In Fig. 7 for the crack at hole #2, s1 and s2 are able to detect it. However, for crack at hole #1, none of the four sensors is capable of detecting the crack. This implies that excitation to an actuator closer to it is required. In this work, we study the case of actuator a4 excited. As for other actuators, the processing to sensor signals is similar.

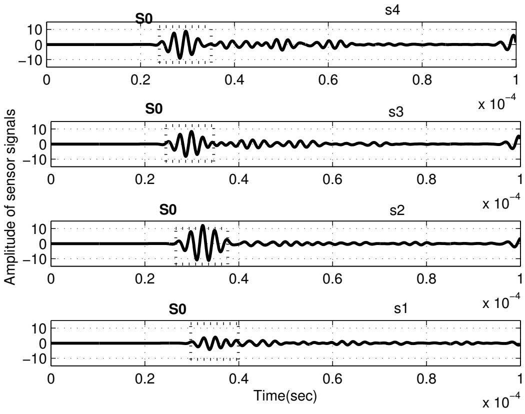


Figure 3. Sensor signals for the pristine plate.

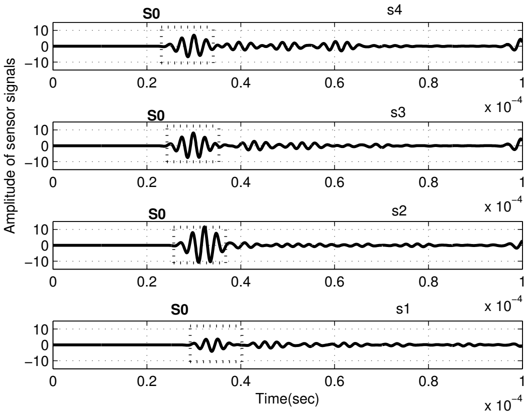


Figure 4. Sensor signals for the plate with crack *l*=6mm at hole#4.

The above analysis shows that the sensors are all useful to crack detection. It is also noticed that S0 wave amplitude as well as its time delay (or, time of flight [2]) relative to actuator signal are sensitive to the crack. Therefore the time

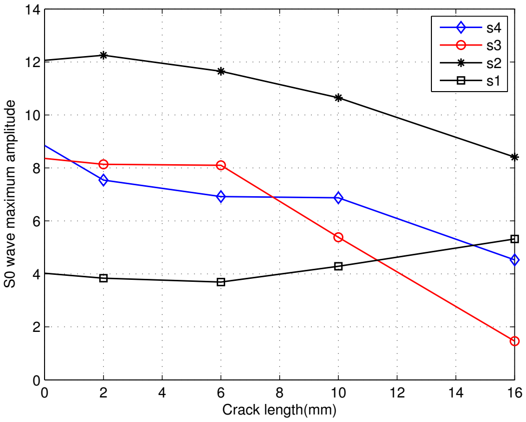


Figure 5. Maximum amplitude of S0 wave versus crack length for crack at hole #4.

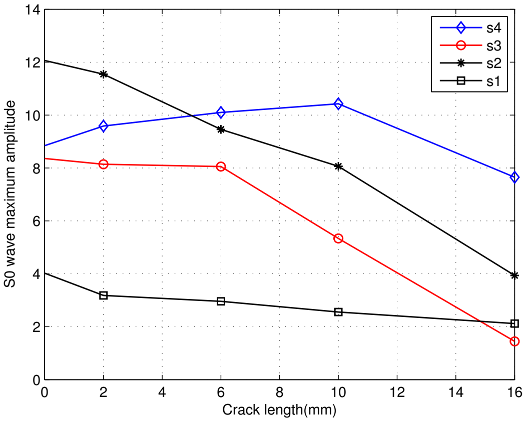


Figure 6. Maximum amplitude of S0 wave versus crack length for crack at hole #3.

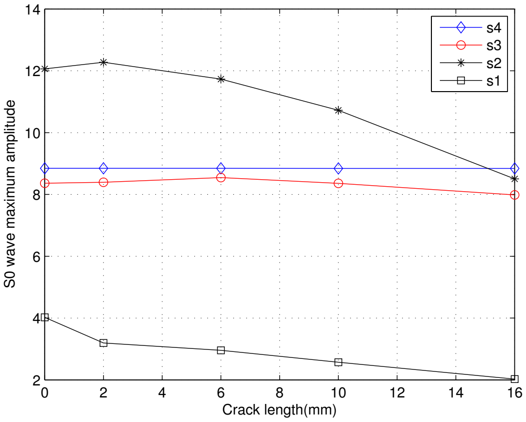


Figure 7. Maximum amplitude of S0 wave versus crack length for crack at hole #2.

ISBN: 978-1-61208-144-1

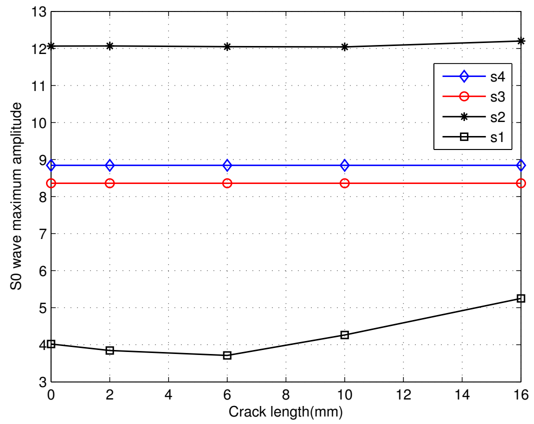


Figure 8. Maximum amplitude of S0 wave versus crack length for crack at hole #1.

sequences of S0 waves of sensors s1-s4 will be used as features for classification in this paper. Take the crack at hole #4 as an example. The combined S0 wave time sequence of sensors s1-s4 is shown in Fig. 9.

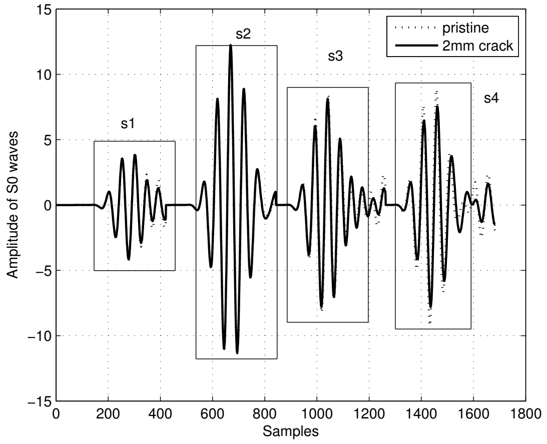


Figure 9. S0 waves for crack at hole #4.

B. Experimental verification

The experimental setup consists of an Arbitrary Wave Generator (AWG) 2041, PZT driver, sensor signal amplifiers, and a programmable GAGE Compuscope 82G card connected to a PC running Labwindows/CVI. The actuation and the sensor signals are amplified by high bandwidth amplifiers. The data acquisition subsystem includes a PCI interface controlled with the PC running Labwindows/CVI, and the data are saved in PC through the GAGE card. The schematic diagram of the experimental setup is shown in Fig. 10.

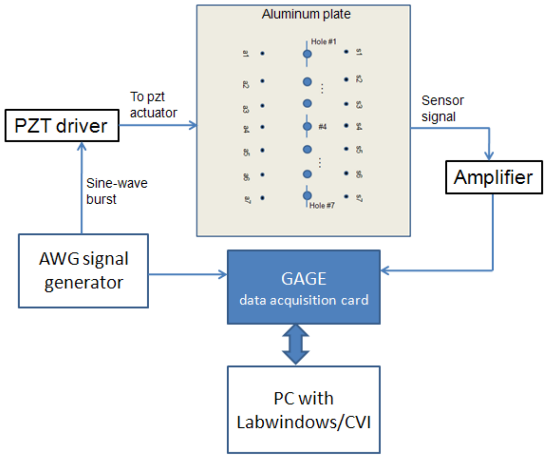


Figure 10. Experimental setup.

Experimental testing on a real plate has been conducted. Circular transducers as sensors and actuators are mounted to the surface of the plate by using epoxy. Excitation signal to the actuator is the same as that in Fig. 2. The sensor signal is collected via the data acquisition card with 20MHz sampling rate. For the pristine plate, sensor signals are displayed in Fig. 11, which agrees well with the simulation results in Fig. 3. We take the plate with crack *l*=6mm at hole#4 as one example with crack. Fig.s 12-13 show the simulation and experimental results, where it is observed that the S0 mode agrees quite well to each other.

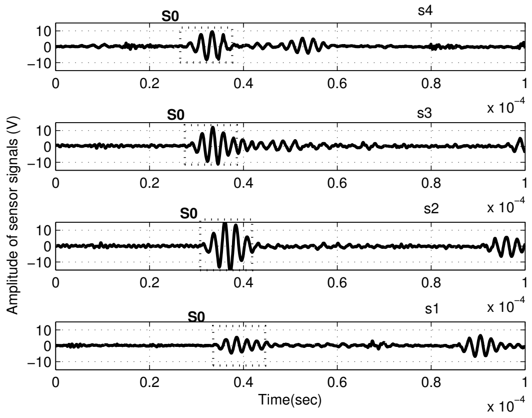


Figure 11. Experimental sensor signals for the pristine plate.

Since the FEM result agrees well with the experiment data, in the next section the FEM results together with some of the experiment data will be used in the neural network training for damage classification.

III. Classification Architecture

This section presents the system architecture being used for the crack classification methodology. The key steps

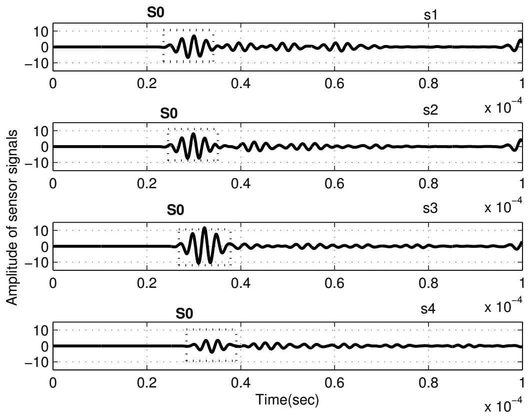


Figure 12. Simulated sensor signals for the plate with crack *l*=6mm at hole#4.

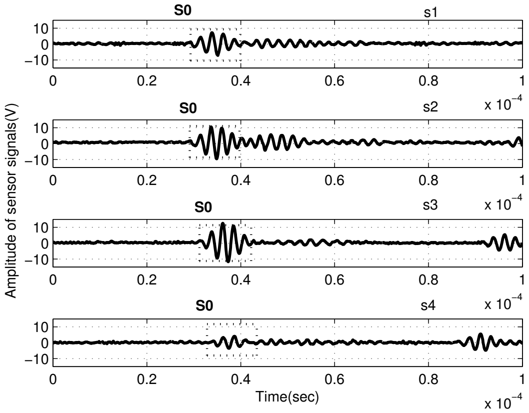


Figure 13. Experimental sensor signals for the plate with crack *l*=6mm at hole#4.

involved in the methodology are 1) S0 wave extraction, 2) Estimation of principal components, 3) Structure condition classification using a neural network.

The proposed approach makes use of an architecture that consists of a neural network to classify different structure conditions. The scheme employed in the approach is shown in Fig. 14, and is detailed in subsequent sections and outlined below.

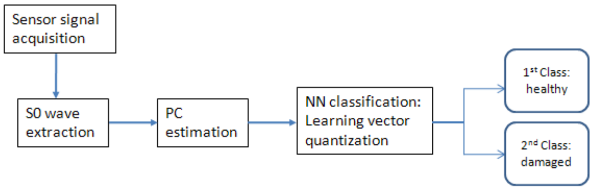

Sensor signal acquisition
S0 wave extraction
PC estimation
NN classification: Learning vector quantization
1st Class: healthy
2nd Class: damaged

Figure 14. Classification architecture.

The first step of the methodology involves the extraction of S0 wave. The time sequences of S0 waves from several sensors are used as inputs to the stage of principal component estimation. This step directly uses the time-domain signal of the sensors, which reduces the processing time for the sensor signals.

The second step given in the proposed approach involves the dimension reduction of the sensor signal using Principal Component Estimation. Principal component analysis (PCA) is a statistical technique used for data compression by determining a linear transformation matrix $W \in R^{m \times n}$ $(m < n)$. The data $X \in R^{n \times 1}$ is compressed and a lower dimension data $y \in R^{m \times 1}$ is yielded and given by

$$y = WX. \qquad (1)$$

The PCA technique is to reduce the number of features representing a data by discarding the ones which have small variance and retains only those that have large variance. It uses singular value decomposition method in calculating the eigenvectors of the co-variance matrix formed by analyzing the sensor data. Only those eigenvectors are selected which give the maximum information about the data. These chosen eigenvectors form the matrix W.

The last and the main step of classifying structure conditions are obtained from the methodology of neural network. The neural network is based on Learning Vector Quantization (LVQ) nets. The LVQ network is trained using Kohonen learning rule to classify structure conditions.

IV. Classification performance analysis

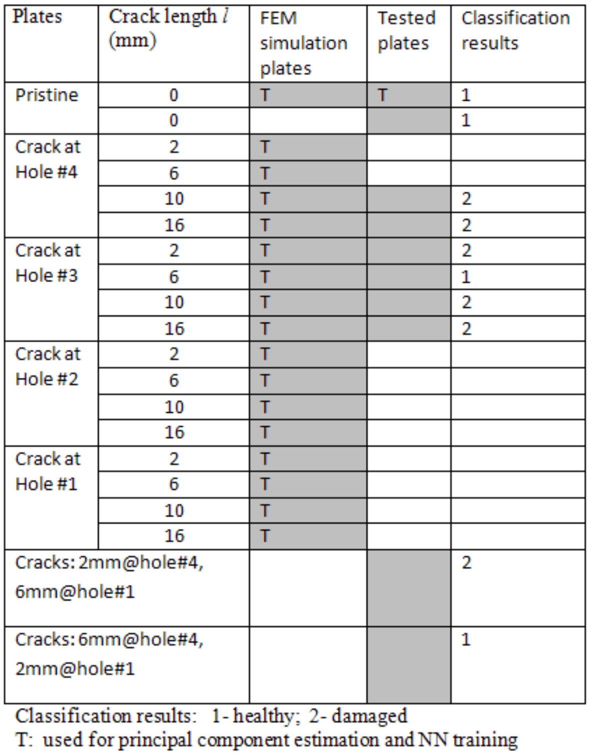
The time domain data of S0 waves from sensors 1 to 4 discussed in Section II is used as feature to classify the cases into categories relative to the structure condition. In this paper, two classes are assigned: 1 representing no damage, and 2 representing damage with crack.

As shown in Table 1, the shaded parts are the types of plates under investigation. It is shown that the data used for principal component estimation and neural network training are mostly coming from FEM results. Only one case is from a real specimen as the pristine plate. For principal component estimation and neural network training, these data are expanded as 1019 pristine plate and 736 damaged plates, which means the sensor data for pristine plate is repeatedly used for 1019 times, and the sensor data for each damaged case is repeatedly used for 46 times. The dimension of S0 wave combination from the four sensors is 1684. After principal component estimation, the dimension is reduced from 1684 to 13.

The used LVQ neural network ('newlvq' function in Matlab) is a a two-layer network. The first layer is a competitive layer that uses the compet transfer function and calculates the distance from an input to each row of the input weight matrix. The second layer is a linear layer having purelin neurons. In this application, the number of hidden neuron of the first layer is 4 and the class percentages are 45% and 55%.

As shown in Table 1, testing results from 10 real plates (indicated by the shadow) are used to verify the classification method. Based on the trained LVQ neural network, 8 plates are classified correctly, and 2 plates are not classified correctly. It is noted the NN training is mainly based on the FEM simulation results. The trained NN leads to such a classification results for the tested specimens is acceptable and promising. It is also noticed that the trained NN is potentially able to identify combined cracks, although it is solely based on the single crack cases.

Table 1. Health classification with sensor data from s1-s4.



Plates	Crack length l (mm)	FEM simulation plates	Tested plates	Classification results
Pristine	0	T	T ▨	1
	0		▨	1
Crack at Hole #4	2	T		
	6	T		
	10	T	▨	2
	16	T	▨	2
Crack at Hole #3	2	T	▨	2
	6	T	▨	1
	10	T	▨	2
	16	T	▨	2
Crack at Hole #2	2	T		
	6	T		
	10	T		
	16	T		
Crack at Hole #1	2	T		
	6	T		
	10	T		
	16	T		
Cracks: 2mm@hole#4, 6mm@hole#1			▨	2
Cracks: 6mm@hole#4, 2mm@hole#1			▨	1

Classification results: 1- healthy; 2- damaged
T: used for principal component estimation and NN training

V. Conclusion

In this paper, the aluminum plates with the riveted holes and possible crack damage at these holes have been studied and the surface-mounted piezoelectric sensor/actuator network has been utilized to detect the crack. The 400 kHz sine wave burst has been used as diagnostic signals and injected to actuators and it propagates to sensors in order to detect the integrity of the structure. The combination of time-domain S0 waves from all sensitive sensor signals has been directly used as features to detect the crack damage. After the principal component estimation, the reduced-size data work as input for the LVQ neural network training. The neural network training has utilized a series of FEM simulation results, since it has been found that the FEM results have a good agreement with the experimental testing results on real plates. The performance of the classification has been finally validated by using the testing results from 10 real plates.

References

[1] J. -B. Ihn, and F. -K. Chang, "Detection and monitoring of hidden fatigue crack growth using a built-in piezoelectric sensor/actuator network: I. Diagnostics", *Smart Materials and Structures*, vol. 13, pp. 609-620, 2004.

[2] J. -B. Ihn, and F. -K. Chang, "Detection and monitoring of hidden fatigue crack growth using a built-in piezoelectric sensor/actuator network: II. Validation using riveted joints and repair patches", *Smart Materials and Structures*, vol. 13, pp. 621-630, 2004.

[3] P. Dang, H. Stephanou, F. Ham, and F. L. Lewis, "Facial expression recognition using a two stage neural network", *Proceedings of the 15th Mediterranean Conference on Control & Automation*", Athens, Greece, 27-29 Jul., 2007.

[4] T. Kohonen, Self-Organizing Maps, Second Edition, Berlin: Springer-Verlag, 1997.

[5] Z. Su, L. Ye, and Y. Lu, "Guided Lamb waves for identification of damage in composite structures: A review", *Journal of Sound and Vibration*, vol. 295, pp. 753-780, 2006.

[6] V. Giurgiutiu, Structural Health Monitoring with Piezoelectric Wafer Active Sensors, Burlington, MA: Academic Press/Elsevier, 2008.

[7] K. Zgonc, and J. D. Achenbach, "A neural network for crack sizing trained by finite element calculations", *NDT & E International*, vol. 29, no. 3, pp. 147-155, 1996.

[8] V. Lopes Jr., G. Park, H. H. Cudney, and D. J. Inman, "Smart structures health monitoring using artificial neural network", *Structural Health Monitoring*, pp. 976-985, 1999.

[9] C. Zang, M. I. Friswell, and M. Imregun, "Structure damage detection using independent component analysis", *Structural Health Monitoring*, vol. 3, pp. 69-83, 2004.

[10] U. Bork, R. E. Challis, "Artifitial neural networks applied to Lamb wave testing of T-form adhered joints", *Proc. of the Conference on the Inspection of Structural Composites*, Bentham Press, 1994.

[11] L. H. Yam, Y. J. Yan, J. S. Jiang, "Vibration-based damage detection for composite structures using wavelet transform and neural network identification", *Composite Structures*, 60, pp. 403-412, 2003.

[12] Z. Su, L. Ye, "Lamb wave propagation-based damage identification for quasi-isotropic CF/EP composite laminates using artificial neural algorithm, part I: methodology and database development", *Journal of Intelligent Material Systems and Structures*, 16, pp. 97-111, 2003.

[13] L. Ye, Z. Su, C. Yang, Z. He, and X. Wang, "Hierarchical development of training database for artificial neural network-based damage identification", *Composite Structures*, 76, pp. 224-233, 2006.

A Smart Grid Testbed using Wireless Sensor Networks in a Building

Kwang-Soo Kim Hyunhak Kim Tae-Wook Heo Yoonmee Doh Jong-Arm Jun
RFID/USN Research Department
Electronics & Telecommunications Research Institute
Daejeon, Republic of Korea
e-mail: {enoch, hh.kim, htw398, ydoh, jajun}@etri.re.kr

***Abstract*—This paper describes the implementation and results of a field demonstration that monitors the usage and the generation of electricity in a two-story building. The field demonstration took place in Jeju to increase energy efficiency and to evaluate the stability of the developed system including smart meters, a wind power generator, a photovoltaic power generator, a rechargeable battery, electric vehicle chargers, light controllers, and a smart outlet. The light controllers exchange data through a power line communication; the other devices exchange those through a wireless sensor network based on ZigBee. A centralized monitoring server has operated to collect periodically all data such as the amount of the electricity consumption and generation and to control the amount of electricity consumption and charge. This testbed provides valuable insights about design decisions of a smart grid using wireless sensor networks.**

Keywords-wireless sensor networks; smart grid; testbed; home area; energy efficiency

I. Introduction

The rapid industrialization and indiscriminate development in a lot of countries have increased greenhouse gas emissions, caused environmental degradation, and depleted natural resources. To overcome these problems, many counties have sought to develop technologies to reduce the use of natural resources as well as the greenhouse gas emissions. A smart grid is considered one of innovative technologies that reduce the greenhouse gas emissions by increasing energy efficiency [1][2][3].

A smart grid refers to a next generation electric power network that combines information technologies and power technologies. The smart grid includes all functions of a utility such as power generation, electricity distribution, electricity transmission, and energy trading. Some of these functionalities related to utilities are already implemented through some automation systems such as SCADA (Supervisory Control and Data Acquisition) and DAS (Distribution Automation Systems) [4]. Recently, the smart grid is extending to include customers. Therefore, it becomes a system that enables two-way communications between consumers and suppliers [5]. Through the communications, the energy consumption of a consumer is transmitted to an electric power company and the company sends control messages to reduce the energy consumption to the consumer. The energy consumption is measured by a smart meter on the consumer side and the energy control is executed by smart appliances. In addition to the devices, the smart grid includes distributed energy such as renewable energies.

There are several competing technologies for capturing and transmitting the electricity usages of consumers in the smart grid, such as wired technology, power line communication (PLC) technology, and wireless sensor network (WSN) technology [6]. Although each technology has its own advantages, the WSN technology is very promising candidate among these technologies for several reasons. The WSN technology represents an emerging set of technologies that will have profound effects across a range of industrial, scientific, and energy management applications [7]-[15]. The WSN can reduce wiring cost and time for the smart grid deployment. Also, the WSN technology can reduce labor costs by simplifying installation. Moreover, it is one of key solution for facilities that frequently reconfigure spaces and places where a wire communication is difficult to apply. Meanwhile, in the residential area, the WSN is regarded as a part of the home network system. Accordingly, various service concepts which integrate the smart grid with home networks can be derived [6][7][8]. By introducing WSN technologies which assure network flexibility and mobility, it is easier to provide value added services like electricity equipment control.

The ZigBee technology is one of the most popular wireless standards to implement the monitoring and controlling of the energy consumption for the smart grid. The ZigBee Alliance published the smart energy profile for interoperable products that monitor, control and automate the delivery and use of energy. The profile includes several specifications related to the advanced metering, the demand response and load control, pricing, and text message [16].

We implemented several devices based on the ZigBee standards and some of them were certified by the ZigBee Alliance. They were installed at a testbed in Jeju Island, South Korea. Our system focuses on measuring, monitoring, and controlling the energy consumption on the customer side.

The remainder of this paper is organized as follows. The motivation is discussed in Section II. The detailed design of the testbed is described in Section III. Finally, Section IV provides the conclusion and the future work.

II. Motivation

In this section, we discuss the motivation of this study. In South Korea the demand of electricity is growing faster every year. If the peak electricity demand grows higher than the power generation, then the power network is broken down. Figure 1 shows the increasing trend of the power production and the consumption in South Korea from 2005 to 2009. We can clearly see in Figure 1 that the electricity consumption is growing every year. To supply the energy for consumers without outage, the utility company, KEPCO (Korea Electric Power Corporation), can do two activities: constructing a new power plant and reducing energy consumption. The construction of a new plant may not be reasonable in a certain aspect. It requires a lot of costs and time to construct a new power plant. Also, the new plant might operate and generate the electricity only when the energy consumption approaches peak load. This situation is ineffective for utilities. Therefore, many utilities have sought to decrease the energy consumption, in other words, they want to balance the power generation and the power consumption by controlling the energy consumption.

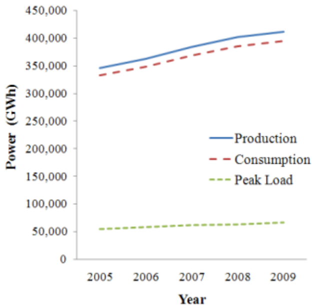


Figure 1. Trend of energy production and consumption in Korea (Source: KEPCO's Statistics, May 2010)

III. Smart Grid Testbed Design

The building in which the system was installed is a two-story building and the area of each story is about 30m × 20m. The system consists of five main components: (i) application, (ii) server, (iii) gateway, (iv) electric devices, and (v) information networks connecting all devices. They are shown in Figure 2.

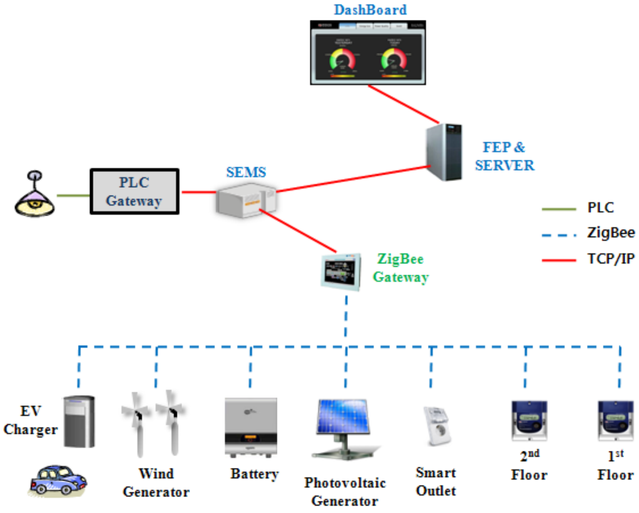
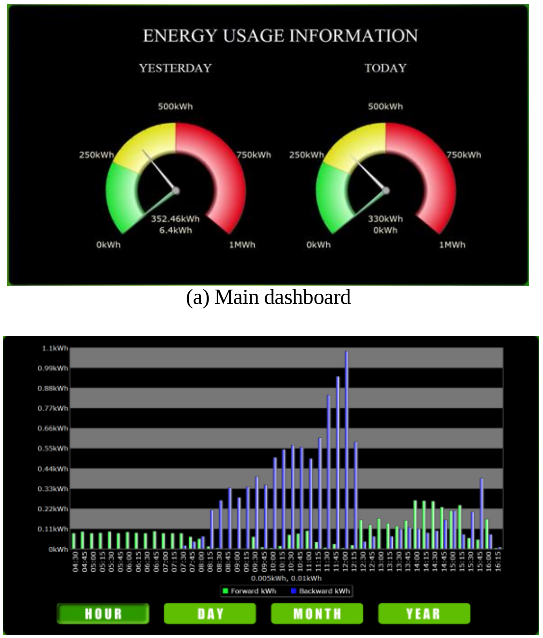


Figure 2. System configuration

A. Application

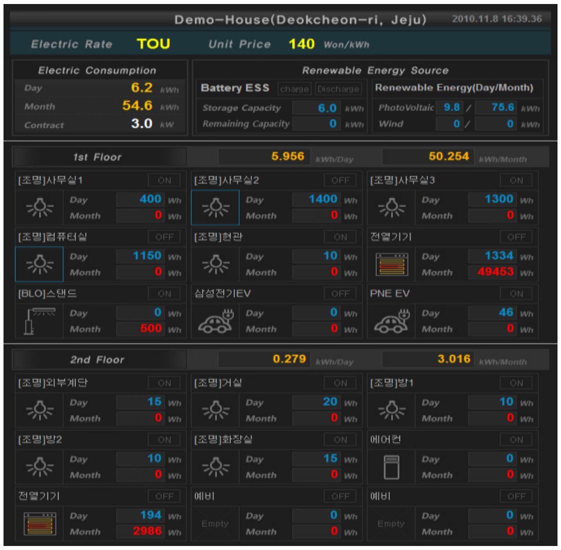
The main function of the application is to display all measurements according to the requirements of users. Two examples are shown in Figure 3. Figure 3-(a) indicates the main dashboard that displays and compares the electricity usages of today and yesterday. If the electricity gauge is displayed in red, then the user had better control the electricity usages because the electricity usage is over a threshold. Figure 3-(b) displays the electricity usages according to the specific period. The electricity usage has been measured and transmitted on every 15 minutes by the smart meters and the smart outlet. The graph based on hour will be changed according to the search condition such as day, month, or year.



(a) Main dashboard

(b) Power usage

Figure 3. Power usage information



Demo−House(Deokcheon−ri, Jeju) 2010.11.8 16:39.36

Electric Rate TOU Unit Price 140 Won/kWh

Electric Consumption
Day 6.2 kWh
Month 54.6 kWh
Contract 3.0 kW

Renewable Energy Source
Battery ESS charge Discharge
Storage Capacity 6.0 kWh
Remaining Capacity 0 kWh
Renewable Energy(Day/Month)
PhotoVoltaic 9.8 / 75.6 kWh
Wind 0 / 0 kWh

1st Floor 5.956 kWh/Day 50.254 kWh/Month

[조명]사무실1 ON Day 400 Wh Month 0 Wh	[조명]사무실2 OFF Day 1400 Wh Month 0 Wh	[조명]사무실3 ON Day 1300 Wh Month 0 Wh
[조명]컴퓨터실 OFF Day 1150 Wh Month 0 Wh	[조명]현관 ON Day 10 Wh Month 0 Wh	전열기기 OFF Day 1334 Wh Month 49453 Wh
[BLO]스탠드 ON Day 0 Wh Month 500 Wh	삼성전기EV OFF Day 0 Wh Month 0 Wh	PNE EV ON Day 46 Wh Month 0 Wh

2nd Floor 0.279 kWh/Day 3.016 kWh/Month

[조명]외부계단 ON Day 15 Wh Month 0 Wh	[조명]거실 ON Day 20 Wh Month 0 Wh	[조명]방1 ON Day 10 Wh Month 0 Wh
[조명]방2 ON Day 10 Wh Month 0 Wh	[조명]화장실 ON Day 15 Wh Month 0 Wh	에어컨 ON Day 0 Wh Month 0 Wh
전열기기 OFF Day 194 Wh Month 2986 Wh	예비 OFF Empty Day 0 Wh Month 0 Wh	예비 OFF Empty Day 0 Wh Month 0 Wh

Figure 4. SEMS's user interface

B. *Server*

The server is divided into SEMS (Smart Energy Management Server) and FEP (Front End Processor). The functionalities of SEMS and FEP are similar; however, their coverage is different. SEMS is installed a relatively narrow area such as a home, a building, a university campus, a factory, an apartment, etc; on the other hand, FEP covers a wide area such as a city by connecting to a lot of SEMSs. Therefore, SEMS gathers and manages the electricity usage at the fine-grained level, in other words, it captures directly the electricity usage of each electricity device. FEP summarizes the values gathered by SEMSs. Figure 4 displays the user interface of SEMS. The electricity usages of lighting, EV (Electric Vehicle) chargers, a battery, and renewable power generators are displayed separately. SEMS sends a query at 15-minute intervals to gather the electricity usage to all electric devices and sends the gathered data to FEP. In addition, SEMS sends a control message (e.g., turn a device on/off, charge/discharge the battery) to the smart outlet, the battery, and the light controller. The control message is generated by a demand response program in SEMS according to predefined rules. For example, if the electricity usage exceeds a threshold, the demand response program sends a turn-off message to the outside lighting devices. If the usage exceeds the threshold again, then it sends another message to those near to windows.

C. *Gateway*

The gateway becomes a bridge between the wireless sensor network and SEMS through an IP-based network. The gateway provides the gateway device interface and the smart energy profile 1.0 announced by ZigBee Alliance [16][17] as well as provides the functions that connect and manage all electric devices within HAN (Home Area Network). The gateway includes PANC (Personal Area Network Coordinator) starting the network formation and routing messages between the gateway and each electric device. The gateway is implemented on Linux 2.6 in the C programming language, and uses PHP (Personal Hypertext Preprocessor) for the web application to exchange data with SEMS. The gateway and SEMS exchange XML messages following REST (Representational State Transfer) specification [17]. Figure 5 shows the gateway hardware and the main window.

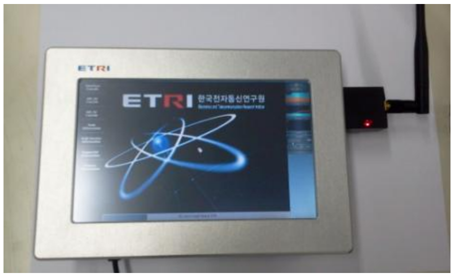


Figure 5. Gateway

D. *Electric Devices*

The electric devices consist of five smart meters, two wind power generators, a photovoltaic power generator, a rechargeable battery, two electric vehicle chargers, two light controllers, and a smart outlet. The light controllers exchange their data and control messages through PLC and the other devices exchange those through a wireless sensor network. The smart meters measure the electricity usage of the first and second floor, the two electric vehicle chargers, and the smart outlet as well as the electricity generation of two wind generators and one photovoltaic generator. The wind and the photovoltaic generator produce 0.8KWh and 6 KWh of energy, respectively. The measurements of the meters are transmitted to the gateway. The meters send LP_Data (Load Profile Data) including forward active power, backward active power, reactive power, etc. The forward power indicates that it is supplied for a customer by a utility; the backward power indicates that it is supplied for the utility by the customer. The outlet measures the electricity usage of an appliance (e.g., TV, air conditioner, refrigerator, etc.) which connects to it and transmits the measurement to the gateway. The rechargeable battery is a storage device which repeats charging and discharging according to the predefined conditions. It can store the power generated by the wind and the photovoltaic generator and dispatch power to supply electricity for streetlights over a night. Its current state and the charge level are transmitted to the gateway. Depending on a battery type and capacity, the high speed and the low speed EV charger can take about 3 hours and 12 hours to fully recharge a battery. The light controller measures the electricity usage of the first and the second floor, and transmits them to the PLC gateway.

TABLE I. COMPARISON RESULTS OF COMMUNICATION SCHEMES

Competitiveness Items	ZigBee	Narrowband PLC	Z-Wave
Communication Range	1~75m+	Max 1Km(bet. PLC modems)	30~500m
Data Transfer Speed	40K~250Kbps	1Kbps~20Kbps	10-40Kbps
Communication Stability	Partly Limited	Very Limited	Limited
Interoperability with HAN Devices	Very High	Low	Medium
Standardization Support	Global Standard	De facto Standard	Local Standard
Security	Very Good	Good	Good
Scalability(Number of Device Adders)	255+	20-255	232
Convenience in Deployment	Non-line-of-sight feature is suitable for various environment	needs to deploy repeaters and modems on the wire	Non-line-of-sight feature is suitable for various environments
Battery Support(One AAA battery with network activation on a minute basis)	Weeks	No capability of battery support	Weeks ~ Months
Mobility	High	Low	Medium

E. *Information Networks*

In a smart grid, two-way communication allows the data exchange between the customer side and the supply side.

There are several technologies that can be used in the smart grid [6]. We compare the technologies in terms of communication range, data transfer speed, interoperability with HAN devices, standardization support, etc. The result is shown in Table I.

We select PLC and the ZigBee technology. As PLC uses the existing power line infrastructure, it is used to capture and to control the electricity usage of lighting devices in our testbed. On the other hand, newly installed devices are developed based on ZigBee because it is one of the most popular wireless standards to implement the monitoring and controlling the energy consumption for the smart grid through the smart energy profile. Therefore, our wireless sensor network system is developed based on ZigBee specification and IEEE 802.15.4. The sensor network installed on each floor consists of one gateway and several nodes corresponding to the electric devices respectively. In this building environment, the devices can exchange messages via one-hop communication with star topology because every device is within the radio range of the gateway. After PANC included in the gateway starts the network formation, a node willing to associate with the network starts the association procedure by requesting for a beacon with channel scanning. A joined node permits the association by beaconing with setting permit-joining flag on. Once a node has associated the network, it maintains three data tables: routing table, neighbor list table and link cost table. The maintaining of those tables allows the further expanding of the network up to the mesh topology, and the size of each table is resizable according to the network size.

IV. Conclusion and Future Work

This paper describes a smart grid testbed using a wireless sensor network within a small building. To monitor and control the usage and the generation of the electricity in the building, we installed two monitoring servers and several electric devices including five smart meters, two wind power generators, a photovoltaic power generator, a battery, two electric vehicle chargers, two light controllers, and a smart outlet. The light controllers exchange their data and control messages through PLC and the other devices exchange those through a wireless sensor network. By visualizing the electricity usage and running a demand response program based on the electricity usage, the energy consumption could be saved. Also, the building owner could reduce the amount due on the electricity bill because most of the consumed energy has been supplied by the renewable power generators.

In the future, we will execute a demand response program combining with a real-time pricing policy. In addition, we will perform the economic analysis on the smart grid testbed to apply the devices to other households and buildings.

Acknowledgment

This work is supported by Development of Self-Powered Smart Sensor Node Platform for Smart and Green Building [10035570] funded by the Ministry of Knowledge and Economy, Republic of Korea. The authors would like to thank KEPCO-KDN members for working together.

References

[1] The National Energy Technology Laboratory, “A Vision for the Smart Grid,” 2009.

[2] SBI, “Smart Grid Technologies, Markets, Components and Trends Worldwide,” 2009.

[3] ROA Group Korea Consultants, “Introduction to Smart Grid: Latest Developments in the U.S., Europe and South Korea,” Jul. 2009.

[4] M. J. Jang, B. N. Ha, and S. W. Lee, “The Study on the Design of the Smart Grid Test-Bed in KEPCO,” Selected Topics in Power Systems and Remote Sensing, Proc. 10th WSEAS/IASME Intervational Conference on Electric Power Systems, High Voltages, Electric Machines, 2010, pp.348-350.

[5] Pacific Northwest National Laboratory, “Pacific Northwest GridWise™ Testbed Demonstration Projects,” Oct. 2007.

[6] S. J. Kim, J. H. Seo, J. A. Jun, and C. S. Pyo, “Advanced Metering Infrastructrue (AMI) Service for Efficient Energy Management,” Proc. 19th European Regional International Telecommunications Society Conference, Oct. 2008.

[7] S. J. Kim, “Smart energy management for buildings with wireless sensor technology,” Proc. 20th European Regional International Telecommunications Society Conference, Oct. 2009.

[8] K. Kim and J. A. Jun, “Smart Energy Server using Wireless Sensor Networks,” Proc. International Symposium on Remote Sensing, Oct. 2009.

[9] I. F. Akyildiz, D. Pompili, and T. Melodia, “Underwater acoustic sensor networks: research challenges,” Ad Hoc Networks, Vol. 3, 2005, pp. 259-279.

[10] W. Chen, L. Chen, Z. Chen, and S. Tu, “WITS: A Wireless Sensor Network for Intelligent Transportation System,” Proc. the First International Multi-Symposiums on Computer and Computational Sciences, Vol. 2, Apr. 2006, pp. 635-641.

[11] J. A. Gutierrez, D. B. Durocher, and B. Lu, “Applying Wireless Sensor Networks in Industrial Plant Energy Evaluation and Planning System,” Proc. IEEE IAS Pulp and Paper Conference, Jun. 2006, pp.1-7.

[12] J. K. Hart and K. Martinez, “Environmental Sensor Networks: A revolution in the earth system science?,” Earth-Science Reviews, vol. 78, 2006, pp.177-191.

[13] P. Jiang, H. Ren, L. Zhang, Z. Wang, and Z. Xue, “Reliable Application of Wireless Sensor Networks in Industrial Process Control,” Proc. the 6th World Congress on Intelligent Control and Automation (WCICA), Vol. 1, Jun. 2006, pp. 99–103.

[14] K. Kim, J. Jun, S. Kim, and B.Y. Sung, ”Medical Asset Tracking Application with Wireless Sensor Networks,” Proc. International Conference on Sensor Technologies and Applications (SENSORCOMM), Aug. 2008, pp. 531-536.

[15] J. A. Stankovic, “When Sensor and Actuator Networks Cover the World,” ETRI Journal, Oct. 2008, pp. 627-633.

[16] ZigBee Alliance, “Smart Energy Profile Specification,” (075356r15ZB), Dec. 2008.

[17] ZigBee Alliance, “ZigBee Gateway Device Specification,” (075468r30ZB), Jul. 2010.

Designing and Implementing a Middleware for Data Dissemination in Wireless Sensor Networks

Ronald Beaubrun
Department of Computer Science and Software Engineering
Université Laval
Quebec, Qc, Canada
e-mail: ronald.beaubrun@ift.ulaval.ca

Jhon-Fredy Llano-Ruiz, Alejandro Quintero
Department of Computer and Software Engineering
École Polytechnique de Montréal
Montreal, Qc, Canada
e-mail: {jhon-fredy.llano-ruiz, alejandro.quintero}@polymtl.ca

***Abstract*— In this paper, we propose an approach for designing and implementing a middleware for data dissemination in Wireless Sensor Networks (WSNs). The designing aspect considers three perspectives: device, network and application. Each application layer is implemented as an independent Component Object Model (COM) Project which offers portability, security, reusability and domain expertise encapsulation. For result analysis, the percentage of success is used as performance parameter. Such analysis reveals that the middleware enables to greatly increase the percentage of success of the messages disseminated in a WSN.**

Keywords- Data dissemination, design, implementation, middleware, wireless sensor network.

I. Introduction

The main goal of a Wireless Sensor Network (WSN) is to gather environmental information in a specific region and make it available to users. For this purpose, it uses a set of sensor nodes, *i.e.*, a set of devices that sense and measure environmental variables, such as light, temperature, humidity and barometric pressure [1]. Another important component of a WSN is the Base Station (BS). Since sensors are normally battery-constrained and equipped with low system capabilities, they need to transfer their collected data to a long-life device. Laptops, Personal Computers (PCs), handhelds and access points to a fixed infrastructure are examples of physical devices used as BSs. To make the communication possible between SNs and BSs, a gateway (GW) is set in between, acting as a bridge. Figure 1 shows an example of a WSN.

In a WSN, the exchange of information between the SNs, the GW and the BS is done through a data-dissemination technique where the information is transported towards different destinations [2-5]. For this purpose, a middleware is required between the network and the applications to offer tracking capabilities of the disseminated information. Using a data dissemination protocol, this middleware can take on-time decisions when a maximum end-to-end delay constraint is exceeded. This paper proposes an approach for designing and implementing a middleware for data dissemination in WSNs.

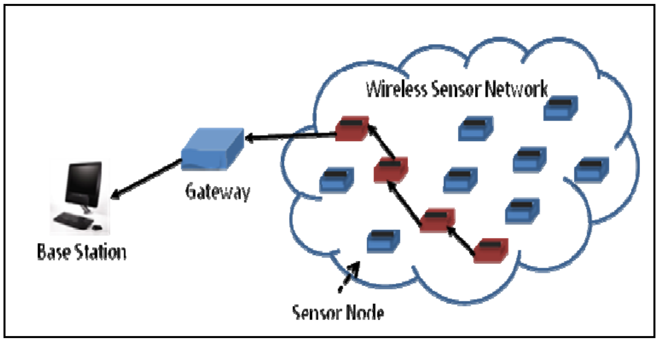


Figure 1. Example of a Wireless Sensor Network.

The rest of the paper is organized as follows. Section II presents the designing aspects of the middleware. Section III focuses on the software components that lead to the middleware deliverables. Section IV describes the implementation of each software component. Section V presents some results and analysis, whereas Section VI gives some concluding remarks.

II. Designing Aspects

A. Reference architecture

Figure 2 illustrates the general architecture considered for this research from the infrastructure point of view. It integrates a WSN with two other networks: the source of information is the sensor network, whereas the destination can be Internet or a Cellular Network. This architecture considers two roles: the *Message Originator* (MO), which is responsible for initializing the notification process, and the *Message Terminator* (MT), which receives the information

ISBN: 978-1-61208-144-1

and sends back a response. MT is a role played by any person or device in the system. In case of a person, it can be either a *Security Group* (SG) member, or a *User Group* (UG). The MO represents each single sensor node that is deployed. It collects information that could be disseminated. If an event is detected, the sensor node starts the dissemination process towards the gateway (GW), using the forwarders in between. The GW is responsible for receiving the information sent by any node in the WSN, and conveys it to the base station (BS). Once the BS receives the information, it will make a decision depending on its own configuration, *e.g.*, Send information to UG and SG through different protocols, such as *Short Message Services* (SMS), *email* or *twitter*.

B. Model roadmap

Figure 3 presents a general overview of the middleware. Therein, the middleware is initially related to *delay-constrained applications*, as it aims to produce support for such type of applications. In order to guarantee this support, it requires a direct communication with *data dissemination protocols* at any moment. Therefore, the top layer represents *delay-constrained applications* that use the middleware which, in turn, is the intermediate layer. In the meantime, it imposes some requirements (*e.g.*, end-to-end delay) to the underlying *data dissemination protocols* located in the bottom layer.

The middleware deals with different data dissemination protocols, and it requires to be executed on different types of devices (*i.e.*, SNs, GWs, BSs), which forces each environment to control different configurations and specificities. For such a purpose, the approach from [6] is adopted. It considers three points of view: *device*, *network* and *application*. Firstly, *device perspective* focuses on each device and its components, considering five features: *type of devices*, *operating systems*, *radio technology*, *development technologies* and *storage*. *Type of devices* represents different machines where the middleware is intended to be executed (*i.e.*, SNs, GWs, BSs). *Operating systems* represent different operational platforms running on the types of devices (*i.e.*, *TinyOS*, *Linux* and *Windows*). *Radio technology* is used to establish communication with other nodes of the architecture (*i.e.*, 802.11). Additionally, *development technologies* features need to be taken into consideration. For the suitability of these technologies and their widely acceptance in the academia, *nesC*, *Java* and *C++* have been chosen. Finally, *storage* takes care of the persistence of the information when needed (*i.e.*, databases, XML files).

Secondly, *network perspective* represents the dissemination of information among the network. It takes into account several network characteristics, *e.g.*, end-to-end delay, confirmation mechanisms and energy optimization. In other words, the *network perspective* takes into account three network services in order to achieve the requirements: *delivery manager*, *message sender manager* and *service manager*. *Delivery Manager* (DM) is responsible for managing the delivery process. It tracks messages sent along the process while considering delay constraints. It includes: reporting, receiving and analyzing capabilities. *Message Sender Manager* (MSM) is in charge of the message sending process. It is made up of three main processes: listening, analyzing and sending. *Service Manager* (SM) is a service that allows managing the protocols the system works with and the resources associated to each of them.

Finally, the *application perspective* represents the applications using the middleware services. It is divided into two main categories: *delay-constrained applications* and *user applications*. On one hand, delay-constrained applications have strict Quality of Service (QoS) constraints, and are used to warn people in emergency events. They require a continuous feedback from the middleware. On the other hand, user applications tolerate lower QoS constraints due to their specific goals. Failures or delays are not as critical as they are for the former category. As a result, confirmation mechanisms could be avoided or delayed. Despite of that, user applications can use the middleware as well.

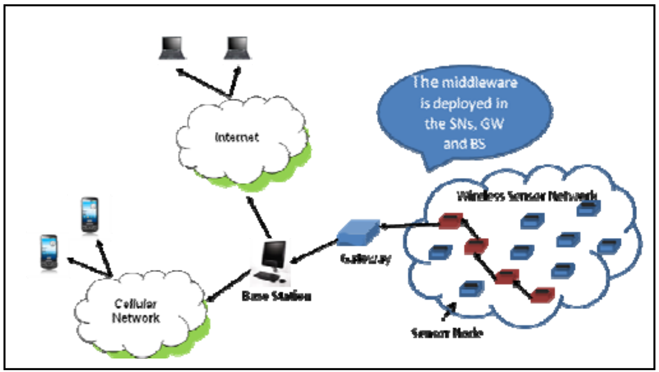


Figure 2. Global Architecture.

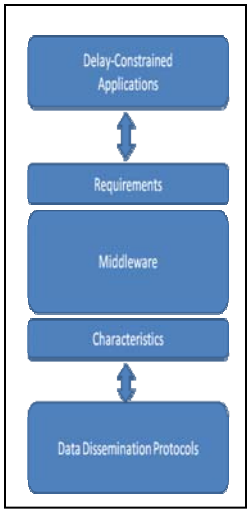


Figure 3. General overview of the middleware.

The integration of such perspectives constitutes the roadmap of this proposition, guarantying a holistic view of the system. This amalgamation is intended to show that all perspectives are present at any time in the system and the intersection of all of them produces the middleware. Figure 4 depicts such integration. The 3D-view offers the possibility to analyze the system from different perspectives while preserving the unity and respecting the requirements and constraints.

III. SOFTWARE COMPONENTS

This section focuses on presenting the software components that lead to the middleware deliverables. Firstly, the class diagram that shows the class interactions within the whole system is presented. Later on, the sequence diagrams that show the interaction of the architecture components are explained.

A. *Class diagram*

Figure 5 presents the class diagram of the middleware, giving a static view of the system. It is divided into four logical layers which depict the main components presented in the reference architecture. The first three layers refer to five main components: *Interfaces*, *Message Sender Manager*, *Delivery Reporter Manager*, *Data Access Manager* and *Service Manager*, whereas the bottom layer represents the data dissemination protocols to be used. On top of the diagram, a set of *Interfaces* classes offers a unique way for consumers to use the middleware services. It is made up of three classes that interact with the second layer components. In the second layer, the *Message Sender Manager* is responsible for managing the sending process while considering three main classes: *Listener*, *Analyzer* and *Sender*. *Listener* senses new messages that arrive to the middleware. *Analyzer* consists of four classes, which means that all classes need to participate in the process when the analyzer is executing. Finally, *Sender* is in charge of sending the analyzed message. Then, the *Delivery Report Manager* classes track the message status. Similarly to the previous component, it also considers three classes: *Reporter*, *Analyzer* and *Receiver*.

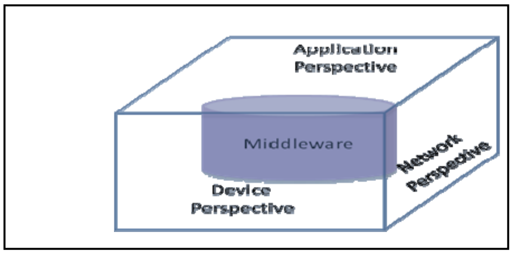


Figure 4. Middleware roadmap.

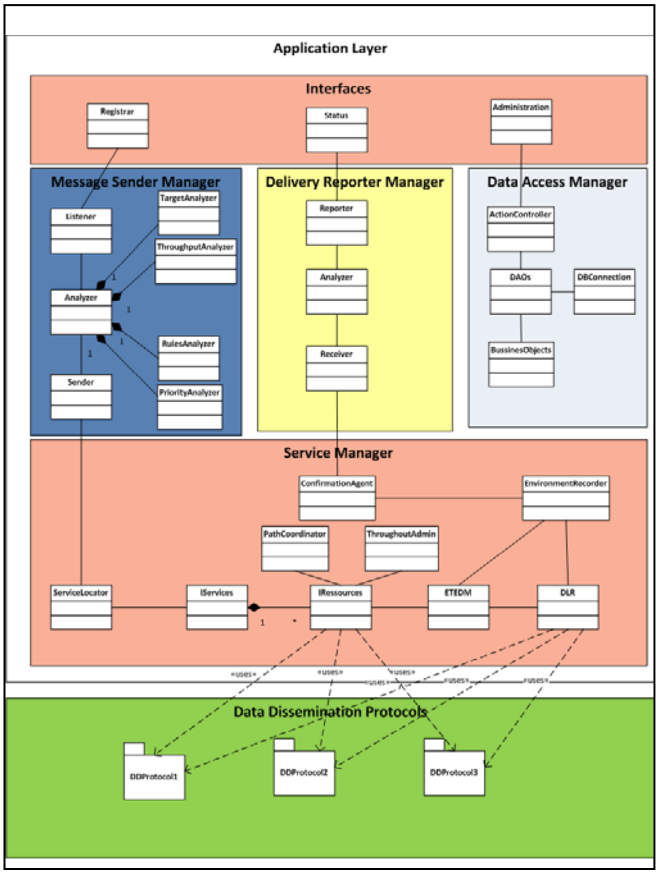


Figure 5. Class diagram of the reference architecture.

Furthermore, *Data Access Manager* is responsible for providing and modifying the data models (*i.e.*, databases and configuration files). It uses an *ActionController* which is responsible for receiving an action to be executed and identifying which component in the system will realize it. Normally, this action is assigned to a Data Access Object (DAO), which in turn, affects the information relying on any Business Objects (BOs). Finally, the *Service Manager* is responsible for interacting with the protocols and the network to complete either the message sending process, or the delivery report process. It consists of a set of classes that offer system characteristics, such as end-to-end delay (ETEDM), delivery report (DLR), environment events (*EnvironmentRecorder*), Confirmation features (*ConfirmationAgent*) and sending of messages (*IServices* and *IRessources*). *Service Manager* relies on a *ServiceLocator* which identifies the most appropriate services and protocols according to the application requirements.

As previously discussed, the middleware is located in the application layer. In order to perform its tasks, it should have access to specialized protocols that are normally located in lower layers in the communication stack. For such reason, the bottom layer shows the available protocols to be used and their interactions with the middleware. It is important to notice that this proposal is protocol independent, which means that any protocol could be used as long as it supports the application requirements. Therefore, it is up to the implementers to choose the right dissemination protocol.

B. *Sequence diagrams*

The sequence diagrams present a dynamic view of the system. Two main processes are described: the message sending process which shows how the components participate in order to offer end-to-end delay and confirmation support to the messages sent, and the delivery report process which enables to have knowledge about messages states at any moment.

1) Message sending process: As depicted in Figure 6, this process is initiated by a sensor, a gateway or a base station when a new message arrives. Any of them may register a new message using the *Registrar* interface. This interface puts the message into a Queue waiting for *Listener* to be in charge of it. *Listener* is a daemon process responsible for the surveillance of new messages that arrive. For this purpose, it executes asynchronous calls to *MessageQueue*. Once it discovers a message standing there, it takes the message and passes it to a new phase to be analyzed. This process is broken up into 4 stages: analysis of destinations, priorities, rules and throughput. These stages heavily depend on the environment where the middleware is deployed. Once the whole analysis is completed, a *Sender* class is called to send the message. Later, the *ServiceLocator* class receives the *Sender* request in order to locate the service and the resource that will be responsible for disseminating the information towards the destinations. For such a purpose, this class takes into consideration basic information, such as priorities and rules. Once the resource is identified (*i.e.*, dissemination protocol with its parameters), *IRessource* begins to interact with the protocol, which finally is responsible to convey the information to the destinations, considering the application requirements.

At the same time, ETEDM, which is the process to offer timeliness support, is activated. It controls delay-constraints for each message sent while verifying the acknowledgements (ACKs) or negative acknowledgements (NACKs) sent by the protocol. If no response is received by the end of this time period, it asks the *ServiceLocator* to look for another service and resource to disseminate the information, *i.e.*, the lookup process. This cycle is repeated based on the middleware configuration.

2) Delivery report process: Any device (*i.e.*, a sensor node, a gateway, a base station) or internal component in the architecture (*e.g.*, a sender) may want to know the status of a message sent at any time. The sequence shown in Figure 7 details how this process is executed. Once a device or a component interrogates the *Status* interface, this request is transferred to the system, then analyzed further to identity the message that is going to be tracked. Once this identification is performed, *ConfirmationAgent* is interrogated. It reads and analyzes the information presented by *EnvironmentRecorder*, which tracks all the events that happen with the message, such as end-to-end delay information, DLR and network failures. Based on this analysis, *ConfirmationAgent* presents a response to the system, which is sent back to the *Status* interface, then to the user or component interested in this information.

IV. IMPLEMENTATION

The application is divided into three main logic components: *Interfaces*, *Business Rules* and *Data Services*. Each layer is implemented as an independent Component Object Model (COM) Project which offers portability, security, reusability and domain expertise encapsulation.

A. *Interfaces Layer*

The *Interfaces* layer exposes functionalities as services and variables. It provides a method called *registrar* for the applications to register the events. The definition of this method is presented as follows:

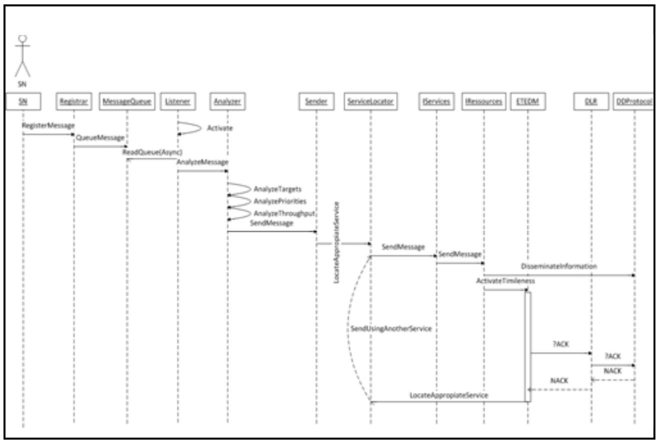


Figure 6. Message sending process.

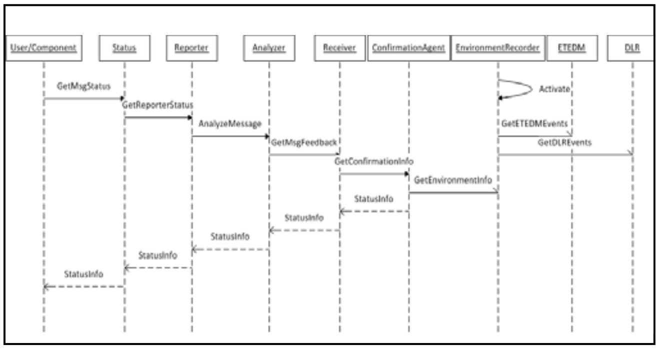


Figure 7. Delivery report process.

```
public static void registrar(int priority,
                             String shortDescription,
                             String description,
                             String source,
                             int type,
                             String comments);
```

It receives six mandatory parameters. First, *priority* is used to establish the priority of the message (*e.g.*, 100=emergency). Then, *shortDescription* contains a brief description about the event. The third parameter, *description*, contains a more detailed description of the incident (*e.g.*, sensor *x* registered a value of light intensity *y* in the *z*-building). Next, *source* indicates the origin of this information. Then, *type* refers to the type of the originator (*e.g.*, sensor node, gateway, base station). Finally, *comments* permits to include any additional information required to complement the message. This information can be presented in an XML format for a better portability. Using this service, the events that come from the WSN are initiated in the middleware.

B. *Business Rules Layer*

The *Business Rules* layer is the core of the system, since it implements the basic components: *Message Sender Manager*, *Service Manager* and *Delivery Report Manager*, enabling messages to be sent through different protocols. To set up these protocols, an XML file is generated. It might be noticed that each protocol is composed by one or multiple resources, supporting the definition made in Figure 5. Table I describes the tags composing the file. As described in this table, each resource might require several parameter values to be described and configured. Figure 8 presents a fragment of *resources.xml* file for the implemented prototype.

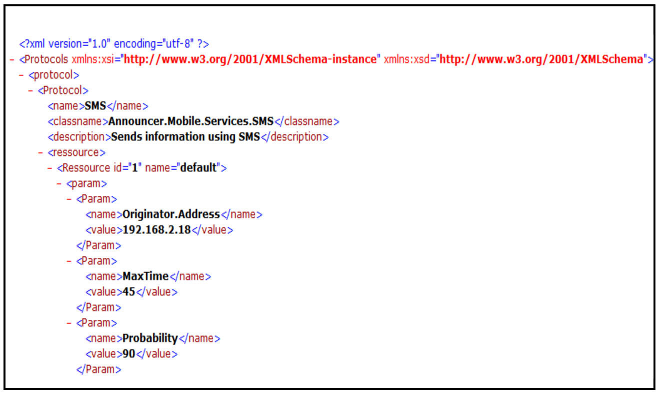
It can be noticed that the instance shows an SMS resource configuration. The *tag name* is used to identify the protocol used. The tag class describes the name of the class that implements the service. It is dynamically executed using on-the-fly *.Net* capabilities (also known as assemblies). This feature makes the environment execution more versatile, since it only requires setting up the XML. The information is sent in strict order according to its appearance in this file. The maximum set up time for each resource to complete its task is obtained from the XML file. This information is defined using a probabilistic approach based on studies done on the efficiency of these resources, as stated in [7]. The DLR interface is simulated using these probabilistic values to know whether the message was successfully received or not.

C. *Data Services Layer*

This layer is responsible for providing the interfaces the access to the information. This information is mainly stored in two locations: the database and the XML resources file. Figure 9 presents the Entity/Relation (E/R) diagram for the middleware. It can be seen that there is a table called *queued_message*, where the message is initially queued using the *registrar* service. Then, the middleware using the *listener* processes moves the record to the *message* table. Later on, after the analysis is done, the single message is multiplexed into multiple records. Each message is addressed to a single user, using a different protocol and device (*e.g.*, SMS-blackberry, Email-iPhone), as defined in the XML file and in the database configuration. This information is stored in the *sent_message* table. The DLR obtained from each service is recorded in the *status* attribute. By using this information, the middleware knows the state of each single message sent to any user in the system.

TABLE I. XML RESOURCES FILE DESCRIPTION

Tag Name	**Tag Description**
Protocols	Indicates the beginning and the end of the resources file
Protocol	Indicates the beginning or the end of a protocol
name (protocols)	Contains the name of a protocol
Classname	Describes the name of the class fully specified. *Package.ClassName.* This value is used by the middleware to dynamically execute the class using on-the-fly capabilities (*i.e.*, assemblies loaded and executed when needed). It allows the middleware to execute assemblies that might or might not be part of it.
description	Brief description of the protocol
Resource	Indicates the beginning or the end of a resource. For instance, a SMS could be sent using different SMS Gateways.
name (resource)	Contains the name of a resource.
param-name	Details all the parameters required to describe a resource. For instance, a SMS Gateway requires an IP address, a port, a user, a password and URL among others. It additionally describes maximum time to wait for a response and the probability of receiving an ACK.



```
  <?xml version="1.0" encoding="utf-8" ?>
- <Protocols xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  - <protocol>
    - <Protocol>
        <name>SMS</name>
        <classname>Announcer.Mobile.Services.SMS</classname>
        <description>Sends information using SMS</description>
      - <ressource>
        - <Ressource id="1" name="default">
          - <param>
            - <Param>
                <name>Originator.Address</name>
                <value>192.168.2.18</value>
              </Param>
            - <Param>
                <name>MaxTime</name>
                <value>45</value>
              </Param>
            - <Param>
                <name>Probability</name>
                <value>90</value>
              </Param>
```

Figure 8. Resource file.

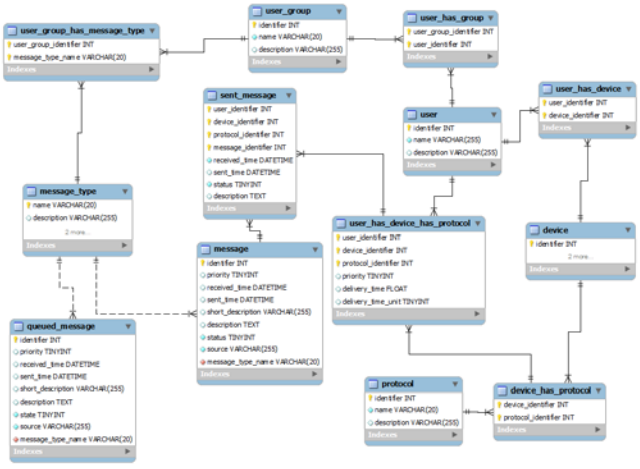


Figure 9. Entity/Relation Diagram.

V. RESULTS AND ANALYSIS

For the experiments, we use the architecture shown in Figure 2, with 3 sensor nodes in the WSN. For each message sent through a resource (*i.e.*, SMS, *email* or *twitter*), the percentage of success, *i.e.*, the ratio between the number of messages sent and those successfully received by the destination, is registered. These values are then processed and analysed in MATLAB. Table II presents a fragment of the results obtained from the first experiment. The first column shows the corresponding statistical attributes analyzed, *i.e.*, the percentage of success, the number of received ACKs, the number of received NACKs or no responses (NRs). For the middleware, the percentage of success is 98.25%. Accordingly, 7 destinations are not successfully notified among 400 messages sent.

Now, we can analyze the percentage of success for each resource in 20 experiments. Figure 10 shows that the middleware outperforms the other resources taken individually. More specifically, the overall success of the middleware is close to 98%, which represents a great improvement when compared with the performance of the resources individually. For instance, SMS shows an average success of 78%. A slightly increment is seen in *email* with 79%. Finally, *twitter* offers the lowest success of the three individual resources (61%).

TABLE II. RESULTS FROM THE FIRST EXPERIMENT

	SMS	Email	Twitter	Middleware
Percentage of Success	79.50%	70.73%	70.83%	98.25%
ACK	318	58	17	393
NACK, NR	82	24	7	7

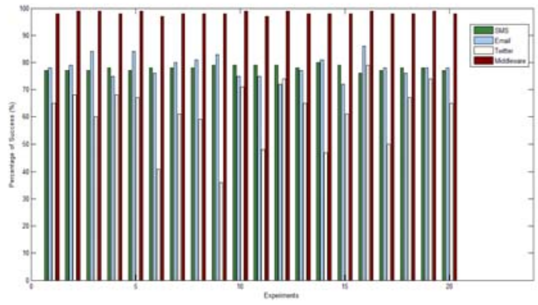


Figure 10. Comparison of percentage of success.

VI. CONCLUSION

In this paper, we proposed an approach for designing and implementing a middleware for data dissemination in WSNs. For the analysis, the percentage of success is used as performance parameter. Such analysis reveals a middleware success close to 98%, which is highly superior to the success of other individual resources.

ACKNOWLEDGMENT

This research has been supported by Prompt-Quebec and Imaginovation.

REFERENCES

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," Computer Networks, vol. 52, pp. 2292-2330, August 2008.

[2] H. M. Ammari and S. K. Das, "A trade-off between energy and delay in data dissemination for wireless sensor networks using transmission range slicing," Computer Communications, vol. 31, pp. 1687-1704, June 2008.

[3] D. Virmani and S. Jain, "Comparison of proposed data dissemination protocols for sensor networks using J-Sim," in 2009 IEEE International Advance Computing Conference. IACC 2009, Patiala, India, 2009, pp. 1179-1186.

[4] Y. Zhang and L. Wang, "A comparative performance analysis of data dissemination protocols in wireless sensor networks," in Proceedings of the 7th World Congress on Intelligent Control and Automation, Chongqing, China, 2008 pp. 6663-6668.

[5] S. Saha and M. Matsumoto, "A framework for disaster management system and WSN protocol for rescue operation," in TENCON 2007 - 2007 IEEE Region 10 Conference, Taipei, Taiwan, 2007, pp. 1315-1318.

[6] F. C. Delicato, L. Fuentes, N. Gamez, and P. F. Pires, "A middleware family for VANETs," in Ad-Hoc, Mobile and Wireless Networks. 8th International Conference, ADHOC-NOW 2009, Murcia, Spain, 2009, pp. 379-384.

[7] R. Pries, T. Hobfeld, and P. Tran-Gia, "On the suitability of the short message service for emergency warning systems," in VTC 2006-Spring. 2006 IEEE 63rd Vehicular Technology Conference, Melbourne, Vic., Australia, 2006, pp. 991-995.

The BSNOS Platform: A Body Sensor Networks Targeted Operating System and Toolset

Joshua Ellul
Department of Computing
Imperial College London
London, UK
Email: jellul@imperial.ac.uk

Benny Lo
Deparment of Computing
Imperial College London
London, UK
Email: benny.lo@imperial.ac.uk

Guang-Zhong Yang
Deparment of Computing
Imperial College London
London, UK
Email: g.z.yang@imperial.ac.uk

***Abstract*—Body sensor networks face different challenges than those faced in traditional wireless sensor networks. Challenges faced include fewer, more accurate sensor nodes and an increased requirement of context awareness, however, body sensor networks are relieved from high scalability. Programming sensor networks, in general, can be a daunting task due to the limited computation, memory and energy resources available. Operating systems for wireless sensor networks have been proposed which focus on the challenges they face. We believe that an operating system and toolset which focuses primarily on the challenges and properties of body sensor networks could help ease the burden of programming body sensor network applications. In this paper, we present an operating system which is focused on facilitating body sensor network application development.**

***Keywords*-body sensor networks; operating systems;**

I. Introduction

Body sensor networks have emerged as a promising platform to enable scientists to further understand how the human body operates with a means of remotely monitoring different environmental conditions accurately. Operating systems for wireless sensor networks have been the primary tool in developing body sensor node applications in the past. Developers require expertise in C and embedded programming to be able to create applications. Even seasoned programmers find the task of programming such applications daunting and time consuming. Domain scientists rely on and wait for such applications to be developed before they can extract any useful information from the environment they wish to investigate.

The challenges faced in body sensor networks vary to that of traditional wireless sensor networks. Body sensor networks usually consist of smaller networks of short range communication, whilst traditional wireless sensor network research has been heavily focused on routing and MAC protocols to facilitate scalability and longer range communication.

In this paper, we present the BSNOS platform, an operating system and platform designed to especially meet the requirements of body sensor network applications. The operating system exposes a Java programming environment enabling novice programmers to develop applications with ease. Also, in that it is an operating system, we believe that it should be usable without requiring any low level development. Thus, we have included in the operating system a default application which allows domain experts to monitor environments and configure the sampling and transmission rates of collected data.

The primary contribution of this paper is that we have designed the first operating system targetted especially for body sensor networks, which utilizes run-time compilation of bytecode to provide an efficient Java execution platform for resource constrained devices.

The remainder of this paper is structured as follows. Section II provides an overview of related work. Section III describes the motivation behind why this work is necessary. We provide an overview of the system design in Section IV, followed by implementation details in Section V. We evaluate our work in Section VI and then conclude in Section VII.

II. Related Work

Operating systems commonly used for sensor network development include TinyOS [1] and Contiki [2]. TinyOS exposes an event driven execution model programmed using a component model. TinyOS applications are programmed using the nesC language developed specifically for TinyOS. Contiki also provides an event-driven programming model, however, programs are coded in C. It is very hard for non-embedded developers to write programs for these operating systems due to the level of embedded and event-driven programming experience required.

More recent initiatives to enable Java for sensor nodes aims to facilitate an easier programming paradigm for sensor nodes [3] [4] [5] [6]. The Squawk virtual machine [4] allows for Java code to be executed on Sun SPOT devices. Sun SPOT devices have substantially more resources than that of more popular sensor nodes (especially those used for body sensing). The Squawk virtual machine has a program memory footprint of 270kB which is larger than that available for most body sensor nodes available.

Darjeeling [6] and TakaTuka [5] are two virtual machines which provide a Java interpreter for resource constrained devices. Interpreters are infamous for their high execution overheads, and as can be seen in [6], the overhead is quite large. To overcome such interpretation overheads, run-time compilation of bytecode [3] was proposed for sensor nodes, which provides substantially faster execution of bytecode compared to interpreted versions.

III. Motivation

Body sensor networks provide a different environment and paradigm to that of traditional wireless sensor networks [7]. The size of the environment (the body) to be monitored is much smaller and is geometrically the same to any other (body) deployments, unlike the environments which traditional sensor networks are deployed in (which vary from one deployment to another). This results in smaller sized networks with nodes placed in, usually, known places. A body sensor network rarely grows (or shrinks) in size. Also, once a node is placed, it is very unlikely to move. Transmission ranges of sensor nodes do not require long distances. Additionally, nodes are usually one hop away from every node in the same body sensor network. These properties of body sensor networks relieve body sensor nodes from complex MAC and routing protocols which is a primary requirement for traditional wireless sensor networks.

Although body sensor networks will consist of a smaller numbers of nodes, relieving the nodes from scalability issues, this also means that nodes will not be able to rely on neighbouring nodes for redundancy. Thus, body sensor networks require more accurate and robust sensing algorithms. More so, typical body sensor networks do not usually have the luxury of long sleep periods since events must be caught immediately due to the repercussions of missing a vital event, which in turn affects the lifetime of the sensor node. However, most body sensor network deployments do not require long lifetimes since they are usually used temporarily to analyse or detect specific conditions or applications.

Due to the geometrical, environmental and requirement similarities between different body sensor network deployments and applications, a substantial amount of application logic is common amongst body sensor network applications. Such logic includes context awareness both of the internal environment (the body) and the external environment, logging of data (either distributed or centrally), uploading of data to a pc or server either real-time or post-collection, and sending alerts when an interesting event has occurred.

Programming sensor network applications is a challenging task which most often requires experienced embedded developers to develop such applications and also may involve adopting new programming languages and models. This is primarily due to the operating systems, languages and tools which are currently used including C based operating systems [1][2].

The motivation behind our work is to provide an operating system which facilitates ease of development of body sensor network applications for novice programmers and domain experts. We plan to achieve this by focusing on the intrinsic properties of body sensor network applications. Whilst at the same time, sacrificing features required for traditional wireless sensor networks which are not inherent in body sensor networks in aim of delegating such relieved resources for ease of programmability and body sensor network specific functionality. By exposing an easy to use standard API we envisage that algorithms can seamlessly be shared, switched and compared. Thus, enabling an open development research platform for body sensor network applications.

IV. Design

One of the main motivations behind our work is to provide an easy to use programming environment for developing body sensor network applications for novice programmers. Most available tools for sensor network applications rely on C (or flavours thereof such as nesC) as a programming language. However, we have decided to opt for a Java programming environment to facilitate an easy to use programming environment. As mentioned in Section II, several initiatives have been made to port Java to sensor nodes, most of which use an interpreter to execute bytecode. As shown in [6], interpretation of bytecode suffers from great execution overheads. We have, therefore decided to utilise a run-time compiler similar to [3] in aim of offering a Java programming environment without incurring substantial execution overhead. An overview of the toolchain and operating system is shown in Figure 1. We will follow by describing each component of the toolchain and operating system.

1) Scripting: We would like to target not only programmers but also domain experts who more than likely do not have a large amount of experience programming. Thus, we have included in our architecture a means of being able to control how a body sensor network application would work on a higher level. For those with slight knowledge of programming we have provided a scripting tool which can be used to create simple applications by configuring the main execution thread using a Java like syntax without having to deal with the intrinsic properties of Java. The scripting source is passed into the Java Source Generator which creates equivalent Java source of the specified application, which can then be passed through the toolchain as normal Java source. The reason for doing this is that we can facilitate such abstractions without incurring any extra overhead on the sensor node itself.

2) Parameter Configuration: For domain experts without any knowledge of programming, we have provided a tool to configure body sensor network applications' sensing and transmission rates, which should allow domain scientists

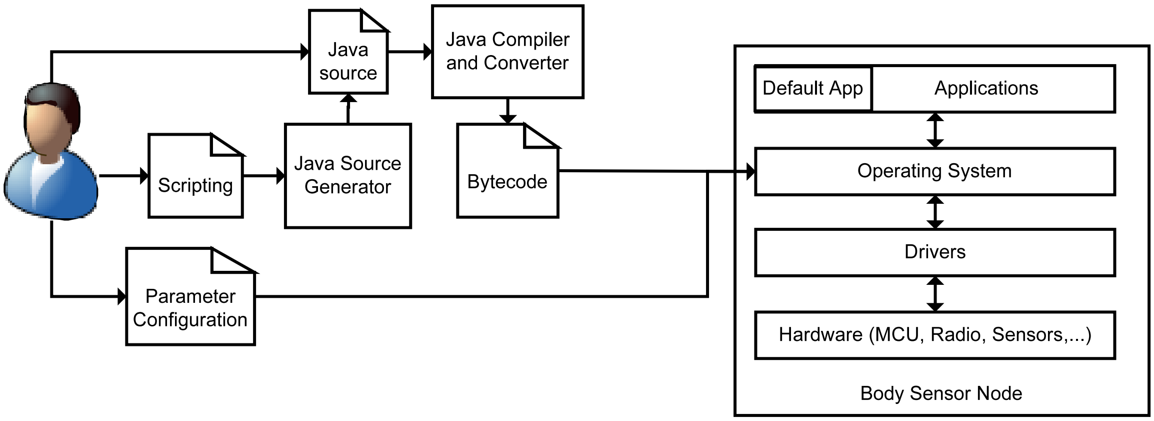


Figure 1. An overview of the toolchain and operating system.

to extract initial data before moving on to more complex applications. BSNOS is initially loaded with the Default Application. The Default Application will sample the connected sensors according to the parameters set using the parameter configuration tool. The data will then be sent to a base station at a period also defined by the parameter configuration tool.

3) Java for Sensor Nodes: Java source is passed into the Java compiler and converter which generates an altered version of Java bytecode which is more suitable for resource constrained devices. The details of this compilation and conversion process is similar to that described in [3]. The generated bytecode differs to that of standard Java bytecode in that String literals relating to class and function names are completely removed. Also, Java bytecode is based on a 32 bit stack width, which for majority of sensor network applications is not required. Thus, the converter also changes the bytecode to use a 16 bit stack width. The bytecode generated can then be disseminated to sensor nodes over a serial connection or over-the-air. The benefits of transmitting bytecode instead of native code is that bytecode is smaller in size, and thus, less energy is required to transmit code updates.

4) Applications to Hardware: At the lowest level of the BSNOS stack lies the underlying hardware including the microcontroller, radio and sensors. Driver code installed on the microcontroller will expose the different peripherals and hardware features which are required to be used. The operating system can then expose the hardware via the drivers to provide common lower level services required by applications.

5) The BSNOS Kernel: Figure 2 provides a view of the different operating system and facilitating tool components. The BSNOS Kernel encapsulates services which are essential to provide an easy to use Java programming and operating system environment. The main kernel components are outlined below:

Java Run-time Environment: The run-time compiler generates native code which mimics the Java operand stack used by the original bytecode. Due to this a Java run-time environment is required. The run-time environment exposes basic Java functionality such as array manipulation, object creation, exception handling amongst other features.

Garbage Collector: Java relieves programmers from the task of memory management. However, after objects or arrays are no longer required, their memory is still being occupied, and thus a check is required to release such memory that is no longer being used. This is the job of the Garbage Collector.

Run-time Compiler: By providing a Java programming environment, the learning curve for novice programmers to develop applications for body sensor networks will be decreased. However, since the early days of Java, it’s slow speed was notable as was summed up in [8] with their statement ”Java isn’t just slow, it’s really slow, surprisingly slow.” This was primarily due to the fact that Java’s execution platform was completely based on interpreters. In aim of providing an efficient execution platform for Java based applications, we have opted to use a run-time compiler similar to [3] which compiles the bytecode it receives to native code which the underlying microcontroller can execute without incurring any interpretation overheads.

Software Manager: Body sensor networks like other wireless sensor networks and traditional computing systems require updates from time to time due to various reasons including bug fixes, new features or complete new applications. The Software Manager enables efficient remote updating by providing dynamic loading and unloading of code at the granularity of classes or functions.

Scheduler: At the core of the kernel lies the Scheduler. The Scheduler is responsible for scheduling execution of threads and events. Figure 3 depicts how events and threads are scheduled. Interrupts (IRQ) raised on microcontrollers have higher priority than the main thread of execution. Similarly, we have designed events in BSNOS to have a higher

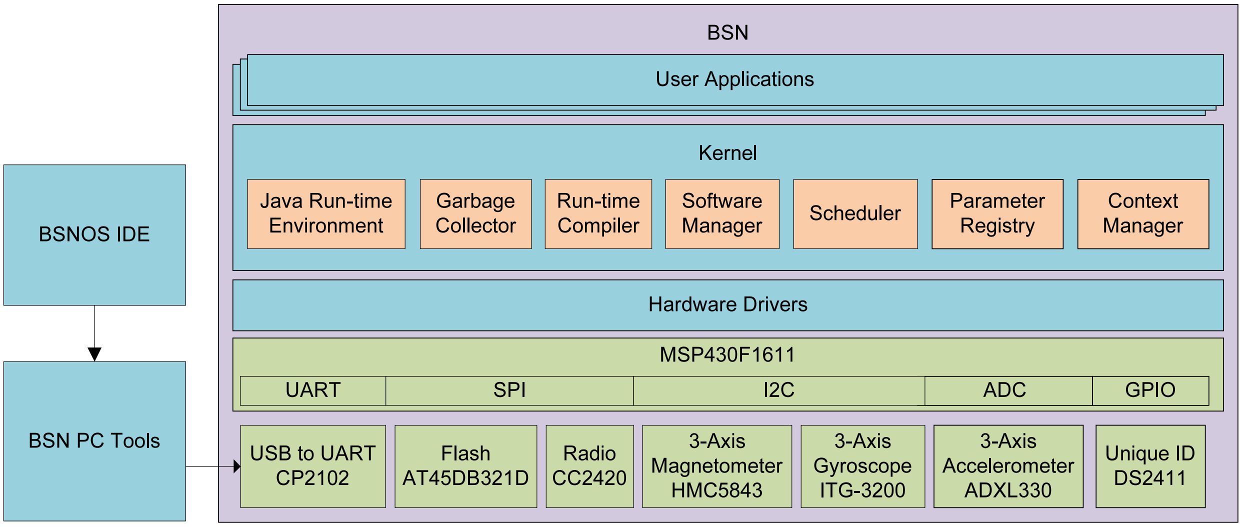


Figure 2. The operating system and facilitating tool components.

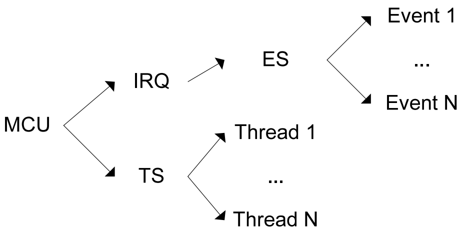


Figure 3. The scheduler execution policy is demonstrated above. Preference is given to microcontroller interrupts (IRQ) which in turn then invoke system events by the Event Scheduler (ES). The Thread Scheduler (TS) is invoked when no interrupts are pending, which in turn executes threads in a round-robin fashion.

priority than other threads since the primary aim of body sensor network applications is to monitor the external events. Thus, when an interrupt is raised the BSNOS kernel buffers any related events and then invokes the Event Scheduler (ES) which will then dispatch any events which are waiting to be scheduled.

The Thread Scheduler (TS) is responsible for scheduling thread execution. We have implemented a round-robin scheduler which provides equal execution slots to each thread in aim of providing a lightweight threading API. However, in future we will investigate whether more sophisticated scheduling techniques should be adopted.

Parameter Registry: Sensor networks application users often require to change parameters which alter the way applications would operate. It is to the best of the authors knowledge that other sensor network operating systems have however not provided any API to cater for such a common feature, leaving it up to the application developer to provide a mechanism for storing, altering and retrieving of parameters. The Parameter Registry is a storage area in the operating system whereby parameters can be stored, retrieved and updated by different applications using a standard API.

Context Manager: Context is an element that is predominant in body sensor networks. We would like to encapsulate context aware logic that is common to body sensor networks in the operating system. The Context Manager provides a built in mechanism to support applications in their contextual needs. The human body provides an environment which (usually) consists of the same structure amongst different subjects. Thus, we have created an API to facilitate developers in defining the position of sensor placement.

V. Implementation

The BSNOS kernel is written in C and compiled using Texas Instruments' Code Composer Studio v 4.0.1 for the MSP430F1611 microcontroller. As an initial target node, we have decided to port the operating system for the BSN platform [9]. A detailed view of BSNOS, hardware and facilitating tools is depicted in Figure 2. We have implemented drivers for Atmel's AT45DB321D 4MB DataFlash and Texas Instruments' CC2420 Radio which communicate with the microcontroller over SPI. The I^{2}C protocol is used to communicate with Honeywell's HMC5843 3-Axis Magnetometer and InvenSense's ITG-3200 3-Axis Gyroscope. ADC channels are used to sample the analog readings for Analog Device's ADXL330 3-Axis Accelerometer and a unique ID is provided to the operating system using Dallas Semiconductor's DS2411 chip which is communicated with using the proprietary 1-Wire protocol.

The ROM footprint for each module is listed in Table I, totalling to just over 12 KB.

We have implemented a lightweight API which provides an easy to use interface. A sample of the BSNOS API is

Table I
MODULE FOOTPRINT

Module	ROM
Run-time System	2 KB
Java Run-time	500 B
Run-time Compiler	8 KB
Threading	400 B
Class Loader	400 B
Drivers	1 KB

listed in Table II. To demonstrate the API, a sample program is listed in Figure 4.

VI. EVALUATION

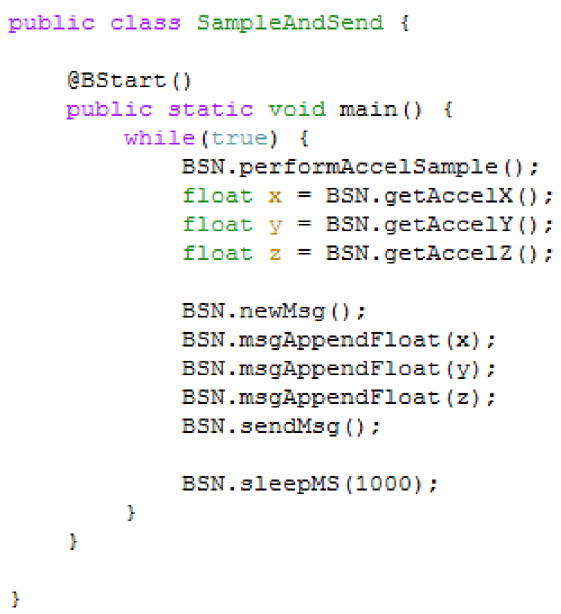
Evaluation of run-time compilation code compared with interpreted code and native code generated from C source code has been evaluated in [3]. The aim of BSNOS is to provide an easy to use platform which is focused primarily on higher data rates due to the nature of common body sensor network applications. We have conducted evaluation as regards to the maximum throughput that can be achieved using default configurations of BSNOS and TinyOS. Figure 5 depicts the maximum transmission rates of messages of size 1, 50 and 100 bytes for BSNOS and TinyOS. As can be seen from the graph BSNOS achieves a substantial increase in maximum throughput when compared with TinyOS. Obviously, however, this is due to the radio driver implementation and the minimal MAC layer that was used. Were a similiar radio implementation to be used with TinyOS, than a similiar throughput would be achieved.

VII. CONCLUSION

In this paper we have presented, BSNOS, a new lightweight operating system designed specifically for body sensor networks. We have implemented BSNOS on the BSN node as the first target platform. The aims of the operating system is to allow novice developers to focus on the specific application rather than the intrinsic properties of embedded programming. We aimed to achieve this by exposing functionality common in body sensor network applications such as parameter configuration and context. We envisage that by providing a standard API and a Java programming environment, developers would be able to seamlessly share and compare different implementations and algorithms. The contributions of this paper are as follows:

Table II
SAMPLE API

Module	Function
Accelerometer	`void performAccelSample()`
	`float getAccelX()`
	`float getAccelY()`
	`float getAccelZ()`
Context	`BodySegment getBodySegment()`
Flash	`void writeByte(byte b)`
	`byte readNextByte()`
	`void dumpFlashToSerial()`
Gyroscope	`void performGyroSample()`
	`float getGyroY()`
Parameters	`short getShortParameter(Parameter p)`
Radio	`void msgAppendFloat(float f)`
	`void msgAppendShort(short s)`
	`void msgAppendInt(int i)`
	`void sendMsgToNode(short nodeid)`
Serial	`void serialSendByte(byte b)`
	`byte serialReceiveByte()`
Threading	`void sleepMS(short ms)`
Unique ID	`long getUniqueID()`



```
public class SampleAndSend {

    @BStart()
    public static void main() {
        while(true) {
            BSN.performAccelSample();
            float x = BSN.getAccelX();
            float y = BSN.getAccelY();
            float z = BSN.getAccelZ();

            BSN.newMsg();
            BSN.msgAppendFloat(x);
            BSN.msgAppendFloat(y);
            BSN.msgAppendFloat(z);
            BSN.sendMsg();

            BSN.sleepMS(1000);
        }
    }

}
```

Figure 4. Sample source for a sample and send application which samples the accelerometer and sends the data to the base station.

- We have presented the first operating system targeted

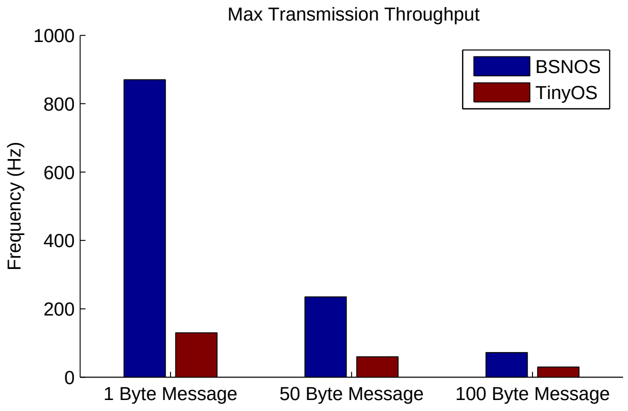


Figure 5. Maximum transmission throughput for BSNOS and TinyOS using default configurations.

for body sensor network applications.

- We have proposed a parameter registry for sensor node operating systems which releases the developer of implementing such functionality.
- We have implemented first steps towards integrating body sensor network context awareness into an OS.
- We have designed the first operating system which utilizes run-time compilation of bytecode to provide an efficient means of executing Java applications for resource constrained embedded devices.

We plan to continue development on the operating system with the following future work:

- We envisage that domain experts would be able to write body sensor network applications using a GUI block programming environment. Thus, we will investigate into integrating such tools.
- We have currently implemented a round-robin thread scheduling policy, however would like to investigate into more sophisticated policies and the associated tradeoffs and benefits.
- The current implementation of the Context Manager exposes functionality to easily define and share context amongst code and applications. We would like to further investigate whether we can integrate commonly used algorithms into the operating system or as system libraries that can be dynamically loaded.

Acknowledgment

We would like to thank James Patterson for the endless discussions regarding many aspects of the BSN node and support in testing the framework. This work was supported by the ESPRIT project funded by EPSRC.

References

[1] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, “TinyOS: An operating system for sensor networks,” in *in Ambient Intelligence*, 2004. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.129.7716

[2] A. Dunkels, B. Grönvall, and T. Voigt, “Contiki - a lightweight and flexible operating system for tiny networked sensors,” *Local Computer Networks, Annual IEEE Conference on*, vol. 0, pp. 455–462, 2004.

[3] J. Ellul and K. Martinez, “Run-time compilation of bytecode in sensor networks,” in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, 2010, pp. 133 –138.

[4] D. Simon, C. Cifuentes, D. Cleal, J. Daniels, and D. White, “Java on the bare metal of wireless sensor devices: the squawk java virtual machine,” in *Proceedings of the 2nd international conference on Virtual execution environments*, ser. VEE ’06. New York, NY, USA: ACM, 2006, pp. 78–88. [Online]. Available: http://doi.acm.org/10.1145/1134760.1134773

[5] F. Aslam, L. Fennell, C. Schindelhauer, P. Thiemann, G. Ernst, E. Haussmann, S. Rührup, and Z. Uzmi, “Optimized Java Binary and Virtual Machine for Tiny Motes,” in *Distributed Computing in Sensor Systems*, R. Rajaraman, T. Moscibroda, A. Dunkels, and A. Scaglione, Eds., vol. 6131. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–30. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13651-1_2

[6] N. Brouwers, P. Corke, and K. Langendoen, “Darjeeling, a java compatible virtual machine for microcontrollers,” in *Proceedings of the ACM/IFIP/USENIX Middleware ’08 Conference Companion*, ser. Companion ’08. New York, NY, USA: ACM, 2008, pp. 18–23. [Online]. Available: http://doi.acm.org/10.1145/1462735.1462740

[7] G.-Z. Yang, *Body Sensor Networks*. Springer-Verlag New York, Inc., 2006.

[8] P. Tyma, “Why are we using java again?” *Commun. ACM*, vol. 41, no. 6, pp. 38–42, 1998.

[9] B. Lo, S. Thiemjarus, R. King, and G. Yang, “Body sensor network - a wireless sensor platform for pervasive healthcare monitoring,” in *Proceedings of the 3rd International Conference on Pervasive Computing (Pervasive 2005)*, 2005, pp. 77–80.

The KPI-Based Reputation Policy Language

Slim Trabelsi
SAP Research
Mougins, France
slim.trabelsi@sap.com

Luca Boasso
Politecnico Di Torino
Torino, Italy
luca.boasso@studenti.polito.it

***Abstract*—Trust policy languages are implemented to express the trust requirements of the users. These requirements are represented by a set of rules specifying the necessary conditions that should be fulfilled by an entity in order to gain the trust of the evaluator. Most of the known trust policy languages are designed to express credential, authorization and access control requirements for the trust establishment. The credential based approach represents only one aspect of trust. The other main aspects like reputation and recommendation are not covered by these policy languages. In this paper we propose a new policy language for expressing trust requirements for reputation models, and particularly for the KPI-based reputation model in a supply chain scenario.**

***Index Terms*—Trust, Reputation, Supply, Policy, Language, KPI,**

I. INTRODUCTION

Trust is a subjective matter, it depends on a truster's subjective evaluation of past experiences and it depends on the characteristics of the trustee [1]. Still, for making trust usable, we need to be able to express it in terms of measurable quantities (trust metrics and reputation). These metrics will be clearly not able to capture the many subjective and context dependent facets of this complex sociological phenomenon, but still they may be used in a specific context to assess the reliability of the trustee and its capability to ensure privacy, security, and so on. Nowadays, the notion of trust does not rely only on the traditional trust infrastructure (based on certificate verification or recommendation) but it is strongly related to the behavior of the users and their virtual reputation. The traditional trustworthiness models implemented in the internet, such as Amazon or E-Bay, are relying on a subjective rating system in which users estimate the "quality" of the transaction over a numerical scale. Knowing that nobody is able to formalize and explain the difference between two successive values like a transaction rewarded at 9/10 and another one 10/10, we cannot really estimate the correctness and the objectivity of the trust and reputation value. In addition, such trustworthiness models are limited in terms of federation and adaptability; in fact it is very hard to adapt the perception of trust in different domains and conditions. For example the reputation of a transporter cannot be exported to packaging and storage domain, because there is no possibility to map the subjective trustworthiness values between two different domains with two different trust perceptions. In order to address the limitations above, we proposed a less subjective trust model taking into account quantifiable parameters for the computation of the trustworthiness value of an entity [11]. We called these parameters KPI for Key Performance Indicators. In the context of trust, the goal of using these metrics is to quantify the sources of trust and adapt them to the personal perception of each trusting entity. For example if a shipment and a distribution company were sharing the same KPI parameters for their reputation model, like transportation time, package quality, quality of goods, price etc. the federation of trust between these two domains can be handled in an easy way by adapting the reputation calculation according to the local perception of trust. Each trustee in the supply chain can configure a pattern for his trust model according to his objectives and his trust perception expressed through a formal language, for example the trusting entity that prioritizes the delivery time of a good, will obtain a different reputation value than another user that prioritizes the CO2 footprint.

In this paper, we propose a new reputation policy language for expressing easily the personalized trust requirements related to the KPI-based trust model. There are few languages describing the reputation-based trust models, and in the majority of the cases they are not designed for non-expert users, therefore they are far from being user friendly. On the other hand, an increasing number of people are starting to use trust models in the supply chain industry in order to evaluate the trustworthiness of the different nodes of the chains. Most of the time users in that domain are not necessarily security experts, and they encounter major obstacles in the configuration and personalization of reputation models. For this reason we propose here a user friendly policy language, able to express in a very simple way the trust requirements of a user who wants to evaluate the reputation of an entity according to its performance parameters. The long term goal of this language is not limited to KPI based trust model, but to support most of the behavioral trust models.

This paper is organized as follows: in Section 2, we discuss the related work in the domain of Trust and reputation policies, in Section 3, we present a brief use case scenario, in Section 4, we define the KPI-based reputation model, in Section 5, we present the policy language specification, then we briefly provide the implementation details, and finally we conclude our work.

II. Related Work

In the literature, the two aspects of policy based and reputation based management are usually separated [9]. On one side, the policy based trust management focuses on problems related to authorization and access control in open systems; i.e., it determines whether or not an unknown user can be trusted, based on a set of credentials and on a set of policies. On the other side, reputation-based management assesses the trust relationships based on non-certified available information, like recommendations or previous experiences of other users. In this paper we show how we can merge these two approaches by providing a policy language that expresses the trust requirements from a reputation model.

Among the existing trust policy languages we can mention TPL (Trust Policy Language) [6] that is a XML-based language defining the relation between unknown entities to roles. It expresses a mechanism that allows a business to define a policy to map accessed users to roles, based on certificates received from the user and collected automatically by the system. The XML nature of the language makes it appropriate for automated processing, but less suitable for human users. Bonatti and Samarati [5] proposed the PSPL language to regulate service access and information release in large scale networks. This language is designed to express access and release policies in conjunction with a policy filtering mechanism, which allow the parties to exchange their requirements in a compact and privacy preserving way. PSPL has a Prolog-like syntax, in which one can define rules that take into account the elements of the trust model. Blaze et al. proposed the KeyNote policy language [8] that provides a simple notation for specifying both local security policies and security credentials that can be sent over a non trusted network. KeyNote policies and credentials, called "assertions", contain predicates describing the trusted actions permitted by the holders of specific public keys. KeyNote assertions are small and structured programs written in a simple notation based on C-like expressions and attribute/value pairs actions.

All these trust policy languages do not support the reputation models. And, to our knowledge there are very few studies trying to address the expressivity of the reputation requirements by a policy language like TriQL.P [10] that plans to propose a reputation dedicated language, but up to now few results are available from this project .

III. Use case

To illustrate our approach, let us consider a simple supply chain use case. Let us consider an active transport tracking devices attached to returnable transport items, such as crates, rolling containers, pallets and shipping containers. Consider a shipment of milk as it travels from the farm near Rennes in France to a supermarket distribution center in Paris. After collecting the milk in the farm, the farmer has to use a small tank truck to carry his daily production to the local milk collecting center. There the milk is packaged then assembled to pallets and finally charged up to huge transportation trucks. The trucks chip the bricks of milk to the supermarkets in Paris. In order to monitor and evaluate the quality of the transportation process from the farm to the distribution center the supermarket quality manager will setup a KPI requirement list in which he defines the all the quantifiable thresholds that should be satisfied during the entire process. The metrics chosen by the quality manager are for example the transportation time between the farm and the local collecting center, the average temperature, the packaging time and cost, the transportation time between the packaging factory and the supermarket in Paris, the average temperature during the transportation etc. All these indicators are provided by tamperproof sensors. The quality manager usually defines KPIs that represent his business objectives and compute a *reputation* score for each actor in the chain according to the compliance with the requirements described above.. For example a *good* temperature average should be between 3 and 4 degrees Celsius. The transportation time between Rennes and Paris should be 5 hours (more is bad, less is good), etc.

For each delivery day the manager collects from different sensors the indicator values and integrates it to the reputation model in order to evaluate the score of each actor contributing to the chain. This example is quite trivial, and any manager can compute the score with a spreadsheet. The problem becomes serious when in real cases, some managers have to take into account a large number of KPIs. These performance indicator values may be gathered through different sensors located in different places and communicating via different protocols (for example in a remote database, in a XML file from a web service…). In that case the user should have a wide range of computer skills just to collect and convert the values in a suitable format and then to compute the reputation values. This is not always the case. For this reason we define in this paper a formal model to collect KPI values and compute the final score, as well as an easily accessible policy language that expresses these requirements.

IV. KPI-Based Trust Model

We propose a KPI-based trust Model (KPITM) [11] as an approach in which trust evaluation is based on KPIs shared by different users.

A. Repuation model

The KPI-based reputation model takes into account trust metrics based on KPI. In this approach, a user can express his trust preferences via an expressive language that specifies the sources of the performance indicators factors and how the factors should be combined to obtain a reputation score. According to his business objectives, the user is able to prioritize some indicators by setting a strong weight affecting the result of the trust score. These indicator values are then normalized (between 0 and 1) and then aggregated in order to obtain a unified reputation value.

The normalization rule is written as follows:

$$\begin{cases} 1 & \text{if } K_i > K_{max} \\ 0 & \text{if } K_i < K_{min} \\ \dfrac{K_i - K_{\min}}{K_{\max} - K_{\min}} & \text{otherwise} \end{cases}$$

Higher is better KPI normalization

$$\begin{cases} 0 & \text{if } K_i > K_{max} \\ 1 & \text{if } K_i < K_{min} \\ \dfrac{K_{\max} - K_i}{K_{\max} - K_{\min}} & \text{otherwise} \end{cases}$$

Lower is better KPI normalization

Where K_i is the measured performance indicator valueK_{min} and K_{max} are the minimum and maximum values declared in the objectives scale. If lower values are better (e.g., the delivery time example) the value is reversed by subtracting it from 1.

In the KPI-based model, each item has a semantic meaning that explains the context of the measured value related to any performance parameter (e.g., delivery time, temperature, CO2 impact, etc.). The combination of different KPI items offers the possibility to the trustee to customize his trust evaluation by expressing complex semantic queries.

An entity that wants to connect to a KPITM system in order to evaluate the trust of another entity has to select three elements: first, the KPI items that are relevant for him, second, the location where to find the performance indicator value and, lastly, the weights of each KPI in order to prioritize some values during the trust evaluation. All this information must be contained in the core query sent to the KPITM engine that will automatically connect to the different sources, get the values of each item and compute the trust value.

The KPI-based trust model offers the possibility to quantify the trustworthiness according to some domain specific objectives (how should be the conservation temperature range for the milk) and it permits to any trustee entity to determine, which tested element is more trustworthy according to an objective estimation. In particular, our KPI-based trust model allows a trustee to evaluate the weight of a recommendation by applying the business objective scale of the recommender.

More formally, the KPI-based trust model used is composed of three complementary layers:

- Performance Indicator Values: they are collected from the different sources providing the values related to the performance items
- Business Objectives Scale: it is defined by the trustee according to the performance indicators related to their business objectives. An interval of values (min and max) must be chosen for every performance indicator in order to normalize the measured value with a [0, 1] scale. Furthermore a weight factor must be defined to prioritize the performance indicators and to compute the final trust value.
- Trust Level Value: it is the aggregation of all the normalized performance indicators plus, possibly, some external values like the recommendation from other trusted entities.

For example in our scenario these layers are represented in Figure 1, the weight factors are within the circles, whereas the interval of values is represented by the double ended arrow in the same layer.

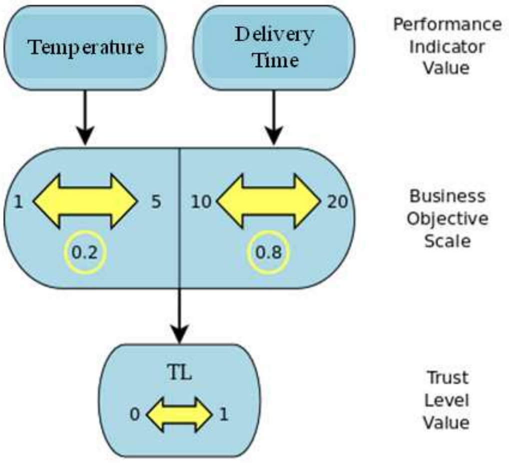


Fig1. KPI-based Trust model of our scenario

B. *Architecture*

We proposed a loosely coupled architecture (Figure 2) for managing the KPI based trust in which we have three independent and complementary layers:

- The Indicator sources: we proposed two kinds of interfaces in order to collect the indicator values that should be used to compute the reputation: a database connector used as an interface to get access to any kind of local or remote database. A Web Service interface for collecting the indicator values published as Web Services
- The KPITM engine in charge of computing the reputation value according to the trust model described in the previous section. It interprets the queries sent by the user via a UI or the policy language, and then it uses the collected indicator values to compute the reputation.
- KPI-based reputation language engine: this component interprets the queries written in the policy language (human readable language used to express the reputation/trust requirements) and translates it into a remote call to the KPITM engine in order to calculate the reputation value.

The choice of this kind of decoupled architecture is motivated by the requirement to have a generic solution independent from the trust/reputation model and from the sources of trust. In this paper, we describe a specific case, where the trust model is based on the KPIs only, but in other

ISBN: 978-1-61208-144-1

cases, we may use another reputation model with other sources of trust, and we want to develop a language generic and flexible enough to be used with a wide range of trust and reputation models. Each layer of this architecture is independent and replaceable.

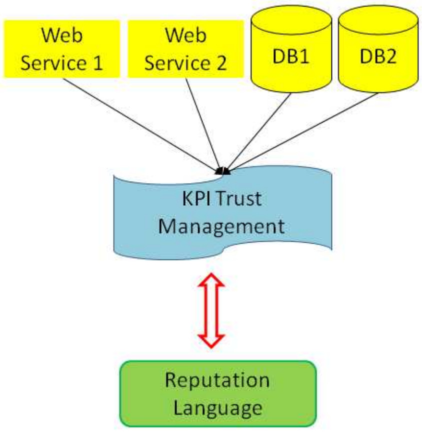


Fig2. Architecture

V. KPI-Based Policy Language

The reputation language allows a user to define queries for the KPITM engine in a simple and concise way. No programming knowledge is required since we propose a script based language.

Referring to our scenario, the following code calculates the trust value of an actor in the chain that we will cal FARMER for example:

```
? example
Actors[Farmer] -> Delivery (20:30:0.8)
               -> Temperature(3:5:0.2)
```

According to our trust model, this query specifies the KPI that are relevant to the manager (*time* and *temperature*), the location of the performance values (location Actors for the actor *Farmer*) and finally the range of acceptable values and the weight for each KPI (20:30:0.8, i.e. delivery time can vary between 20 and 30 minutes with a weight of 0.8).

A. Language Specification

A formal language is a set of sequences of symbols. Elements of this set are called sentences. In the KPI language sentences are programs called scripts. The symbols originate from a finite set called the vocabulary. The set of programs (which is infinite) is defined by rules of their composition. Sequences of symbols that are composed by these rules are said to be syntactically correct or well-formed. The set of rules is the syntax of the language. The program (or sentence of the formal language) consists of parts called syntactic entities, such as declarations, statements or expressions.

Parentheses may be used to group factors or terms. The notation introduced here is known as Extended Backus-Naur Formalism (EBNF) [2].

Besides syntactic entities, denoted by identifiers, we need to substitute elements, also called tokens, taken from the formal language's vocabulary. The vocabulary of the KPI language consists of identifiers, numbers, strings, operators, delimiters and comments. They are called lexical symbols and are composed of sequences of characters. (Note the distinction between symbols and characters.)

In the EBNF notation non-terminal symbols are denoted by English words expressing their intuitive meaning. Terminal symbols are denoted by strings enclosed in quote marks.

B. Lexical Analysis

The representation of terminal symbols in terms of characters is defined using the Latin-1 set. Terminal symbols include identifiers, numbers, strings, operators, delimiters and comments. Blanks and line breaks must not occur within symbols (except in comments and blanks in strings). They are ignored unless they are essential to separate two consecutive symbols. Capital and lower-case letters are considered as being distinct.The lexical rules are now considered in detail:

i. An identifier (*ident*) starts with an upper-case letter followed by a sequence of zero or more letters or digit or the special character "_":
Examples: Actors, Temperature, green_car10

ii. Numbers are of type real, a sequence of digit followed by an optional decimal part:

```
real = digit {digit} [ "." digit {digit} ].
```

Examples: 3.14, 8, 6.33.

iii. A string is a sequence of characters enclosed in quotation marks. A string cannot contain the delimiting quote mark:

```
string = '"' {character} '"' | "'" {character} "'".
```

Examples: "This", "is 'a'", 'short "string"'.

iv. Operators and delimiters are the special characters, character pairs or reserved words listed below. These reserved words cannot be used as identifiers.

v. Comments start with a hash character "#" that is not part of a string and ends at the end of the physical line.

C. Syntax and Semantic

A script begins with an optional chart declaration followed by a sequence of statements:

```
script = [chart] statement {statement}.
```

There are two kinds of statements, assignment and query:

```
statement = assignment | query.
```

1) Assignments

An assignment allows the creation of a variable with a value given by an expression:

```
assignment = "var" ident "=" expression.

expression = ["+"|"-"] term { ("+"|"-")
term}.

term = factor {("*" | "/") factor}.

factor =  real | ident |  "(" expression
")".
```

The above rules specify that an expression can use the mathematical operators for addition, subtraction, multiplication and division. These operators have the usual

ISBN: 978-1-61208-144-1

arithmetic precedence and they are left-associative. The parentheses are used to group expressions and they have the highest precedence. Variables can appear freely in an expression. For example the following assignments are correct and all the expressions evaluate to 8:

```
var a = 4 + 3 * 10 / 5 - 2
var b = 4 + 3 * (10 / 5) - 2
var c = (4 + 3 * 10 / 5) - 2
```

2) *Query*

A query allows the creation of a list of KPIs:

```
query = "?" ident  item {item}.
```

It starts with a question mark character followed by ident that represents the name of the list. A KPI is defined as an item:

```
item = ident "[" ident {"," ident } "]" type
{type}.

type = "->" ident "(" min_max ":" min_max  ":"
weight ")".

weight = expression | "/".

min_max =  expression | "MIN" | "MAX".
```

The above rules define the exact syntax of a KPI. Some examples from Table 1 will make these rules clear. Consider this script:

```
? Example1 Actors[Farmer] -> Delivery (20:30:1)
```

In order to take into account the KPI of type Delivery a second item is added:

```
? Example2

Actors[Farmer] -> Delivery (20:30:0.6)
Actors[Farmer] -> Temperature(3:5::0.4)
```

Notice that the sum of all the weights in the list must be always one. It is possible to add KPIs with different name, type, location, min, max and weight; they are fully customizable as shown in the following example:

```
? Example3

Actors[Farmer] -> Delivery(20:30:0.6)
Actors[Packaging] -> Delivery(40:60:0.2)
Finance[GOOG] -> Price(30:100:0.2)
```

In Example2 there are two KPIs with same name (Farmer) and location (Actors). That script can be written in this equivalent form:

```
? Example4

Actors[Farmer] -> Delivery(20:30:0.6)
                   -> Temperature(3:5:0.4)
```

Let’s suppose that a company wants to evaluate the reputation of two Actors (Farmer and Packaging) based on their delivery time and temperature. A possible script can be:

```
? Example5

Actors[Farmer] -> Delivery(20:30:0.3)
                   -> Temperature(3:5:0.2)
Actors[Packaging] -> Delivery(20:30:0.3)
                      -> Temperature(3:5:0.2)
```

This last example can be rewritten also as:

```
? Example6

Actors[Farmer, Packaging] -> Delivery(20:30:0.6)
                             ->Temperature(3:5:0.4)
```

In fact the four KPIs share the location (Actors) and taken in pair they share also the type, the min, the max and the weight (see Example5). In this way it is enough to list the names inside the square brackets and write the shared part only once. It is important to notice the changes in the weights: the language will split 0.6 and 0.4 like in Example 6 automatically.

3) *Automatic weight and MIN MAX keywords*

Since the sum of the weights for all the items must be equal to one, it is possible to specify only the weights of interest and let the language to calculate the others. To achieve this result use the "/" symbol:

```
? Example7

Actors[Farmer] -> Delivery(20:30:0.6)
Actors[Packaging] -> Delivery(3:5:/)
Actors[Supermarket] -> Delivery(3:5:/)
```

The last two items have a weight of 0.2. So the Example 5 can be rewritten again like this:

```
? Example8

Actors[Farmer, Packaging] -> Delivery(20:30:0.6)
                             -> Temperature(3:5:/)
```

Sometimes can be convenient to use the keywords MIN and MAX:

```
? Example9

Actors[Farmer] -> Temperature(MIN:5:0.6)
               -> Delivery(20:MAX:0.4)
```

The MIN will be replaced by the lowest value of type Delivery in the Actors location. Accordingly to the table defined before this value is 1. The same reasoning applies for MAX, its value is 45.

As discussed before a script contains a sequence of statements so more than one query (and so KPIs lists) can be written:

```
? Farmer
Actors[Farmer] -> Delivery(20:30:0.8)
               -> Temperature(3:5:0.2)

? Packaging
Actors[Packaging] -> Delivery(20:30:0.8)
                  -> Temperature(3:5:0.2)
```

The implementation of the language displays a list of query's name sorted by their resulting trust value:

```
Farmer: 0.95
Packaging: 0.4
```

4) *Graphical charts*

In our prototype language implementation, we also support some essential graphic function. If a script starts with a chart declaration a graphical representation of the results will be displayed. The chart syntax is the following:

```
chart = "Charts" ":" chart_desc {chart_desc}.

chart_desc = "Pie"  [ "{" pie_option {pie_option}
"}" ] |
             "Bar"  ["{" bar_option {bar_option}
"}"] .
```

A chart declaration starts with the keyword "Charts" followed by ":" and a sequence of chart's descriptions (chart_desc). A chart_desc starts with the keyword "Pie" or

"Bar" followed by an optional sequence of options. For a pie chart the possible options are:

```
pie_option =  "title" "=" string |
              "legend" "=" bool |
              "tooltips" "=" bool |
              "3d" "=" bool.
```

While for a bar chart are:

```
bar_option = "title" "=" string |
             "xlabel" "=" string |
             "ylabel" "=" string |
             "horizontal" "=" bool |
             "legend" "=" bool |
             "tooltips" "=" bool |
             "3d" "=" bool.
```

The following example will display the same charts as before but with a 3D effect and with other options enabled:

```
Charts: Pie  {title = "Pie Summary" 3d = true
legend = true tooltips = true}
             Bar  {title = "Bar Summary" 3d = true
xlabel = "Actors"
                ylabel = "Score" tooltips = true}
```

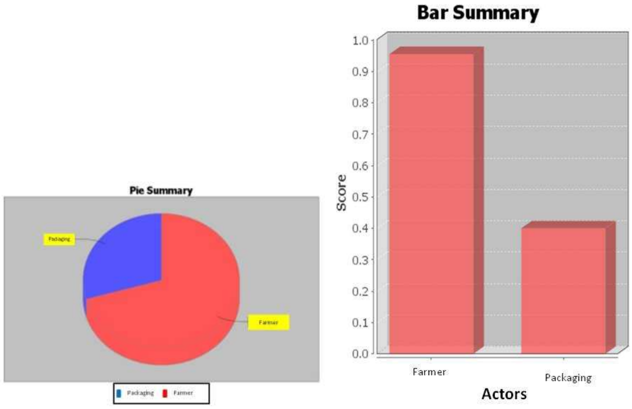


Fig3. Graphical pie and bar charts

D. *Implementation Details*

We implemented a syntax directed interpreter that parses the script given as input and translate it in an internal data structure (a list of objects). In order to execute a script this list is further processed and finally the interpreter uses it to query the web service to collect the results and displays them.We used the ANTLR parser generator [3] to build the parser and the JFreeChart library to create and display the charts[4].

VI. CONCLUSION

In this paper, we presented a new policy language for expressing reputation requirements in the context of supply chain scenarios. This language is now compatible with the KPI-based trust model but can be extended to the other reputation models. Using this new language one can easily specify the location of input sources (sensors) of the reputation model and configure its perception of trust. A visualization script is also integrated to the language in order to represent graphically the results. This language is in his initial phase and we are currently enhancing its capabilities. The first extension will be the support other reputation models than KPI like for example eBay/OnSale or Sporas & Histos [12] models.

VII. ACKNOWLEDGMENT

This work is done in the context of the FP7 EU project TAS3.

REFERENCES

[1] Gambetta, D. "Can We Trust Trust?", in Trust: Making and Breaking Cooperative Relations, Basil Blackwell, 1988

[2] Niklaus Wirth, "What can we do about the unnecessary diversity of notation for syntactic definitions?" CACM, Vol. 20, Issue 11, November 1977, pp. 822–823.

[3] http://www.antlr.org/

[4] http://www.jfree.org/jfreechart/

[5] P. Bonatti and P. Samarati, “Regulating Service Access and Information Release on   the Web,” 7th ACM Conference on Computer and Communications Security, Athens, Greece, November 2000.

[6] A. Herzberg, Mihaeli, Y. Mass, D. Naor, and Y. Ravid,“Access Control Meets Public Key Infrastructure, Or:Assigning Roles to Strangers,” IEEE Symposium on Security and Privacy, Oakland, CA, May 2000.

[7] E. Bertino, S. Castano, and E. Ferrari, “On Specifying Security Policies for Web Documents with an XML-based Language,” Sixth ACM SACMAT, Chantilly, Virginia, May 2001.

[8] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis, “The KeyNote Trust-Management System,” RFC 2704,September 1999.

[9] P. Bonatti , C. Duma , D. Olmedilla , and N. Shahmehri, “An Integration of Reputation-based and Policy-based Trust Management” In proceedings of the Semantic Web Policy Workshop (in conjunction with 4th International Semantic Web Conference), Galway, Ireland, November 2005

[10] C. Bizer, R. Cyganiak, O. Maresch and T. Gauss, “TriQL.P - Trust Architecture” http://www4.wiwiss.fu-berlin.de/bizer/triqlp/

[11] K. Bohm, S. Etalle, J. den Hartog, C. Hutter., S.Trabelsi, D. Trivellato, and N. Zannone, “A Flexible Architecture for Privacy-Aware Trust Management” Journal of Theoretical and Applied Electronic Commerce Research ISSN 0718–1876 Electronic Version VOL 5 / ISSUE 2 / AUGUST 2010 / pp. 77-96

[12] Zacharia, G. and Maes, P., “Trust Management through Reputation Mechanisms”, Applied Artificial Intelligence 14 (2000) pp. 881–907.

Integration of RFID and Wireless Sensor Networks into a Supply Chain Management System

Laurent Gomez,
SAP Research France
Mougins, France
laurent.gomez@sap.com

Maryline Laurent, Ethmane El Moustaine
CNRS Samovar UMR 5157,
Institut Telecom, Telecom SudParis
Paris, France
Maryline.Laurent@it-sudparis.eu,
Ethmane.ElMoustaine@it-sudparis.eu

***Abstract*— Wireless Sensor Networks together with Radio Frequency Identification are promising technologies for supply chain management systems. They both provide supply chain players with goods tracking and monitoring functions along the chain. Whereas RFIDs are rather focusing on identification of goods (e.g., identification, classification), WSNs are meant to monitor and control the supply chain environment. Nevertheless, despite the interest for the supply chain management systems, their integration is often deterred due to the lack of interoperability. In this paper, we propose a software framework which makes easier the integration of both RFIDs and WSNs into supply chain management systems.**

Keywords - RFID; Wireless Sensor Network; Supply Chain Management

I. INTRODUCTION

A. Context

Both WSN (Wireless Sensor Networks) and RFID (Radio Frequency IDentification) are very promising technologies for supply chain management as they introduce automation of product tracking and monitoring, and as such, they reduce the cost of the whole supply chain [19]. Today, there are only few attempts, mostly from industrial initiatives, that aim at integrating WSN and passive RFID tags [7, 8, 11].

Similarly, in the scope of the RESCUEIT project [14], we address the integration of RFIDs and WSNs into supply chain management systems. Together with major industrials such as K+N [15], Group Casino [16], REWE [17], Dr Oetker [18], we identify a clear need of secure and efficient tracking of goods along the supply. Supported by the integration of WSNs and RFIDs either within packaging or at pallet level, good tracking empowers supply chain players with tools for risk assessment along the supply chain, but also with mechanisms for identification of responsibility in case of incident.

B. Motivation

The benefit of such integration for supply chain management could be significant as data collected from RFID and sensors are complementary and could serve for enriching the range of supported services like monitoring and real time traceability. Whereas RFID is rather focusing on identification of goods (e.g., identification, classification), WSNs are meant to monitor and control the supply chain environment. To some extent, RFID are not restricted to unique identification of goods along the supply chain, but can be associated to information related to the classification, and dangerousness of goods. Based on those classifications, and with regards to the regulations (e.g., safety [20], quality), the handling, storage, and transport constraints are identified. In this context, WSNs are meant to enforce those constraints (e.g., incompatibilities with other products, flash points). Based on the sensed supply chain context at runtime, sensors tend to evaluate mismatches between the constraints defined by regulations and the current context. Any mismatch is therefore reported to the supply chain management system as a risk of incident.

As depicted in Figure 1., we propose a distribution of sensors into the supply chain, as foreseen in RESCUEIT [14]. Figure 1. shows an integration of RFID and sensor nodes only at pallet level. This assumption has been validated with end users of the project, like K+N [15]and the Casino Group [16]. In this case, we are addressing only low valuable products (e.g., household, gardening products), which explain the fact that neither goods nor packages are equipped with RFID or sensors. Nevertheless, goods and packages are still identified with barcode. In addition, depending on the value of the product, RFID and sensor can be integrated either at package or even product level.

ISBN: 978-1-61208-144-1

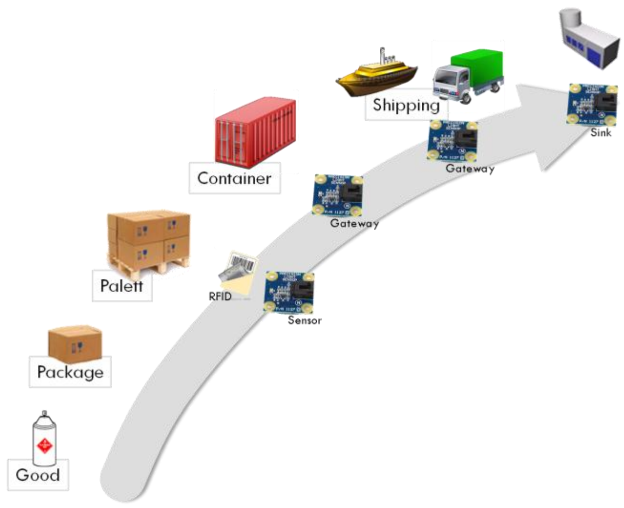


Figure 1. RFID and WSN integration

C. *Problem statement*

Even though the integration of RFID and WSNs appears to be promising for supply chain management systems, it can be deterred due to a lack of interoperability. First they are considered as disjoint networks. The EPCglobal network is designed for supporting RFID only; there is no standard targeting the integration of RFID and WSN into one network. Second, from a technological point of view, RFID are rather supporting single-hop communication between tag and reader, while WSNs supports a larger variety of communication (e.g., single to multi-hop communications). Third, they have different objectives. RFID are used for products identification and traceability. They rely on the information stored within the EPCGlobal infrastructure [10]. This information is related to the characterization of tagged assets. On the opposite, sensors have the objective to control and monitor the physical environment of assets, but not specifically associated to assets. Moreover they are able to self-store the useful information and regularly report that information to some supervision platform.

Due to different technological capabilities and uses and different objectives, a direct communication between RFID and sensor nodes is barely conceivable. That is why, we propose a software architecture that supports both RFID, especially very low-cost RFID systems, and WSN with the objectives to control, supervise and monitor both the environment of assets and the assets themselves.

D. *Overall Approach*

Our technical approach aims at integrating both WSN and RFID at the software layer. For that purpose, we propose to use EPCIS (Electronic Product Code Information Services) for RFIDs management and a mediation layer for the WSNs management. EPCIS is usually in charge of giving the list of all the features associated to an RFID product. In order to ease the interaction of RFID and sensor nodes, we use a mediation layer, Middleware for Device Integration (MDI), a SAP Research prototype. It serves for the activation of adequate monitoring and controlling mechanisms on the sensors, and to implement interactions with RFID.

As described previously, RFID and sensor are meant to be associated to a single product, package or pallet. In that context, RFIDs have two major roles: (i) identifying automatically goods along the supply chain - as made possible by the EPCglobal standard [10] for RFID but not with sensors - and including their classification, (ii) and tracing RFID for keeping sensors informed of the activity along the supply chain (e.g., storage, transport). As such, RFID are used for their fundamental identification and traceability functions, but also through EPCIS data basis to automatically raise the awareness of sensors about some (transport, storage) constraints (e.g., humidity, temperature) for the tagged good, with respect to the effective regulations. In addition, the traceability function enabled by RFID makes it possible to dynamically inform the sensor of any change of activity in the supply chain (e.g., transport, storage), so sensors can automatically adapt its constraints. Together with classification of goods, and update on supply chain activities, sensor nodes can therefore adapt their monitoring and controlling mechanisms according to regulations.

In this context, both EPCIS and MDI are considered as agents, each having interactions with RFID tags and sensors. As such, this architecture is of great help to make RFID and sensors systems interact through the MDI which handles data from RFIDs and sensors. Note that this approach is very attractive as no new integrated nodes are needed since the whole collaboration process between RFID and sensors is done at the software layer.

E. *Outline*

The remainder of this paper is organized as follows. We present the related work in section II. We describe our approach in section III, together with a architecture for the integration of RFIDs and WSNs into supply chain management systems. Finally, we conclude in section IV, with an outlook on our ongoing activities.

II. RELATED WORKS

To the best of our knowledge, integration of RFID and WSN did not attract the interest of the research community. RFID and sensor networks are both under development but with no interactions in between. Sensors are generally an open system contrary to RFID passive tags that are generally considered as a closed system where modification of their behavior is barely possible. Moreover, whereas sensors support multi-hop communications, current RFID systems are affording only single-hop communications, so no communication among tags is possible. Many issues are to be solved before integrating RFID technology with sensor networks. These issues include protocol design, energy consumption, security and privacy [13].

Moreover, the RFID technology benefits from the standard framework EPCglobal Network [10] that defines a network architecture for managing the RFID product information. This framework serves for trading partners to share product data. It is based on several key network elements including: EPCIS (EPC Information Services) and ONS (Object Naming Service). EPCIS hosts and facilitates access to serialized product information generated by EPC-tagged products. ONS is a centralized authoritative directory that routes information requests about particular EPCs likely to the well known Domain Name System (DNS).

The integration of RFID technology and wireless sensor networks provides promising perspectives as both of them can be used for tightly coupling the physical and virtual worlds.

Few research works are targeting integration of RFID passive tags and sensors. We distinguish two types of integration: hardware and software. On the hardware side, several commercial solutions are provided, which integrate RFID and WSNs in a single device (see Section II.A). On the other side, and similar to our approach, existing research works address the integration of RFID and WSNs at the software level (Section II.B).

A. *Hardware integration*

Integrating sensors into an RFID tag is used only for sensing few parameters. These sensor tags can be passive, semi-passive or active depending on the monitored phenomena. Generally, they only support single-hop communications.

The sensor passive tags draw power from the reader to perform operations. They are activated only when sensing data are required. They can be used to sense photo detection [4], acceleration [5], temperature [4, 6], etc.

The semi-passive sensor tags contain an on-board battery that is exclusively utilized for computation (and not for communication). They can perform measurements independently of the reader, but the energy supplied by the reader is always necessary for communication.

SensTAG [7] is a highly configurable semi-passive sensor tag. It operates at 13.6 MHz, and can be used to monitor a variety of sensor options, like temperature, humidity, strain, shock and light, etc.

The sensor active tags contain a battery from which they draw power for computation, sensing functions, and for communication to readers. They may be compliant with RFID standards, or they can use protocols based on wireless protocols such as ZigBee or WLAN.

Evigia [8], Savi [11] and Crossbow [12] develop many active sensor tags that integrate many embedded sensors like temperature, humidity, light sensor, etc.

Several companies ([11] [12]) also provide active RFID tags embedded with sensors nodes (e.g., temperature, light, presence detection). In the best case, those devices are equipped with GSM/GPRS communication capabilities which make them totally independent from supply chain player's IT infrastructure.

B. *Software integration*

In [3], the authors propose a new architecture made of RFID tags and sensors. It consists in adapting the Application Level Event middleware (ALE) within the EPCglobal Network so that the filtering originally done over tag data is extended to sensor data. The main idea of this architecture is to distinguish data and control mechanisms in both RFID and WSN by introducing a reader management component that provides a uniform interface. This component is able to include WSN by adopting a Universal Plug and Play (UPnP) and the Simple Network Management Protocol (SNMP). As such, the WSN data are delivered to upper layers exactly like for RFID data, because the reader management component is able to handle data coming from RFID and WSN.

C. *Conclusion*

Regarding hardware integration within a single device, the exposed approaches often lack of integration with business applications. Those hardware solutions still require specific development for their integration into supply chain management specific.

Our solution belongs to this category, this type of integration offers advantages, as no new integrated nodes are needed. Moreover this solution provides a common infrastructure to link RFID system and WSN together.

So far, there are no standards proposing the integration of separate RFID and WSN into one network. To the best of our knowledge, our approach is the first combining sensors and RFID passive tags for specific needs for securing supply chains.

III. Our Approach

As introduced in Section I.D, our approach aims at integrating RFIDs and WSNs through the establishment of a communication channel between EPCIS, for RFIDs, and a mediation layer, the Middleware for Device Integration (MDI), for WSNs.

For that purpose, we use the MDI as an interface between RFIDs and WSNs. The main idea of our approach is to integrate RFID and sensors at the software layers. To do this, we make interacting two software elements: the EPCIS (designed for RFID as part of the RFID standards) and the MDI middleware (originally designed by industrials specifically for WSN). These elements support authentication of the tagged products and activation of the monitoring mechanisms within the sensors.

The integration at the software layer is necessary within this solution because we use very low-cost passive RFID tags with existing standards like EPC [10] to authenticate the products whereas we use WSN to monitor products along the supply chain.

A. *Architecture*

As depicted in Figure 2., our architecture is organized in three main components: (i) a palett equipped with an

RFID and a sensor node, (ii) an EPCIS component, and (iii) the MDI.

1) Description of EPCIS

The EPCIS module is a special web service interacting with the whole RFID system (reader, tags, and database) and serving as an RFID system interface by the rest of our platform. That is, it gives access to serialized product information generated by EPC-tagged products and available in the backend database. In other words, the EPCIS module is like a gateway between any requester of tag information and the database containing that information. EPCIS is able to handle large volumes of tag data coming from numerous RFID readers.

2) Middleware for Device Integration.

As mediation layer between EPCIS and sensor nodes, we use a mediation layer called the Middleware for Device Integration (MDI). MDI is a mediation layer developed by SAP Research [21] for the integration of smart items (e.g., WSNs, RFID) into business applications. Based on an OSGi Service Platform [22], MDI is an agent-based middleware which enables both monitoring and controlling of smart items. In Figure 2., the MDI hosts three agents: the sensor interface, the update, and the alerting service. The sensor interface enables communication with nodes, and the alerting service triggers alerts to users.

In addition the updater agent is in charge of the update of monitoring and controlling constraints on the sensor nodes. Depending on the classification and activity update from RFID, this agent updates on the nodes the constraints to be evaluated. For example, an alert whenever temperature reaches a given threshold can be activated. The same alert can be further on deactivated, in case of activity changes along the supply chain. For example, pallet fall alerting must be evaluated during storage in warehouse, but deactivated whenever the pallet is manipulated.

In Figure 3., we identify the following relationships: constraints, based on regulation, depend on asset's classification and activity along the supply chain execution. In addition, risk is depending on mismatch between constraints on goods and their context in the supply chain. If a constraint on temperature is defined, the probability on incident occurrence is depending on a mismatch with the context of the asset.

To that purpose, our updater agent enablement service is in charge of maintaining a monitoring database, based on the nature of the products, and the status of the supply chain process. Accordingly constraints to be evaluated are pushed to sensor nodes.

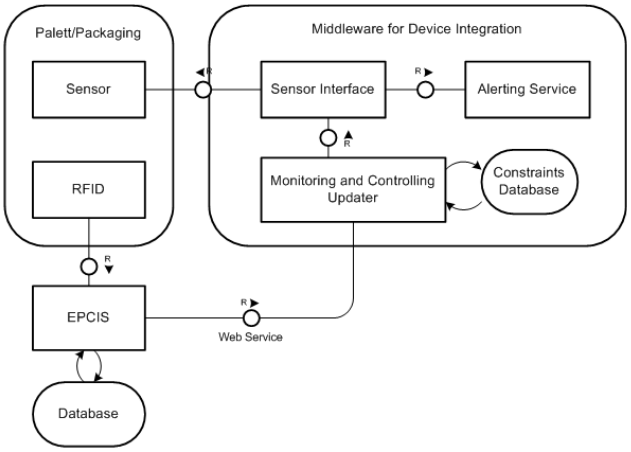


Figure 2. Architecture

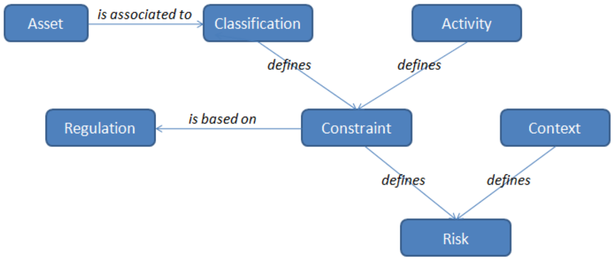


Figure 3. Risk Assessment

B. Interactions between RFID/Sensor Systems

The sensor node on the pallet is capable of in node processing, and is able to measure different contextual information (e.g., temperature, humidity). On the other hand, RFID is passive. As explained in Section III, the EPCIS service stores in its database all the information relative to the RFID tagged pallets: i.e. the SSCC [25] information (e.g. identifier of the pallet along the supply chain), CLP classification of goods (Classification, Labeling and Packaging of substances and mixtures) [24]). Also, the RFID identification is mapped manually to the identifier of the sensor attached to the pallet during the initialization phase of the EPCIS database. Table 1 gives the adopted EPCIS data model.

RFID Id	SSCC	CLP	Sensor Id

TABLE I. EPCIS DATA MODEL

Upon reading a tagged pallet, an RFID reader must send the RFID Id to the EPCIS service. EPCIS is then in charge of informing the MDI that the sensor identified by sensorId monitors a good classified according to the CLP norm. Based on that classification, the service enablement within the MDI extracts a set of constraint to be enforced on the sensor Id node. Those constraints are described in the constraint database, as shown in Table 1. Finally constraints are pushed to the sensor nodes via the MDI.

For example, for a chemical and anytime along the supply chain, a flash point of 13 Celsius degrees is defined by safety regulation. In addition, and only during storage, a constraint on acceleration of the pallet is activated, in order to prevent any pallet fall.

Classification	Status	Set of constraints

TABLE II. SERVICE ENABLEMENT DATA MODEL

Also upon detecting a status change along the supply chain, the RFID system registers that information to its own database and forwards the new status with the RFID Id to the MDI. Status change is then pushed to the sensor nodes.

C. *Implementation*

Our architecture has the advantage of leveraging standardized XML messaging frameworks for communications between EPCIS and MDI.

1) *RFID and EPCIS*

EPCglobal/ISO 18000-6C [10] is the RFID standard for supply chain. It defines a standard interface to enable EPC-related data; it does not specify how the service operations or database should be implemented. RFID tag communicates with the reader using the EPC Generation 2 protocol. The reader communicates with the EPCIS using the Reader Protocol. EPCIS has specific interfaces like EPCIS ALE (Application Level Event) interface, EPCIS Query interface.

EPCIS database stores all the product information like the traceability information, etc. In our solution, EPCIS works and exchanges information with MDI through a common interface thanks to web service.

2) *EPCIS and MDI*

When the EPCIS/Database authenticates the products, it informs the MDI about the nature of the pallet content, but also other tag data (e.g., sensor ID associated to the same pallet than the tag ID) and the product location (e.g., truck, warehouse, latitude, longitude). Then the MDI is enabled to activate adequate monitoring mechanisms on the sensors. This way, the logistician is able to monitor products along the supply chain. The MDI is able to handle large constraints coming from different EPCISs. Our approach consists in distinguishing the data and control mechanisms in both RFID and WSN in two different software layers. The proposed architecture has the advantage of leveraging standardized Web Service messaging frameworks [23] for communications between EPCIS and MDI.

The monitoring and controlling updater exposes a web service method for the update of a good, based on its classification and its status in the supply chain. The web service method signature is defined as follows

ErrorCode updateSensor(Integer SensorId, String GoodClassification, String Activity);

3) *Middleware for Device Integration and Sensors*

As depicted in Figure 2., three agents are running into the MDI. The updater agent is in charge of the update of constraints to be evaluated on nodes. An additional alerting agent is dedicated to the collection of alerts form the nodes, and the forwarding of alerts to end users. Finally a dedicated agent enables the communication between sensors and the two former agents. With respect to the targeted sensor node, a dedicated agent has to be implemented. In our case, we are focusing on crossbow [12] sensor node, which requires a dedicated agent. This agent has to be able to: (i) acquire information (e.g., alert, raw sensor data) from sensor nodes, and (ii) to push information (e.g., commands, constraints) to nodes.

IV. CONCLUSION AND FUTURE WORKS

As a conclusion, this paper proposed a framework for integrating RFID and sensor within the supply chain management systems. With this approach, we firstly aim at demonstrating the benefit of integrating RFID and WSNs. RFID are in charge of identifying good, their classification, and the status within the supply chain process. WNSs, embedded with monitoring and controlling capabilities, are meant to mismatch between regulation on goods, depending on their classification and on the status in the supply chain process. Therefore, WSNs depends on RFID information in order to run adequate alerting service on nodes.

We are planning an evaluation of our approach through a prototype in the scope of the RESCUEIT project. In addition, we are currently working of the design of security mechanisms for mutual authentication between RFID and tags reader, risk assessment delegations to nodes, and secure and efficient tracking of goods along the supply chain. Our goal is to address security requirement regarding the integration of RFID and WSNs for supply chain management systems.

V. DISCLAIMER

The research was partially funded by the German Federal Ministry of Education and Research under the promotional reference 01ISO7009 and by the French Ministry of Research within the RESCUE-IT project [14]. The authors take the responsibility for the content.

REFERENCES

[1] G. Eason, and B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] H. Liu, and M. Bolic, and A. Nayak, and I. Stojmenovic, “Integration of RFID and Wireless Sensor Networks,” SenseID 2007 Workshop at ACM SenSys, Sydney, Australia, 6-9 Nov. 2007.

[3] J. Sung, and T. Sanchez Lopez, and D. Kim “The EPC Sensor Network for RFID and WSN Integration Infrastructure,” PERCOMW '07, Washington, DC, USA.

[4] N. Cho, and S. Song, and S. Kim, and S. Kim, and H. Yoo “A 5.1-uW UHF RFID Tag Chip Integrated with Sensors for Wireless Environmental Monitoring” ESSCIRC, Grenoble, France, Sept. 2005.

ISBN: 978-1-61208-144-1

[5] M. Philipose, and J. R. Smith, and B. Jiang, and A. Mamishev, and A. Mamishev, and A. Mamishev, "Battery-free Wireless Identification and Sensing," IEEE Pervasive Comput., vol. 4, no. 1, Jan.–Mar. 2005, pp. 37–45.

[6] H. Kitayoshi and K. Sawaya, "Long Range Passive RFID-tag for Sensor Networks," Proc. 62nd IEEE Vehic. Tech. Conf., 2005.

[7] SensTag HF Passive RFID Sensing Products, http://www.phaseivengr.com/p4main/Technology/SensTAGWirelessRFIDSensors.aspx, April 2011

[8] Evigia, Container security Tag, http://www.evigia.com/products/hardware/tag/, April 2011

[9] Bill Glover (2006-02-03), "RFID Essentials", O'Reilly Media, Inc, USA.

[10] GS1/EPCglobal Network, http://www.gs1.org/epcglobal, April 2011

[11] Savi, Transportation Logistics and RFID, http://www.savi.com/products/rfid/rfid-tags.php, April 2011

[12] Crossbow, Logistics products, http://www.xbow.com/asset-tracking/products/index.html, April 2011

[13] K.C. Ko, and Z.H. Mir, and Y.B. Ko, "An Energy Conservation Method for Ubiquitous Sensor Networks Integrated with RFID Readers," submitted to WWIC 2007.

[14] L. Gomez, M. Khalfaoui, and E. El-Khoury, and C. Ulmer, and J-P Deutsch, and O. Chettouh, and O. Gaci, and H. Mathieu, and E. El Moustaine, and M. Laurent, H. Schneider, C. Daras, A. Schaad, « RESCUEIT : sécuRisation dE la Chaîne logiStique orientée serviCe depUis le mondE des objets jusqu'à l'univers InformaTique », Workshop Interdisciplinaire sur la Securite Globale, WISG 2011, Troyes

[15] Kuehne + Nagel, http://www.kn-portal.com/, April 2011

[16] Group Casino, http://www.groupe-casino.com/, April 2011

[17] REWE, http://www.rewe.de/, April 2011

[18] Dr Oetker, http://www.oetker.com, April 2011

[19] A. Dada and F. Thiesse "Sensor applications in the supply chain: the example of quality-based issuing of perishables", IOT'08: Proceedings of the 1st international conference on The internet of things. pp. 140-154.

[20] REACH, Registration, Evaluation, Authorisation and Restriction of Chemical substances, http://ec.europa.eu/environment/chemicals/reach/reach_intro.htm, April 2011

[21] SAP Research, http://www.sap.com/about/company/research/index.epx, April 2011

[22] OSGi Alliance, http://www.osgi.org, April 2011

[23] Web Service, http://www.w3.org/2002/ws/, January 2002

[24] Classification, labelling and packaging of substances and mixtures http://ec.europa.eu/enterprise/sectors/chemicals/classification/index_en.htm, April 2011

[25] GS1, "GS1 Identification Key Series", http://www.gs1.org/sites/default/files/docs/idkeys/GS1_SSCC_Executive_Summary.pdf, April 2011

Privacy Preserving Products Tracking in Clustered Supply Chain

Mehdi Khalfaoui, Kaoutar Elkhiyaoui, Refik Molva
Institut Eurecom
2229, Route des Cretes
06560 Valbonne, France
Email: mehdi.khalfaoui@eurecom.fr, kaoutar.elkhiyaoui@eurecom.fr, refik.molva@eurecom.fr

***Abstract*—One of the main applications of supply chain management is product tracking. We define it as tracing the product path along the supply chain. In this paper, we propose a solution to track the product while preserving the privacy of the supply chain actors involved and the path traced. More precisely, this solution allows to identify which path a product has taken in the supply chain, without disclosing sensitive information. To allow product tracking, the product are attached to a sensor node. This latter stores a trace of the product path along the supply chain. The trace is computed using polynomial based signature techniques. We restrict the visibility of the manager of the supply chain by organizing the supply chain facilities into clusters. Also, we encrypt the path traces to ensure security against adversaries. To perform access control in the sensor nodes we use randomized Rabin scheme which is known for being efficient and lightweight. In this paper, sensor nodes are not required to perform heavy computation, which makes our solution feasible. The main achievement of this work is a cryptographic mechanism that allows to the supply chain manager to trace the supply chain entities that product went through, without disclosing the identity of those entities.**

***Keywords*-Supply Chain; Privacy; Tracking; Cluster.**

I. Introduction

Recently, sensor-based applications have become more and more popular. One of their major applications is supply chain management [21]. More precisely, sensor nodes are used to monitor and track products from production, storage to distribution [6]. Furthermore, the heterogeneity of different parties involved in the supply chain, raises new security and privacy challenges. Partners aim at verifying the legitimacy of products in their sites, yet they are reluctant into leaking information about their internal processes.

Studies show that most security incidents involve business partners. This a significantly growing trend over the last few years [2]. Furthermore, the number of security incidents affects the trust between partners in supply chains [2]. Therefore, there are many attempts to overcome security issues and the lack of trust among the partners in the supply chain. In [14], the authors propose a mechanism based on Secure multi-party computation [26]. They aim at achieving security against the business partner. It enables the supply chain partners the computation of joint function without disclosing their inputs data. Only the final input is revealed. Secure multi-party computation can only compute specific families of functions, which restricts the used operations. In our case, Secure multi-party computation cannot compute our path trace. Elkhiyaoui et al. [4] propose RFID-based tracking mechanisms of the products in the supply chain. They allowed to authorized entities to validate the path of the products, without disclosing the identity of the others partners in the supply chain. The issue with this approach is the use of unsuitable arithmetic operations for low capacity devices.

In this context, the paper at hand aims at enabling the manager of the supply chain to verify the validity of the path a product took. Such a verification could allow the manager to detect counterfeits in the supply chain. Here, we assume that each supply chain has its own global manager. However, supply chains are distributed over sites or facilities. The latter reside in different locations and belong to different partners. Hence, the supply chain manager does not have full control over interconnections among the facilities. He also does not have full control over some of the facilities themselves. We propose, as possibility, that the products carry necessary information that will allow to a manager to verify their paths. To this effect, we use the memory capacity of the sensor nodes that are attached to the products to store the path trace.

In this paper, we propose a mechanism to protect the partners' privacy and the product privacy in the supply chain. First, to protect partners privacy against the manager, this latter should have a restricted visibility of the partners internal processes. Thus, we organize the supply chain into clusters such where sites and facilities belonging to the same partner will form a single entity. Then, when product arrives to the manager, he can identify which partners handled it. However, he cannot know precisely which sites and facilities the product visited. Figure 1 illustrates a clustered supply chain. This supply chain consists of : production, storage, and distribution cluster. Each cluster consists of three sites.

Second, to protect product privacy in the supply chain, only the manager should be able to verify the path of products in the supply chain.

The research was partially funded by the German Federal Ministry of Education and Research under the promotional reference 01ISO7009 and by the French Ministry of Research within the RESCUE-IT project. The authors take the responsibility for the content

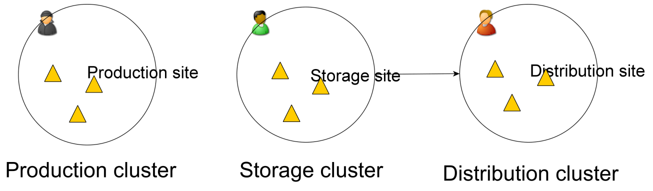


Figure 1. clustered supply chain

Any technical solution that addresses secure and privacy preserving product tracking should take into account the limitations of sensor nodes. These are constrained devices in terms of computation, power and memory. For example, RSA signature [8] cannot be implemented in sensor nodes.

This paper introduces a mechanism to track products in supply chain while protecting sensitive information of supply chain partners and products. The main idea is to organize the supply chain into clusters to restrict the manager visibility, and to attach products to sensor nodes which store an encoding of the products' path. The path encoding is computed using polynomial based signatures for run time fault detection [17]. In order to ensure the validity of the actors that interact with the product, sensors use Rabin scheme to authenticate them. However, Rabin scheme can be easily replaced by other mechanisms such as the one proposed by Gomez et al. [24]. Then, The path trace is encrypted to ensure confidentiality.

The major contributions of this work:

- It allows the supply chain manager to verify the validity of the paths a product took. More precisely, it allows the manager to verify which sequence of clusters, a product have visited.
- It guarantees the privacy of products and therewith partners in the supply chain. Only the manager is able to verify the path the product took.
- It only requires sensors to perform efficient low capacity operations and to store few Kbytes.
- It allows the restriction of product information that can be disclosed to each partner.

II. Solution

A. Preliminaries

A supply chain in this paper simply denotes a set of sites that a product goes through. As discussed previously, we organize the supply chain into clusters such that sites belonging to the same partner belong to the same cluster. These clusters are used to restrict the visibility of the manager of the supply chain. The manager is able to recognize the cluster a product visited but not the exact site the product visited. Thus, in this manner we enforce the privacy of the partners in the supply chain against the manager of the supply chain. The solution proposed involves the following entities:

- **Sensors** S_i: each sensor is attached to a product in the supply chain. A sensor S_i is equipped with a re-writable memory that stores the trace $s_{(S_i,j)}$. $s_{(S_i,j)}$ represents the trace of the path that the sensor took in the supply chain until cluster j. Sensors can also compute a cryptographic function f to authenticate the partners' systems in the supply chain.
- **Manager** M: the manager M attaches S_i to a product and writes into S_i an initial state $s_{(S_i,0)}$. M wants to identify the path a sensor S_i went through. More precisely, M wants to identify the sequence of clusters that S_i went through. M therefore, reads the current state $s_{(S_i,j)}$ of S_i, and decides whether S_i visited legitimate sequence of clusters or not. We assume that M knows which paths in the supply chain are valid or not. In other words, M has a database DB of valid path traces.
- **Clusters** c_k: To enforce the privacy of the partners in the supply chain, the supply chain will be organized into clusters. Each cluster contains a set of supply chain sites that belong to the same supply chain partner. Without loss of generality, we assume that each cluster c_k is equipped with a supply chain actor's system A_k. A_k uses some function f_{A_k} to generate $s_{(S_i,j+1)}$ from $s_{(S_i,j)}$, i.e., $f_{A_k}(s_{(S_i,j)}) = s_{(S_i,j+1)}$. The actor's system A_k can also compute a cryptographic function g to authenticate themselves to sensors.

III. Protocol

A. Protocol overview

In this paper, a sensor S state noted $s_{(S,j)}$ represents the sequence of clusters in the supply chain that S visited. One of the challenges in this work is to encode the sensor's path efficiently, i.e. encoding ha sto be independent of the number of visited clusters by the sensor. For that purpose, we use a technique for run time fault detection to encode paths using polynomials. More precisely, each valid path in the supply chain P_{valid} will match the evaluation of a unique polynomial $Q_{P_{\text{valid}}}(x_0)$ at a fixed number x_0. The efficiency of this encoding relies on two properties: **1)** a path is represented as polynomial evaluation at point x_0, therefore, the size of the encoding does not depend on the number of clusters a sensor visited **2)** for any two different paths in P_1 and P_2, the equation $Q_{P_1}(x_0) = Q_{P_2}(x_0)$ holds only with negligible probability [17]. As a result, the state of a sensor node S at the end of the supply chain can be uniquely mapped to one single path.

However, the path representation as presented above does not suffice to prevent path cloning, i.e., copying the path of a valid sensor into a fake sensor and then injecting the fake sensor in the supply chain. To tackle this problem, sensors store $Q_{P_{\text{valid}}}(x_0)$ added to a keyed HMAC of their unique IDs. HMAC is used for two purposes: first, it ensures that sensors are issued by a legitimate authority and prevents an adversary from injecting its own sensor. Second, it allows to map the Sensor's ID to a random number that cannot be predicted by the adversary. Therefore, an adversary cannot clone a sensor more than once, and thus, cloning cannot be performed in a large scale.

Whenever, a sensor S visits a cluster in the supply chain, the actor's system updates the sensor node's state by updating the polynomial evaluation. In a nutshell, the protocol consists of

- Initialization phase: Supply Chain manager M initializes sensors, and distributes IDs to the different partners.
- Authentication phase: Sensor S authenticates each visited site, before updating its trace.
- Collection phase: Sensors successively store the evaluation of a polynomial. That is achieved by updating the trace of the sensor in each cluster.
- Verification phase: Manager M extracts the sensor's path trace and therewith the polynomial. M checks whether the sensor state corresponds to a valid sequence of clusters.

Privacy and security overview: : To protect product privacy, sensors will store only probabilistic Paillier encryptions [19] of their states, and the actor's systems use homomorphic techniques for arithmetic operations on encrypted path encodings. At the end of the supply chain, M can then decrypt and identify the path. Also, the use of Paillier cryptosystem and HMAC guarantees the security of path encoding.

B. Path Encoding technique

The polynomial path encoding is used in [4]. It is based on techniques for software fault detection. Noubir et al. [17] propose to encode a softwares state machine using polynomials such that the exact sequence of states visited during run-time generates a unique *"mark"*. Therewith, run-time faults can be detected. By considering the actor's system instead of state machine, the path encoding used by Noubir et al. [17] can be applied in our case.

For each cluster c_i in the supply chain, c_i is associated with a unique random identifier $c_i \in \mathbb{F}_q$, where q is a large prime.

As mentioned above, a path in the supply chain is represented as a polynomial $\in \mathbb{F}_q$. The polynomial corresponding to a path $\mathcal{P} = \overrightarrow{c_0 c_1 \dots c_l}$ is defined in Equation (1). All operations are in $\mathbb{F}_q$.

$$Q_{\mathcal{P}}(x) = \sum_{i=0}^{l} c_i x^{l-i} \tag{1}$$

To have a more compact representation of paths, a path $\mathcal{P}$ is represented as the evaluation of $Q_{\mathcal{P}}$ at x_0, where x_0 is a generator of $\mathbb{F}_q^*$. We denote $\phi(\mathcal{P}) = Q_{\mathcal{P}}(x_0)$. The desired property of anti-collision, .i.e $\forall \mathcal{P} \neq \mathcal{P}\prime, Pr(\phi(\mathcal{P}) = \phi(\mathcal{P}\prime)) = \frac{1}{q}$ [17], ensures the uniqueness of the path mark with high probability.

C. Paillier Cryptosystem

The following is description to the Paillier cryptosystem [19] that we use in order to achieve both privacy and security of our mechanism:

Key Generation: Let k be the security parameter. Choose uniformly and at random two k-bit primes p and q, set $N = pq$, and set $\lambda(N) = lcm(p-1, q-1)$. Choose a random base $g \in \mathbb{Z}_N^*$.

Encryption: To encrypt message $m \in \mathbb{Z}_N$, one chooses a random value $r \in \mathcal{Z}_N^*$ and computes the ciphertext as

$$c = \mathcal{E}(m,r) = g^m r^N mod N^2 \tag{2}$$

Decryption: When receiving a ciphertext c, check that $c < N^2$. If yes, retrieve the message m as

$$m = \mathcal{D}(c) = \frac{L(c^{\lambda(N)} mod N^2)}{L(g^{\lambda(N)} mod N^2)} mod N \tag{3}$$

Where $\forall u \in \{u < N^2/u \equiv 1 mod N\} L(u) = \frac{u-1}{N}$

Additive Homomorphic property: Paillier cryptosystem has the property to be additively homomorphic:

$$\mathcal{E}(m_1, r_1) * \mathcal{E}(m_2, r_2) = \mathcal{E}(m_1 + m_2, r_1 r_2) \tag{4}$$

This property allows the execution of arithmetic operations on encrypted data. Therefore, it supports the evaluation the polynomial mark at each cluster of the supply chain without decryption.

Self Blinding: Paillier cryptosystem has the property to be *Self-Blinding*, i.e the property by which any ciphertext can randomly be changed into another without affecting the plaintext. This property is achieved as follows:

$$\forall r \in \mathbb{Z}_N \mathcal{D}(\mathcal{E}(m,r)) = m \tag{5}$$

Therefore, the decryption of any message m is independent of the value of r.

D. Detailed Protocol description

Our protocol consists of an initialization phase, the phase that prepares a sensor to enter the supply chain. Then, the authentication phase to verify the legitimacy of the site before collecting its trace. Collection phase, where the sensor interacts with different Sites and collects its traces. Finally, the path verification phase, when the polynomial mark is extracted by the supply chain manager M and the path gets verified. Figure 2 illustrates the sequence of the different phases, which compose the sensor life-cycle.

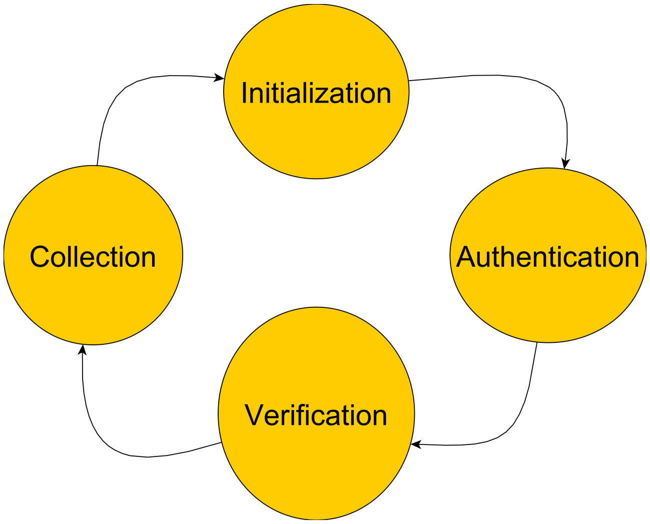


Figure 2. sensor lifecycle

1) Initialization phase: In this phase, we assume that every partner in the supply chain has the necessary resources to perform the following actions:

- supply chain manager M shares with the supply chain actors a Rabin's private key (p_R, q_R). Rabin cryptosystem [22] is used to achieve authentication of the supply chain actors. Rabin encryption is single square modular encryption, which makes it feasible for low capacity devices such as sensor nodes. Rabin's public key $N_R = p_R q_R$ is stored in the sensor to perform the authentication process.
- M generates randomly a list of cluster identifiersc_{list}. For the sake of simplicity, we don't distinguish between cluster c_i and its identifier.
- M generates a Paillier public key pk_M and private key sk_M. Then M sends $\mathcal{E}_M(c_i)$ to each actor belongs to the cluster c_i in a secure way.
- M generates the identifiers list S_{list} of sensors. For the sake of simplicity, we assimilate a sensor S_i and its identifier ID_i. M stores in each sensor S_i, the Paillier encryption of its ID_i, the Paillier encryption of $HMAC_k(S_i)$, where $HMAC_k$ is keyed hash function [3], and k is its secret key.

Now, for each sensor entering the supply chain, M has already stored on it the initial value $s_{(S_i 0)} = E(HMAC_k(S_i))$. The table I illustrates the exchanges messages in the initialization phase between M and S in , and between M and each site in the supply chain.

2) Authentication phase: In this phase, the sensor S and the actor's system A interact in the supply chain by executing the following actions: We assume that the sensor S has visited the clusters c_0, …, c_l. When, S visits the actor's system A_{l+1}, it is already stored the trace $\mathcal{P}_l = \overrightarrow{c_1 c_2 ... c_l}$ that encodes the path from the sites that belongs to the clusters $c_1, c_2, \ldots, c_l$. Therefore, the current state of the sensor is $s_{(S,l)}$, which corresponds to the state of the sensor after interacting with l clusters .

$M \to S,\ g^{HMAC_k(S)}$
$\mathcal{M} \to Site \in c_i,\ \mathcal{E}_M(c_i)$

Table I
INITIALIZATION PHASE

S chooses a random value $r \in \mathbb{F}_{N_R}$ and sends $Rabin(r) = r^2 mod N_R$ to A, while storing the $hash(r)$ and $hash(N_R - r)$. Abduvaliev et al. [1] show that a large possibility of hash functions can be implemented in low capacity devices such as sensor nodes. A decrypts $Rabin(r)$ using its public key. The decryption gives exactly four solutions, r, $N_R - r$, t, $N_R - t$. As the actor does not know which is the real solution, he chooses to send back to S two hash values. The values are chosen in such a way that their sum is not null $mod N_R$. For example A chooses to send $hash(r)$, and $hash(N_R - t)$. Therefore, S considers the authentication as successful, if one of the received value matched one of the stored value. Then, S can start trace collection procedure. Table II illustrates the messages exchanged between S and site to verify if the actor is legitimate or not.

S picks randomly a number r
$S \to Site,\ r^2 mod N_R$
$Site \to S,\ hash(r), hash(t)$

Table II
AUTHENTICATION PHASE

3) Collection phase: After the authentication phase, S starts the collection phase. S sends its current state $s(S, l)$ to the actor's system A_{l+1}. A_{l+1} updates the sensor's state as following:

$$s(S, l{+}1) = s(S, l)^{x_0} {*} \mathcal{E}_M(c_{l+1}) = s(S, l)^{x_0} {*} g^{c_{l+1}} r^N mod N^2 \tag{6}$$

Assuming that our products has to interact with n supply chain partner, the final sensor's state is:

$$\begin{aligned} s(S, n) &= s(S, n-1)^{x_0} * g^{c_n} r_1^N mod N^2, where \quad r_1 \in \mathbb{F}_N \\ &= s(S, 0) * g^{\sum_{i=1}^{n} c_i {x_0}^{n-i}} r_2^N mod N^2, where \quad r_2 \in \mathbb{F}_N \\ &= g^{HMAC_k(S) x_0^n + \sum_{i=1}^{n} c_i {x_0}^{n-i}} r_3^N mod N^2, \quad r_3 \in \mathbb{F}_N \\ &= \mathcal{E}_M(HMAC_k(S) {x_0}^n + \sum_{i=1}^{n} c_i {x_0}^{n-i}) \end{aligned}$$

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-144-1

Table III illustrates the two messages exchanges between S and random site in the collection phase.

$S \rightarrow$ Site, $s(S, i)$
Site $\rightarrow S$, $s(S, i+1)$

Table III
COLLECTION PHASE

4) Verification phase: : In this phase, the supply chain manager M checks if the path recorded in the sensor is a valid one. M extracts the final state from the sensor $s(S, n)$, and decrypts it, so he can extract the path trace $\phi(\mathcal{P})$.

$$\begin{aligned} \phi(\mathcal{P}) &= \mathcal{D}_{TTP}(s(S,n)) = \sum_{i=1}^{n} a_i x^i \\ &= HMAC(S)x_0^n + \sum_{i=1}^{n} c_i {x_0}^{n-i} \end{aligned} \tag{8}$$

Using successive division operations, M extracts the coefficients a_0,a_1,. . .,a_n of the polynomial $\phi(\mathcal{P})$. Then, M computes $HMAC_k(S)$ and compare it with a_n. If $a_n = HMAC_k(S)$, M accepts the sensor. Otherwise, M rejects the sensor. Finally, M checks if the cluster identifiers c_1, c_2, . . ., c_n belongs to the cluster identifiers list, and the path trace $\phi(\mathcal{P})$ proofs that the sequence of the clusters is valid. if one of the identifiers, or the sequence is not valid, M rejects the sensor, and declares the products as non compliant.

IV. EVALUATION

Our protocol is implementable using today's sensors such as Crossbow motes [13] and phidgets [12]. It only requires sensors to store the Rabin public key, which is 1024 bits, and the encrypted state, which is 2048 bits, so a total memory of 3Kb. Through the different steps of the supply chain, the amount of memory needed does not increase. Storing 3Kb of data is feasible in today's sensor hardware. Hempstead et al. [10] show that the memory available in hardware sensors are between 8 kB (i.e. 68Kb) and 132 kB (i.e. 1056 Kb). Therefore, from a memory capacity perspective, our protocol is efficient.

The complexity on the nodes is low. The sensor has to perform at each step of the supply chain, the same arithmetic operations. Therefore, the complexity is linear to the number of the actors in the supply chain. The sensor needs to authenticate the visited site in the supply chain,which means it has to perform one modular square and to compute two hash functions. These operations are necessary to perform a Rabin based authentication. Gaubatz et al. [7] showsthat one (7)modular multiplication in sensor, needs roughly $1\mu J$, which is very low compared to RSA signature requirements [25]. A sensor node of type crossbow has 2 AA batteries, which means roughly $4KJ$ of energy [11].

V. RELATED WORK

The idea of using WSN in the supply chain management to track goods was first suggested in [16]. However, research focused mainly on RFID tags to achieve secure tracking in supply chain. Ouafi and Vaudenay [18] address counterfeiting of products using strong cryptography on RFID tags. Blass et al. [4] presente a tracker, a new mechanism to protect against malicious state update of tags in each step of the supply chain. Secure tracking of specific target using WSN was also addressed in [9]. It describes a mechanism of tracking a moving target based on relaxation algorithms [20]. However, passive RFID tags have limited ressources, which makes security and privacy hard to achieve. As matter of fact, Modular multiplication which is necessity to perform arithmitic operations, cannot be impleted in this type of RFID tags. Only hash functions is implementable in passive RFID tag environment. Chawla et al. [5] check whether covert channels exist in a supply chain that leak information about a supply chains internal details to an adversary using security mechanism implemented in RFID tag. Therefore, tags state is frequently synchronized with a backend-database. If a tags state contains data that is not in the database, the tag is rejected. As supply chain involve different actors, it is difficult to have a single backend-database common among them. Our mechanism focus, however, is on secure and privacy-preserving identification of the path a sensor has taken. Shuihua and Chu [23] detect malicious tampering of a tags state in a supply chain using watermarks. However, there is neither a way to identify a tags path, nor to protect its privacy in the supply chain. Kerschbaum and Oertel [15] detect counterfeits in the supply chain using pattern matching for anomaly detection. This latter can be combined with our mechanism to achieve cloning countermeasure.

VI. CONCLUSION AND OUTLINE

In this paper, we presented a protocol to secure the tracking of products in supply chain. Our main idea is to encode the path of the products using polynomial path encoding. Partners in the supply chain update the path trace successively, such that the path has unique identifier. Our protocol's security and privacy proprieties relies on the semantic security of Paillier and the security of HMAC. It requires only one modular multiplication in each step, and only 3Kb of storage, which ensures its feasibility in available sensors in the market.

In our supply chain scenario, we assume that we have a global supply chain manager. There is no notion of multiple managers. However in real world, That is might not be true.

Supply chain can have a quality, security, and recall manger. Delivering the right information to the right manager is an issue, especially in big scale supply chains. However, this is left to future work

References

[1] A. Abduvaliev, S. Lee, and Y. Lee. Simple hash based message authentication scheme for wireless sensor networks. In *Proceedings of the 9th international conference on Communications and information technologies*, ISCIT'09, pages 982–986, Piscataway, NJ, USA, 2009. IEEE Press.

[2] W.H. Baker, C.D. Hylender, and J.A. Valentine. Data breach investigations report. *Verizon Business RISK Team*, 2008.

[3] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In *Advances in CryptologyCRYPTO96*, pages 1–15. Springer, 1996.

[4] E. Blass, K. Elkhiyaoui, and R. Molva. Tracker : security and privacy for rfid-based supply chains. In *NDSS'11, 18th Annual Network and Distributed System Security Symposium, 6-9 February 2011, San Diego, California, USA, ISBN 1-891562-32-0*, 02 2011.

[5] K. Chawla, G. Robins, and W. Weimer. On Mitigating Covert Channels in RFID-Enabled Supply Chains. *RFIDSec Asia, Singapore*, 2010.

[6] A.J. Clark and H. Scarf. Optimal policies for a multi-echelon inventory problem. *Management science*, pages 475–490, 1960.

[7] G. Gaubatz, J.P. Kaps, and B. Sunar. Public key cryptography in sensor networksrevisited. *Security in Ad-hoc and Sensor Networks*, pages 2–18, 2005.

[8] R. Gennaro, H. Krawczyk, and T. Rabin. RSA-based undeniable signatures. *Advances in CryptologyCRYPTO'97*, pages 132–149, 1997.

[9] R. Gupta and S.R. Das. Tracking moving targets in a smart sensor network. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, pages 3035–3039. IEEE, 2003.

[10] M. Hempstead, M.J. Lyons, D. Brooks, and G.Y. Wei. Survey of hardware systems for wireless sensor networks. *Journal of Low Power Electronics*, pages 11–20, 2008.

[11] http://www.allaboutbatteries.com/Energy tables.html. Last access: 05/06/2011.

[12] http://www.phidgets.com/. Last access: 12/06/2011.

[13] http://www.xbow.com/. Last access: 01/06/2011.

[14] F. Kerschbaum and R.J. Deitos. Security against the business partner. In *Proceedings of the 2008 ACM workshop on Secure web services*, pages 1–10. ACM, 2008.

[15] F. Kerschbaum and N. Oertel. Privacy-preserving pattern matching for anomaly detection in RFID anti-counterfeiting. *Radio Frequency Identification: Security and Privacy Issues*, pages 124–137, 2010.

[16] W. Liu, Y. Zhang, W. Lou, and Y. Fang. Managing wireless sensor networks with supply chain strategy. 2004.

[17] G. Noubir, K. Vijayananda, and H.J. Nussbaumer. Signature-based method for run-time fault detection in communication protocols . *Computer Communications*, pages 405–421, 1998.

[18] K. Ouafi and S. Vaudenay. Pathchecker: An RFID Application for Tracing Products in Supply-Chains. In *International Conference on RFID Security*. Citeseer, 2009.

[19] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in CryptologyEUROCRYPT99*, pages 223–238. Springer, 1999.

[20] K.R. Pattipati, S. Deb, Y. Bar-Shalom, and R.B. Washburn Jr. A new relaxation algorithm and passive sensor data association. *Automatic Control, IEEE Transactions on*, pages 198–213, 1992.

[21] EU Project. Stop tampering of products. 2010.

[22] M.O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. 1979.

[23] H. ShuiHua and C.H. Chu. Tamper Detection in RFID-Enabled Supply Chains Using Fragile Watermarking. In *RFID, 2008 IEEE International Conference on*, pages 111–117. IEEE.

[24] A. Sorniotti, R. Molva, and L. Gomez. Efficient access control for wireless sensor data. *Ad Hoc & Sensor Wireless Networks*, pages 325–336, 2009.

[25] L. Uhsadel, A. Poschmann, and C. Paar. Enabling full-size public-key algorithms on 8-bit sensor nodes. In *Proceedings of the 4th European conference on Security and privacy in ad-hoc and sensor networks*, pages 73–86. Springer-Verlag, 2007.

[26] A.C. Yao. Protocols for secure computations. In *proceedings of the 23rd Annual IEEE symposium on Foundations of Computer Science*, pages 160–164. Citeseer, 1982.

Adaptive Security on Service-based SCM Control System

Gabriel Serme
Eurecom
serme@eurecom.fr

Muhammad Sabir Idrees
Eurecom
idrees@eurecom.fr

***Abstract*—On a large-scale application subject to dynamic interactions, the description and enforcement of security rules are complex tasks to handle, as they involve heterogeneous entities that do not have the same capabilities. In the context of SCM-application for example, we have different goods that are being transported across different systems. At one point, items and systems communicate together to signal presence, report issues during transport, certify validity of previous checks, etc. Security capabilities of the involved parties are heterogeneous and one might want to specify security policies on an abstract level and let the involved systems enforce them according to their contexts and the specific capabilities of each party. In this paper, we propose a framework for security mechanisms adaptation when services are involved by using Aspect-Oriented-Programming (AOP) concepts that can be applied to SCM applications. The novelty is the expressivity of security policy at a global level and the enforcement at a local level, through a specific and distributed aspect model that has a larger semantic to catch up events relevant for business usage and dedicated to security concerns.**

***Keywords*-SOA, Security, AOP.**

I. Introduction

An SCM application can be viewed as a long chain process along which goods have to pass through mandatory gates. It involves various devices, from embedded systems like sensors to large-scale servers in backend systems. Sensors usages are dedicated to data collection and signal triggering. They try to capture real-world status and measure it. Backend systems allow for data processing but need to adapt to all devices communicating with them, as each can have a different communication protocol and data format.

The heterogeneity of platforms and software used in devices makes it difficult to manage simple security rules, especially across a supply chain. In order to deal with the multiple possibilities and not to interfere with the business part of software, one might want to describe security behavior for one system that adapts to security capabilities of systems communicating with it. To do so, we propose an architecture that allows correct modularization of security concerns to quickly intervene in applications and make them adapt to the conditions they can face up to.

The application uses the SOA architectural style to provide a loosely-coupled platform where entities can integrate with each other. In the following sections, we start by explaining the different concepts we are using in our proposed architecture. Namely, Web Services and SOA concepts, security properties we aim to express in an adaptive manner and also AOP (Aspect-oriented programming) paradigm. Then, we describe the proposed architecture and process to handle service adaptation with two examples highlighting difficulties to adapt security for systems accordingly.

II. Services

Service Oriented Architectures (SOA) enable a world of loosely-coupled and interoperable software components towards reusability. Nowadays, the main entity used to represent a software service is a Web Service. Web-Services represent a paradigm defined by W3C as ”a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, conveyed using HTTP with an XML serialization in conjunction with other Web-related standards” [1]. Web Services can also be addressed through other transport mechanisms such as JMS or ESBs.

The Web Service standards stack goes beyond the atomic service, and proposes different approaches depending on the level of abstraction. Service behavior can be defined when linking different services together, *e.g.,* with BPEL4WS or BPMN 2.0 [2]. It allows definition of service composition to realize a so-called business process.

III. Security

As services are advancing fast and are being extensively deployed in applications spanning different organizations, it becomes crucial to ensure security and trust for these applications to hold their promise. It was only recognized in recent years that services are themselves susceptible to various attacks at different levels of system conceptualization [3].

Since SOA has a flexible set of design principles used during the phases of system development, integration, and evolution, one obvious and common challenge is to secure SOA. This often involves invasive modifications, in particular to enable new security functionalities that require modifications to applications. Furthermore, enforcing crosscutting security functionality in service-based systems is difficult to specify and implement because the security of services and their compositions is not modular. Modifications made to one part of an application may interact strongly with the security properties of other parts of the same application. Security properties generally pervade software systems, that is, security properties crosscut service-oriented architectures. Enforcing service level security needs specialization based on the implementation.

We propose in the following an aspect-based service model that proposes an original method to introduce several security properties. These security properties are specified in a security policy language that is then interpreted to generate crosscutting concerns. It includes *Integrity* which relates to communication, storage (resources), and execution (process - infrastructure) integrity. The execution environment integrity is an important security objective together with data integrity measures. *Confidentiality* relates to message exchange between entities such as sensors and services, or need-to-know principles limitation applied to specific resources. *Authentication and Authorization* crosscut applications to decide at several points if a given subject is allowed to perform an action on a given resource. Whereas authorization decisions are mainly on server-side, authentication mechanism needs to adapt all peers to agree on the scheme, including sensors authentication. Applying *non-repudiation* requires the implementation of an asymmetric encryption scheme in the execution environment supporting the computation.

The aforementioned properties represent security goals we want to apply on applications by adapting them with our framework.

IV. Aspect-Oriented Programming

The term Aspect-Oriented-Programming [4] (AOP) has been coined around 1995 by a group led by Gregor Kiczales, with the goal to bring proper separation of concerns for cross cutting functionalities. Roots for foundations can be traced back to adaptive programming, or composition filters [5]. O. Selfridge introduced a notion that can be related to AOP as ”demons that record events as they occur, recognize patterns in those events, and can trigger subsequent events according to patterns that they care about” [6]. But the approach has then derived to become a discipline apart.

The aspect concept is composed of several advice/pointcut couple. Pointcuts allow to define where (points in the source code of an application) or when (events during the execution of an application) aspects should apply modifications. Pointcuts are expressed in pointcut languages and often contain a large number of aspect-specific constructs that match specific structures of the language in which base applications are expressed, such a pattern language based on language syntax. Advices are used to define modifications an aspect may perform on the base application. Advices are often expressed in terms of some general-purpose language with a small number of aspect-specific extensions, such as the *proceed* construct that allows the execution of the behavior of the base application that triggered the aspect application in the first place. The main advantage using this technology is the ability to intervene in the execution without interfering with the base program code, thus facilitating maintainability.

V. Architecture proposal

In this paper, we shape the solution we are currently implementing at the service layer. The root of the problem is to instrument several services at the same time, potentially not under the same execution environment to realize a specific security property. Illustration examples are available in next section. The architecture is presented in Figure 1. It contains two parts, involving design and runtime part. The design part involves business stakeholders to define aforementioned security policies (also denoted rules), and security experts to provide concrete security mechanisms as pre-defined aspects. The runtime part of the architecture leverages the aspect model to modify the different execution environments and make them satisfy security policies specified by business and security stakeholders. The main piece is the runtime engine whose goal is to detect a certain *state* across platforms. The state is described by *rules* composed of *predicates*. Upon matching between rules and a state, platforms coordinate to realize a new behavior. Locally, systems implement *mechanisms* to realize a behavior specified in the knowledge base and make usage of *context information* available at execution.

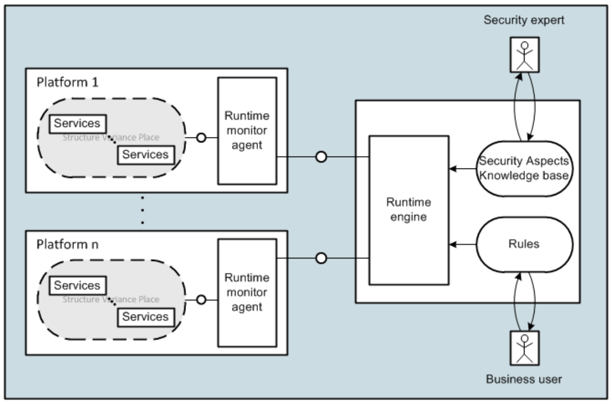


Figure 1. Proposed architecture

Runtime monitor agents and runtime engine work together to realize a distributed aspect model, introduced in [7]. The advantage of such an architecture is to intervene in one specific organization where we separate security concerns across different platforms. The security code is no longer tied to the business code. Rather, it is decoupled from business code intentionally and bound with the distributed pointcut language. The runtime engine uses rules to gather state of various services at the same time, and security aspects are used in advice to dispatch security behavior across several services. For example, in the integrity and origination scenario, the distributed aspect model tracks all places where messages come in the system to taint them. Then, the system tracks these taint messages to weave security aspects when behavior is needed.

Security policies are specified by business users, security experts and architects then derived in rules to realize a specific security property. To provide an enriched semantic addressing all concerns of services and security at the same time, we are developing a policy which contains high level description of wanted behavior, through predicates. The predicates allow matching with a particular state. The language semantic relates to events and actions a platform can generate, *i.e.,* services, messages and resources. *Services* can be atomic or composite, *e.g.,* orchestration of services with BPEL4WS or BPMN2.0, or simply a service that consumes other services. Examples of predicates that can be used are ”receive” or ”reply” to match a service call. There are also predicates for *messages* and *resources*. Predicates have different level of abstraction relating to the service stack we discussed above.

The policies also contain behavior that stakeholders want to introduce across systems. In our framework, we address only security concerns such as the ones described in previous sections: confidentiality, integrity, authorization, etc. In Listing 1, the policy describes integrity and non-repudiation presence in messages when they are issued by sensors. The message origination is verified when we are able to verify signature. The behavior is described in an abstract way to indicate which parties are concerned and what shall be executed. Platforms receive this behavior and are in charge of translating it according to mechanisms available for the given platform. The Listing 2 in next section is an exemple of java code to verify a given signature for a message. The runtime monitor detects a certain application state through the aforementioned predicates. Upon matching, the wanted behavior is read from policies and spread to concerned systems to satisfy and realize security properties.

```
message_in (issuer, msg):
  issuer in (s:sensors) => verify_integrity (s,
       msg), verify_origination (s, msg)
  msg.taint(UNSAFE) #Default
  msg:integrity, msg:non_repudiation => msg.taint(
       SAFE)

verify_integrity (msg):
  msg.contains(integrity), integrity.check(msg) =>
        msg:integrity

verify_origination (msg, issuer):
  msg.contains(sign), sign.issued(issuer) =>
       verify_origination (msg, sign,issuer), msg:
       non_repudiation
```

Listing 1. Policy snippet for integrity and non-repudiation check

Our framework heavily relies on aspect oriented paradigm. The runtime monitor is able to detect system state using a distributed aspect pointcut language. On matching system state, advices are executed with the system’s context through context exposition mechanisms. For example, a service can expose information about inputs, outputs, service origination, as well as other security-related information.

The proposed architecture allows definition of security policies for service systems at a global level that are then enforced at a local level in an semi-automatic mode. We propose to decouple definition of specific security properties from the base application, and let declaration through rules respecting application owners’ needs.

VI. Application Example

We describe two scenarios illustrating when our framework can be applied in a SCM application. The long term scenario (cf Fig. 2) is a military container that is sent to supply a camp thanks to a boat transport. Shipment is not direct and the container has to pass through several intermediaries, to refuel, change boat, etc.. Thus, it has been decided by the army to frequently track and check containers when they stop in harbors. The communication between containers and the army system is made through Web Services and use the Harbor system to certify the shipment advancement.

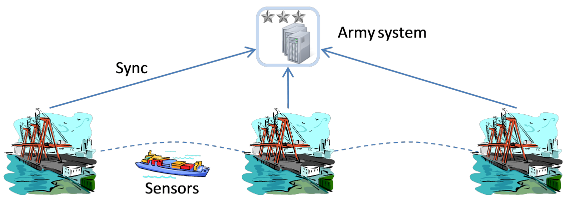


Figure 2. Army shipment and control system

The first scenario leveraging our framework is on adaptation and tracking of sensitive data. It highlights integrity and non-repudiation scenario and how it impacts the existing architecture on the army systems. Over time, the army deployed containers with protection mechanisms to detect failure or intrusion composed of sensors and nodes. Different software solutions are shipped with containers with different security capabilities.

Maintaining applications, both in nodes and back-end systems, is costly. It requires business owners to specify each possible use case, at a time t. Upon release of new version, they need to extend existing specifications and activate their development team. The development team has to correctly implement the solution and to not break the previous solutions. The release cycle can be counted in weeks or even months and is error prone. A suitable solution is to have a framework that knows what to do given a certain situation. For example we want to allow communication between all nodes versions and the back end system while keeping track of sensitive nodes - those which do not implement security mechanisms. An example of policy we might define to detect various versions of protocols is described in Listing 1. The policy language used is not yet fully developed and is used as an illustration.

```
 1 @Aspect
 2 Class Verification{
 3   //...
 4   @Covers(SPL.verify_origination) //Provided by
         framework
 5   boolean verifyOrigination (Byte[] msg, Byte[]
         msgdsig, Identity issuer){
 6     //get public key of issuer
 7     X509EncodedKeySpec pubKeySpec = new
           X509EncodedKeySpec(Security.getPubKey(
           issuer));
 8     KeyFactory keyFactory = KeyFactory.getInstance
           (”DSA”, ”SUN”);
 9     PublicKey pubKey = keyFactory.generatePublic(
           pubKeySpec);
10
11     //get message signature
12     Signature sig = Signature.getInstance(”
           SHA1withDSA”, ”SUN”);
13     sig.initVerify(pubKey);
14     sig.update(msg);
15
16     //verify
17     boolean verifies = sig.verify(msgdsig);
18     return verifies;
19     }
20 }
```

Listing 2. Java snippet for proof of origin as aspect

Figure 3 shows the sequence diagram of two containers notifying the back-end system. Containers one (C1) and two (C2) both send the same type of information to the army system. But C2 uses a newer protocol which includes a proof of origin to avoid tamper risk on transmission. The rectangles and their attached dotted-lines in the figure are points in architecture where our framework intervenes and injects mechanisms. With our framework, the back-end system intercepts data coming in the system and verifies it, thanks to a runtime monitor agent. It detects security protections from containers and provide to back-end services the data formatted accordingly. A taint mechanism marks data depending on its state and policy in place. Detection is made through platform implementation, such the one described in Listing 2. The listing respects policy declaration, as shown in Line 4 which bind the code - hence the behavior with the policy. The method signature is extracted from the leaf policy $verify\_origination(msg, sign, issuer)$. It then allow verification of signature. The message is tainted depending the method execution result. The piece of code is processed only upon correct matching and return information understandable by the ”runtime engine”. In our example, the origination of $data1$ cannot be verified. As the policy in Listing 1 expresses, the data is marked as $UNSAFE$. When headquarters request this data, the army system knows the data is unsafe and can propose notification mechanisms to warn user about data uncertainty.

The second use-case shows authorization and confidentiality check with our framework. The adaptation relies heavily on the context. We first explain the authorization part. The army system receives a document composed of parts with different authorization level : L1 and L2. We are in the context of a Mandatory Access Control thus strong hierarchy and definition of who is allowed to perform which action on which resource.The resource is composed of two parts. One that contains logistics information, such as freight id, container weight or event history of harbour. The second part contains details about freight : composition of the fret, final destination and usage, etc. The second part contains strategical information, that only high-ranked militaries can consult. As shown in Figure 4, a lieutenant and the logistics officer try to access resources sent by the container. The former can access all parts while the latter can only access the L2-part. When the lieutenant accesses the sensitive part of the data, the runtime monitor detects usage of a sensitive data and adapt the platform to provide confidentiality between involved peers : encryption for the lieutenant before data transmission and decryption mechanism after transmission.

Our framework intervenes from security rules upon detection of data from a specific container, the runtime monitor triggers a code that marks different parts of the data. Then, it introduces a behavior when sensitive information is transmitted. The concrete implementation of mechanisms is made locally. For instance, when the lieutenant requests the sensitive part of the data, the encryption/decryption mechanisms are executed. Systems agreed upon a behavior then implementation and execution of security mechanisms is made locally.

VII. Related work

We divide the related work in two separate categories addressing security-related solutions with aspects, or aspects for services. The former often imply modelling of security properties beforehand to latter enforce them correctly on the system. Translation mechanisms are often hand-written. The second category focuses on AOP and how to introduce its underlying concepts in services. To the best of our knowledge, no concrete work has been done to address security concerns that pervades both applications and services while proposing decoupling from the business code.

In [8], Baligand uses AOP with Web Services to introduce non-functional requirements, following a policy. The difference with our work is they do not cover simultaneous orchestration of different services to realize one capability. [9] has the same goal but proposes an XML-centric approach to specify pointcuts and advice whereas we rely on automatic matching from policy rules. In [10], Ganesan *et al.*. addresses an aspect model for composite services. They

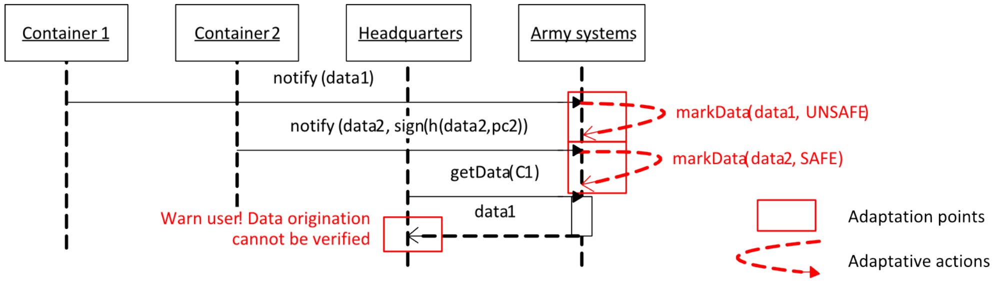


Figure 3. Multi platform adaptation

introduce a specification language to design non-functional requirements as distributed aspects, but they do not cover security *per se*. In [11], Mostéfaoui *et al.* also address a framework to decouple security concerns with aspects on web services. They use frames concept to have a configuration including both composite and component level. In [12], the authors provide an architecture to have distributed aspects to modularize and adapt non-functional requirement but only for composite services. Also, their approach implies advice code to be already on target platform for execution. In [13], Jakob *et al.* use AOP to secure distributed systems. Their approach is rather to specify early security properties thanks to pointcut language tied to an architecture diagram. [14] exposes a concrete use case of applying authentication with AOP in a SOA-based sensor architecture. Whereas it provides concrete mechanisms as we aim to do, we go a step further by binding these mechanisms with policies. It makes policy analysis way more consistent over time. In [15], Mourad *et al.* use an AOP-based language for security hardening. The language introduces concepts close to pointcuts. Therefore, the language does not cover services.

VIII. Conclusion

Addressing cross-cutting concerns that pervades services with strong focus on security lead us to a new architecture proposal. We have seen through our example that this architecture can be applied to an SCM use case. It gives tools and methods from early phase of application design to implementation and maintenance of sensors to gather accurate context information. From modelling information, ones decide what are specifications that have to be enforced during the execution of the application. In other words, the proposed architecture allows definition of security policies for service-based systems at a global level that are then enforced at a local level in an semi-automatic mode. We propose to decouple definition of specific security properties from the base application, and let declaration through rules respecting application owners' needs. A prototype is under development to address the runtime part - modification of different execution environment with Aspects to introduce security features. Currently, we limit complexity to one platform at a time. We want to investigate modifications across platforms in future work - platforms not located under a same administrative domain. It requires trust and mechanisms to ensure synchronisation, guaranties that security is effectively implemented to mention a few.

Acknowledgment

This work has been carried out in the CESSA (Compositional Evolution of Secure Services using Aspects) project, supported by the ANR, the French national research organization (project id.: 09-SEGI-002-01).

References

[1] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard, "Web services architecture,"

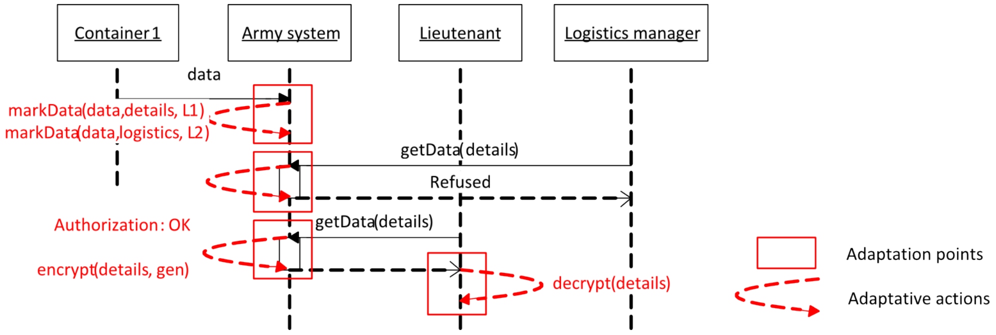


Figure 4. Authorization and confidentiality mechanisms

http://www.w3.org/TR/ws-arch/, vol. 99, no. 7, pp. 1–100, January 2004.

[2] R. Hull and J. Su, “Tools for design of composite web services,” in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, ser. SIGMOD ’04. New York, NY, USA: ACM, 2004, pp. 958–961.

[3] M. Jensen, N. Gruschka, and R. Herkenhöner, “A survey of attacks on web services,” *Informatik - Forschung Und Entwicklung*, vol. 24, pp. 185–197, 2009.

[4] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin, “Aspect-oriented programming,” in *ECOOP*, ser. Lecture Notes in Computer Science, M. Aksit and S. Matsuoka, Eds. Springer Berlin / Heidelberg, 1997, vol. 1241, pp. 220–242.

[5] C. V. Lopes, “AOP: A historical perspective (What’s in a name?),” in *Aspect-Oriented Software Development*, R. E. Filman, T. Elrad, S. Clarke, and M. Akşit, Eds. Boston: Addison-Wesley, 2005, pp. 97–122.

[6] O. G. Selfridge, “Pandemonium: a paradigm for learning. In Mechanisation of Thought Processes,” in *Proceedings of a Symposium Held at the National Physical Laboratory*. London: HMSO, 1958, pp. 513–526.

[7] L. D. B. Navarro, M. Südholt, W. Vanderperren, B. De Fraine, and D. Suvée, “Explicitly distributed aop using awed,” in *Proceedings of the 5th international conference on Aspect-oriented software development*, ser. AOSD ’06. New York, NY, USA: ACM, 2006, pp. 51–62.

[8] F. Baligand and V. Monfort, “A concrete solution for web services adaptability using policies and aspects,” in *Proceedings of the 2nd international conference on Service oriented computing*, ser. ICSOC ’04. New York, NY, USA: ACM, 2004, pp. 134–142.

[9] M. M. B. Hmida, R. F. Tomaz, and V. Monfort, “Applying aop concepts to increase web services flexibility,” in *Proceedings of the International Conference on Next Generation Web Services Practices*, ser. NWESP ’05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 169–.

[10] K. Ganesan, S. K. Mohalik, and C. Raj, “A distributed aspect model for composite service,” in *International Workshop on Service-Oriented Engineering and Optimization*, 2008.

[11] G. K. Mostéfaoui, Z. Maamar, N. C. Narendra, and S. Sattanathan, “Decoupling security concerns in web services using aspects,” *Information Technology: New Generations, Third International Conference on*, vol. 0, pp. 20–27, 2006.

[12] K. Ponnalagu, N. Narendra, J. Krishnamurthy, and R. Ramkumar, “Aspect-oriented approach for non-functional adaptation of composite web services,” in *Services, 2007 IEEE Congress on*, july 2007, pp. 284 –291.

[13] H. Jakob, N. Loriant, and C. Consel, “An aspect-oriented approach to securing distributed systems,” in *Proceedings of the 2009 international conference on Pervasive services*, ser. ICPS ’09. New York, NY, USA: ACM, 2009, pp. 21–30.

[14] S. V. Patel and K. Pandey, “Soa using aop for sensor web architecture,” in *Proceedings of the 2009 International Conference on Computer Engineering and Technology - Volume 02*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 503–507.

[15] A. Mourad, M.-A. Laverdière, and M. Debbabi, “A high-level aspect-oriented based language for software security hardening,” in *SECRYPT*, J. Hernando, E. Fernández-Medina, and M. Malek, Eds. INSTICC Press, 2007, pp. 363–370.