IARIA

SENSORCOMM 2012

The Sixth International Conference on Sensor Technologies and Applications

WSNSCM 2012

The Second International Workshop on Sensor Networks for Supply Chain Management

ISBN: 978-1-61208-207-3

August 19-24, 2012

Rome, Italy

SENSORCOMM 2012 Editors

Sergey Yurish, IFSA - Barcelona, Spain

Yenumula Reddy, Grambling State University, USA

Stephane Gervais-Ducouret, Freescale, France

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

Laurent Gomez, SAP Labs France - Mougins, France

SENSORCOMM 2012

Foreword

The Sixth International Conference on Sensor Technologies and Applications [SENSORCOMM 2012], held between August 19-24, 2012 in Rome, Italy, continued a series of events 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 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 today on the ground. 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 can not 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 2012 also featured the following workshop:

- WSNSCM 2012, The Second International Workshop on Sensor Networks for Supply Chain Management

We take here the opportunity to warmly thank all the members of the SENSORCOMM 2012 Technical Program Committee, as well as the numerous reviewers. The creation of such a 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 2012. 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 2012 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that SENSORCOMM 2012 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 technologies and applications.

We are convinced that the participants found the event useful and communications very open. We also hope the attendees enjoyed the historic charm Rome, Italy.

SENSORCOMM 2012 Chairs:

S. Biju Kumar, Philips Research - Eindhoven, The Netherlands
Canfeng Chen, Nokia Research Center - Beijing, China
Javier Del Ser Lorente, TECNALIA-Telecom - Zamudio (Bizkaia), Spain
Petre Dini, Concordia University, Canada / China Space Agency Center, China
Hristo Djidjev, Los Alamos National Laboratory, USA Teng Rui, National Institute of Information and Communication Technology, Japan
Joshua Ellul, Imperial College, London, UK
Elena Gaura, Coventry University, UK
Laurent Gomez, SAP Labs France - Mougins, France
Jens Martin Hovem, Norwegian University of Science and Technology, Norway
Sarfraz Khokhar, Cisco Systems, Inc., USA
Jaime Lloret Mauri, Polytechnic University of Valencia, Spain
Pascal Lorenz, University of Haute Alsace, France
Michael Niedermayer, Fraunhofer IZM, Germany
Alessandro Pozzebo, Università degli Studi di Siena, Italy
Yenumula Reddy, Grambling State University, USA
Harkirat Singh, Samsung Electronics Co., Korea
Mylène Toulgoat, Communications Research Centre - Ottawa, Canada
Jean Philippe Vasseur, Cisco Systems, Inc., France
Sergey Yurish, IFSA, Spain

SENSORCOMM 2012

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 2012 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 2012 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 2012 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 2012 Technical Program Committee

Saied Abedi, Fujitsu Laboratories of Europe LTD. - Middlesex, UK
Abdalrahman Al-Qubaa, Newcastle University, UK
Mothanna Alkubeily, Université de Technologie de Compiègne, France
Boushra Alkubily, Université de Technologie de Compiègne, France
Tariq Alsboui, Manchester Metropolitan University, UK
Al-Khateeb Anwar, Politecnico di Torino, Italy
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
Majid Bayanil Abbasy, Universidad Nacional de Costa Rica, Costa Rica
Rezaul K Begg, Victoria University, Australia
Paolo Bellavista, University of Bologna, Italy
Karabi Biswas, Indian Institute of Technology - Kharagpur, India
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
Chao-Tsun Chang, Hsiuping Institute of Technology, 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
David Cuartielles, Malmö University, Sweden
Debabrata Das, International Institute of Information Technology - Bangalore, India
Danco Davcev, University for Information Science & Technology "St. Paul the Apostle" - Ohrid, Republic of Macedonia
Javier Del Ser Lorente, TECNALIA-Telecom - Zamudio (Bizkaia), Spain
Jerker Delsing, Lulea University of Technology, Sweden
Behnam Dezfouli, University Technology Malaysia (UTM), Malaysia
Vincenzo Di Lecce, Politecnico di Bari, Italy
Mari Carmen Domingo, Barcelona Tech University, Spain
Wan Du, Nanyang Technological University (NTU), Singapore
Juan Carlos Dueñas López, Universidad Politecnica de Madrid, Spain
Sylvain Durand, LIRMM/Université Montpellier II, France
Imad H. Elhajj, American University of Beirut, Lebanon
Joshua Ellul, Imperial College London, UK
Xiang Fei, Coventry University, UK
Sándor Fekete, Braunschweig Institute of. Technology, Germany

Paulo Felisberto, Institut for Systems and Robotics-Lisbon / Universidade do Algarve, Portugal
Gianluigi Ferrari, University of Parma, Italy
Armando Ferro Vázquez, Universidad del País Vasco - Bilbao, Spain
Paul Fortier, University of Massachusetts Dartmouth, USA
Miguel Garcia Pineda, Polytechnic University of Valencia, Spain
Elena Gaura, Coventry University, UK
Hamid Gharavi, National Institute of Standards and. Technology (NIST) - Gaithersburg, USA
Chris Gniady, University of Arizona, USA
Stephane Grumbach, INRIA, France
Jianlin Guo, Mitsubishi Electric Research Laboratories - Cambridge, USA
Malka N. Halgamuge, The University of Melbourne, Australia
Mohammad Hammoudeh, Manchester Metropolitan University, UK
Vincent Huang, Ericsson Research - Stockholm, Sweden
Abhaya Induruwa, Canterbury Christ Church University, UK
Vasanth Iyer, International Institute of Information Technology, India
Shaghayegh Jaberi, Islamic Azad University - Tehran, Iran
Imad Jawhar, United Arab Emirates University - Al Ain, UAE
Zhen Jiang, West Chester University, USA
Adrian Kacso, University of Siegen, Germany
Aravind Kailas, University of North Carolina - Charlotte, USA
Kyoung-Don Kang, Binghamton University, USA
Riad Kanan, The Institute of Engineering Sciences, Switzerland
Dimitrios A. Karras, Chalkis Institute of Technology, Hellas
Fotis Kerasiotis, University of Patras / Rio-Patras, Greece
Sarfraz Khokhar, Cisco Systems Inc., USA
Thorsten Kramp, IBM Research Zurich, Switzerland
Evangelos Kranakis, Carleton University - Ottawa, Canada
Srđjan Krčo, Ericsson Research, Ireland
Danny Krizanc, Wesleyan University - Middletown, USA
Erlend Larsen, The Norwegian Defence Research Establishment (FFI) - Kjeller, Norway
Seongsoo Lee, Soongsil University - Seoul, Korea
Chiu-Kuo Liang, Chung Hua University - Hsinchu, Taiwan
Qilian Liang, University of Texas at Arlington, USA
Weifa Liang, Australian National University - Canberra, Australia
Thomas Lindh, STH/KTH - Stockholm, Sweden
André Luiz Lins de Aquino, Federal University of Ouro Preto, Brazil
Hai Liu, Hong Kong Baptist University, Hong Kong
Jaime Lloret Mauri, Polytechnic University of Valencia, Spain
Elsa María Macías López, University of Las Palmas de Gran Canaria, Spain
Abdallah Makhoul, Université de Besancon - Belfort, France
Vladimir Marbukh, NIST, USA
José Ramiro Martínez de Dios, University of Seville, Spain
Alireza Masoum, Twente University, The Netherlands
Natarajan Meghanathan, Jackson State University, USA
Nader Faisal Jaafar Mohamed, UAEU, UAE
Jose M. Moya, Universidad Politécnica de Madrid, Spain
Deok Hee Nam, Wilberforce University, USA
Enrico Natalizio, INRIA Lille - Nord Europe, France

Wilfried Uhring, University of Strasbourg, France
Fabrice Valois, INRIA SWING / CITI, INSA-Lyon, France
Jean Philippe Vasseur, Cisco Systems, Inc., France
Armin Veichtlbauer, Salzburg Research Forschungsgesellschaft m.b.H., Austria
Roberto Verdone, Università di Bologna, Italy
Manuela Vieira, UNINOVA/ISEL, Portugal
Michael 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
Dongfang Yang, National Research Council Canada - London, Canada
Rehana Yasmin, University of Birmingham, UK
Chih-Wei Yi, National Chiao Tung University, Taiwan
Norihiko Yoshida, Saitama University, Japan
Sergey Y. Yurish, IFSA - Barcelona, Spain
Yifeng Zhou, Communications Research Centre, Canada
Tanveer Zia, Charles Sturt University, Australia

WSNSCM 2012 PROGRAM COMMITTEE

Workshop Chair

Laurent Gomez, SAP Labs France - Mougins, France

WSNSCM 2012 Technical Program Committee

Omar Gaci, Institut Supérieur d'Etudes Logistiques, France
Mehdi Khalfaoui, EUROCOM, France
Annett Laube, Bern University of Applied Science, Switzerland
Maryline Laurent, Institut Telecom / TELECOM SudParis - Evry, France
Jean-Frederic Myoupo, University of Picardie-Jules Verne, France
Joachim Possega, Universtät Passau, Germany
Wilfrid Real, IBM La Gaude, France
Gabriel Serme, EUROCOM, France
Alessandro Sorniotti, IBM Research Zürich, Switzerland
Slim Trabelsi, SAP Labs France - Mougins, 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

Development of a Wireless Sensor Node for Environmental Monitoring

Daryoush Bayat, Daryoush Habibi and Iftekhar Ahmad
School of Engineering
Edith Cowan University
Joondalup, Western Australia, Australia
{d.bayat, d.habibi, i.ahmad}@ecu.edu.au

***Abstract*—The Wireless Sensor Network (WSN) has emerged as an exciting research field in recent years and has the potential to revolutionize applications such as industrial monitoring and automation, environmental monitoring and biomedical science. Wireless sensor networks achieve these objectives using sophisticated networks of low cost smart sensors that co-operatively monitor and communicate information such as gas concentration, humidity, temperature, etc. One of the shortfalls of the existing wireless sensor network technology is limited coverage area. Environmental monitoring applications such as bushfire detection require vast coverage areas for wireless sensor networks to be economically effective. In this paper, we present the architecture and characteristics of a long range wireless sensor network targeted for environmental monitoring applications. Firstly, the hardware design and software implementation are explained and then the achieved results are evaluated. The developed wireless sensor node can cover several hundreds of kilometers by using a proprietary mesh protocol.**

Keywords-wireless sensor network; DIGIMESH; long range; 9-Xtend

I. INTRODUCTION

Wireless sensor networks have become an important field in wireless technology**.** Wireless sensor networks have certain features that differentiate them from other wireless networks. A typical wireless sensor network has processing and sensing capabilities in addition to the expected communication capabilities. Wireless sensor networks also have very stringent power consumption requirements and have very limited computational and storage capacities. A typical wireless sensor network should be able to function unattended for extended periods of time, usually a number of months off say two AA batteries.

The challenges involved in the design of WSN are how to minimize power consumption and increase the computational efficiency. Most of the power in WSN is consumed during data communication. It is, therefore, very important to implement efficient networking algorithms.

Some of the interesting applications of wireless sensor networks include habitat monitoring [1], environmental monitoring [2], and biomedical applications [3]. The design choice of a wireless sensor network is very application dependant. For example, for environmental monitoring applications we ideally need a wireless sensor network that can cover large geographical areas, whereas in biomedical applications, the wireless sensor network should be tiny and accurate. Nevertheless, the design of the wireless sensor networks should be modular to some extent, in order for the development and manufacturing to be economically viable.

One of the shortfalls of the currently developed sensor nodes is that they can only cover small geographical areas due to their limited transmission range. For applications such as environmental monitoring, the coverage area needs to be at least several kilometers. One example of environmental monitoring application is bushfire detection. A bushfire monitoring system should cover tens of kilometers in order to be effective. Even though the short transmission range of existing wireless sensor networks can be compensated by deploying a mesh topology, it would be economically infeasible to have thousands of densely deployed sensor nodes to monitor bushfire. In fact, one of the advantages of WSN is its low deployment cost.

One possible solution to address the transmission range limitation of the existing sensor nodes is to connect the local wireless sensor network to a larger network, such as the internet, through a gateway node. The weakness of this solution is that it relies on pre-existing infrastructure. For example in a forest-fire detection application, the sensor nodes are scattered in remote areas where there is no other network coverage. In this situation, the nodes must be able to route the messages without relying on any existing network infrastructure.

Another solution is to modify the transceiver hardware of the existing sensor nodes to increases their receiver sensitivity. The JCUMote [4], a sensor node based on the MICAz achieves this through the use of RF power amplifiers. The advantage of this solution is that, it utilizes the already developed technology in the MICAz, but the increased transmission range is still not sufficient for applications such as forest-fire monitoring.

The aim of this paper is to develop a wireless sensor network with vast coverage to be used for environmental monitoring applications. In order to achieve this, we firstly developed the hardware platform. We then developed the node logic or software and integrated a long range transceiver with advanced built-in networking capabilities into the system.

The field tests results are very promising with approximately 22 kilometers of line-of-sight range. The

ISBN: 978-1-61208-207-3

implementation of a mesh network has greatly extended the coverage area and has resulted in enhanced reliability and robustness of the overall network.

II. METHODOLOGY

A. Node Architecture

The hardware of the developed sensor node is composed of two sensors, transceiver, antenna, expansion connectors, external memory and a microcontroller. The developed sensor node interfaces with a PC via a USB connector, and can be reprogrammed using the dedicated on-board programmer. A block diagram of the sensor node architecture is shown in Figure 1.

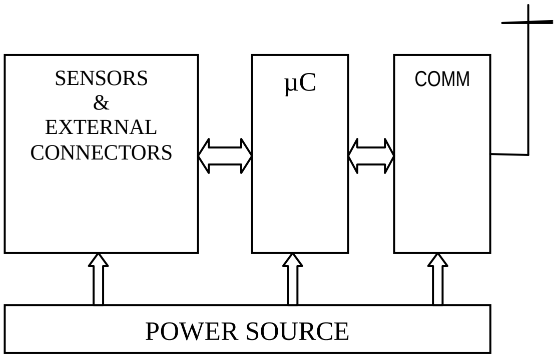


Figure 1. Architecture of the Sensor Node.

B. Sensors and Expansion Connectors

There are two analog sensors integrated into the developed wireless sensor node, namely, humidity and temperature sensors. These sensors were chosen because of their popularity in many environmental monitoring applications.

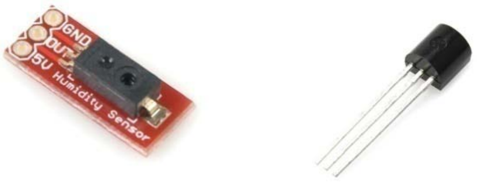


Figure 2. HIH-4030 and LM335 Sensors.

HIH-4030 [5] from Honeywell was chosen as the humidity sensor. HIH-4030 is an analog humidity sensor that can be connected directly to the microcontroller due to its near linear output. The special packaging provides protection against environmental hazards such as condensation, dirt, dust, and other chemicals.

According to the datasheet, output voltage and relative humidity are related according to the following equation.

$$V_{out} = (V_{cc}) (0.0062(RH) + 0.16 \qquad (1)$$

where V_{out} stands for the output voltage of the sensor, RH for relative humidity in percentage and V_{cc} for the supply voltage. The output of the sensor is connected to one of the ADC channels on the microcontroller via a current limiting resistor.

In order to compensate for temperature changes, which affect the precision of the sensor, the following equation is used.

$$\text{True RH} = (\text{Sensor RH}) / (1.0546-0.00216T) \qquad (2)$$

where True RH stands for the compensated relative humidity and T for the temperature in degrees Celsius.

LM335 [6] from National Instruments was chosen as the temperature sensor. LM335 is an analog temperature sensor that operates over a -55 degrees Celsius to 150 degrees Celsius temperature range. It can be directly connected to the microcontroller due to its near linear output response.

According to the datasheet the calibrated output of the sensor and temperature in Celsius are related according to the following equation.

$$T(C) = T_0 (K) (V_{out}/ V_{outT0}) – 273 \qquad (3)$$

where T(C) stands for temperature in degrees Celsius, T_0 (K) for a reference temperature in Kelvin, $V_{outT0}$for the output voltage at the reference temperature and V_{outT} for the output voltage at any temperature. The chosen reference temperature for this sensor was 298 degrees Kelvin and the corresponding output voltage was measured to be 2.94 volts.

The developed sensor node has expansion connectors for two analog sensors, two SPI compatible sensors and three I^2C compatible sensors. I^2C and SPI interfaces were implemented because of their popularity in digital sensors. An external EEPROM is also connected to the microcontroller via I^2C interface.

C. Processor

ATmega168 is the chosen processor for the developed wireless sensor node due to its user friendly interface and author's prior experience in programming Atmega family of microcontrollers. The important factors in choosing the processor were: power consumption, built-in interfaces for UART, SPI, I^2C, speed, and size.

D. Communication and Networking

The transceiver is where the communication and networking take place in the sensor node. The important factors for choosing the transceiver were transmission range, network topology support, power consumption and size. The Xbee-PRO from Digi International, the RMX-232 from Embedded Communication Systems and the 9-Xtend from MaxStream [7], [8], [9] were evaluated. The 9-Xtend RF radio module was the chosen transceiver due to its superior transmission range and support for mesh networking. The transceiver is connected to the microcontroller via UART interface. The following table summarizes the characteristics of the evaluated transceivers.

TABLE I. SUMMARY OF EVALUATED TRANSCEIVERS.

	Xbee-PRO	RMX-232	9-Xtend
Power Consumption	954 mW	630 mW	3650 mW
Dimension (cm)	3.29x2.44x0.546	8.5x6.5x2.5	3.65x6.05x0.5
range	9.6 Km	2 Km	64 Km

DIGIMESH [10], a proprietary mesh network protocol, is the networking protocol used in the developed sensor node. In a mesh network, messages are routed through several nodes in order to reach the final destination. Other than the extended range, DIGIMESH offers a unique set of capabilities that makes it suitable for our sensor node. Some of the important features of DIGIMESH include:

- Self healing
- Flexibility to expand network
- Elimination of expensive gateway routers
- Reliability

Self healing means that during node failures, the network can find an alternative path to the destination. Flexibility to expand the network is due to the fact any node can be added and removed from the network without affecting the functionality of the network as a whole. The homogenous nature of the network results in equal functionality of all the nodes; therefore there is no need for any gateway node with enhanced functionality. This makes the configuration of the network substantially easier. Finally, reliability is achieved through the use of acknowledgments and retransmissions.

The routing algorithm in DIMIMESH is very similar to the AODV (Ad hoc On-demand Distance Vector) [22] algorithm. There is an associative routing table for each node that maps the destination address to its next hop address. Thus messages from the source node will go through possibly several nodes until they reach the destination node. When the source node doesn't have a route for the specified destination, it will initiate a Route Discovery process.

During the Route Discovery process, the source node broadcasts a Route Request message. Upon reception of Route Request message, the intermediate nodes rebroadcast the Route Request message if they don’t have a better route back to the source node, otherwise they drop the message. When the destination node eventually receives the Route Request message, it unicasts a Route Reply message back to the source node. Therefore, the source node might receive multiple Route Reply messages. It will choose the one with the best round trip route quality. It should also be noted that the destination address of the source nodes has to match with the source address of the destination node.

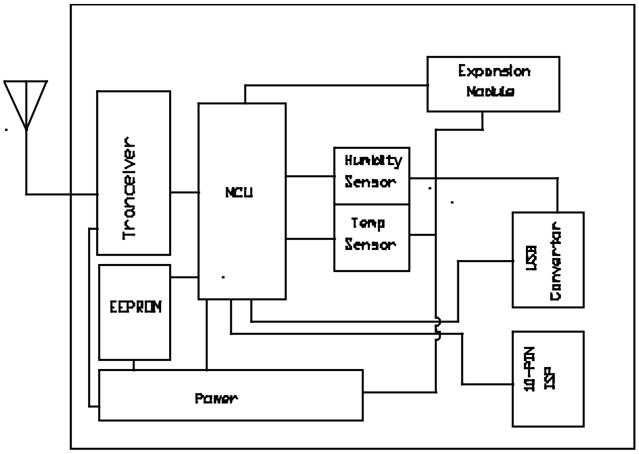


Figure 3. Detailed Block Diagram of the Sensor Node.

E. Power Consumption

There are several wireless sensor nodes already available in the market such as TelosB [11], MicaZ [12], Waspmote [13], etc, all of which have low power consumption, but they have limited transmission range, usually several hundred meters. There are other long range transceivers available, but they are targeted for industrial applications and require constant power supply, and are usually very bulky and expensive.

F. Final Design

A detailed block diagram of the developed sensor node is shown in Figure 3. A PCB for the complete design, including the microcontroller, transceiver, sensors and other peripherals was designed using ExpressPCB [11] manufacturer as shown in Figure 4. The PCB has two copper layers and is 80mm by 70mm in size. A five volt battery is housed underneath the PCB.

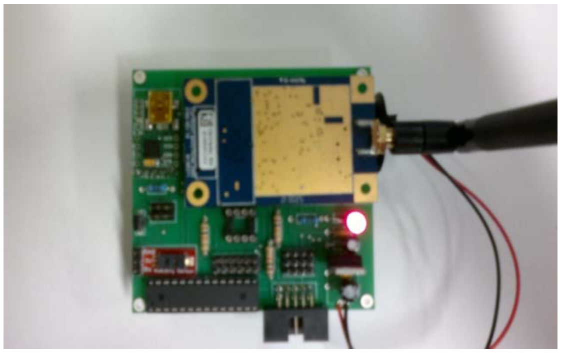


Figure 4. Completed Sensor Node.

G. Node Logic

The node logic is shown in the following flowchart. In the first step the node is initialized. During node initialization, the ADC, UART, I^2C and SPI modules are configured and the associated registers are initialized. Then, the sensors are sampled including any external sensors attached to the sensor node. The readings from the sensors are digitalized and calibrated. The calibrated results are compared against predefined thresholds and if they exceed, an alarm message is generated and sent to the transceiver via the UART interface. The transceiver then takes care of packetisation and RF transmission of the message to the base-station. After a brief wait, the above procedure is repeated.

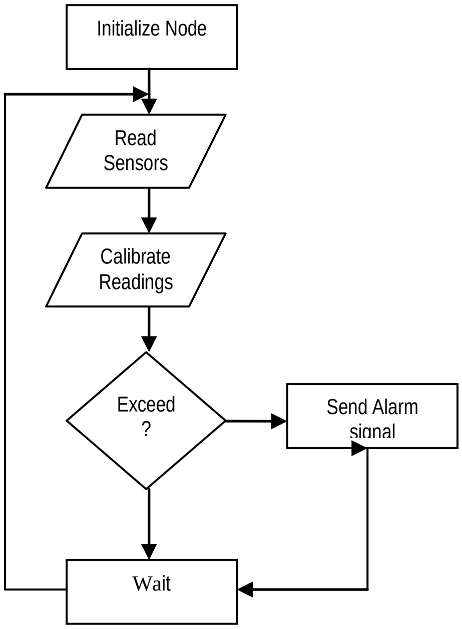


Figure 5. Sensor Node Logic.

III. RESULTS

The modulation technique used in the 9-Xtend RF radio module is BFSK (Binary Frequency Shift Keying) [23]. In BFSK, digital data is carried using discrete variations in the carrier frequency. In other words, we will represent 0s with one frequency and 1s with another frequency. At the same time, the 9-Xtend RF radio module also utilizes Frequency Hopping Spread Spectrum (FHSS) [24]. In FHSS, the carrier frequency is varied within the available bandwidth according to a pseudo-random function. Consequently, the carrier signal will change with time and is no longer fixed. Figures 6 and 7 demonstrate two different snapshots of frequency spectrums measured by portable spectrum analyzer at a close distance from the sensor node.

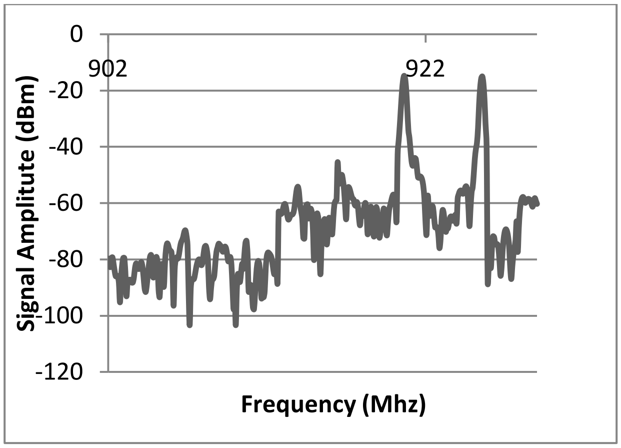


Figure 6. Frequency Spectrum of the Sensor Node at t0.

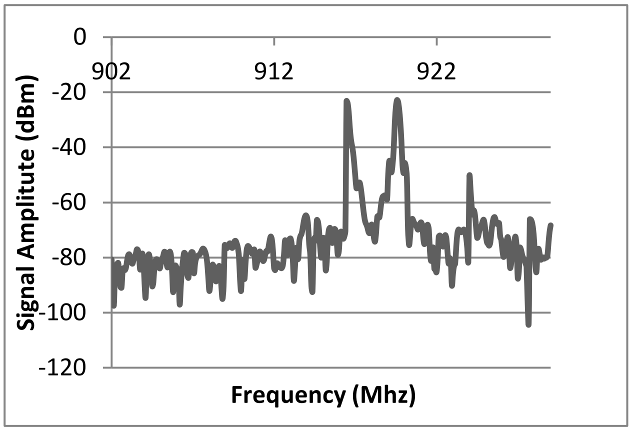


Figure 7. Frequency Spectrum of the Sensor Node at t1.

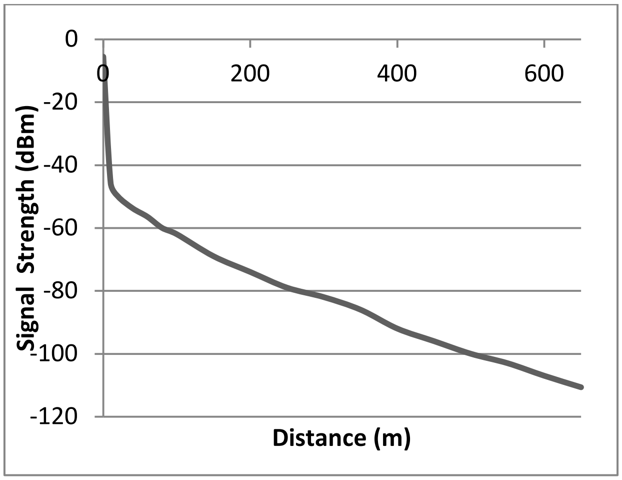


Figure 8. Residential Area.

From figures 6 and 7, it is evident that 0s and 1s are transmitted using two different frequencies (i.e., two peaks around -20 dBm) and the corresponding frequency for 0s and 1s varies with time. These features reduce the amount of interference from other sources and improve the security of the developed sensor node.

The transmission range of the developed sensor node was tested under three different environmental conditions. The maximum line of sight range is almost 22 km. These tests were conducted at 9600 bits per second RF throughput, 1 watt antenna output power and using a dipole antenna with gain of 2.1 dBm. The following graphs demonstrate the signal strength vs. distance at different conditions. The maximum range in vegetated areas is 1100 m

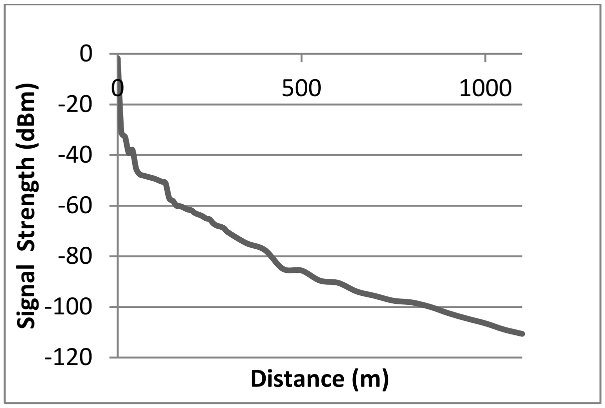


Figure 9. Densely Vegetated Area.

IV. CONCLUSION AND FUTURE WORK

This paper proposed the development of a long range wireless sensor network for environmental monitoring applications. Initially, the hardware of the sensor node was designed and then a long range RF radio was integrated into the system. The developed sensor node was further enhanced by deploying the DIGIMESH protocol to improve range and overall reliability and flexibility of the proposed wireless sensor network. Future works include developing an algorithm for improving power consumption efficiency and incorporating a 2.4GHz radio transmitter so that the node can be used for both long and short range communications.

REFERENCES

[1] T. Naumowicz et al., “Wireless Sensor Network for habitat monitoring on Skomer Island,” in *LCN*, Denver, 2010, pp. 882-889.

[2] S. Rachman, I. Pratomo, and N. Mita, “Design of low cost wirelesssensor networks-based environmental monitoring system for developing country,” Proc. APCC, Tokyo, 2005, pp. 1-5.

[3] G. Arrobo and R. Gitlin, “New approaches to reliable wireless body area networks,” Proc. IEEE International Conference on Microwaves, Communications, Antennas and Electronics System, USA, 2011, pp. 1-6.

[4] S. Willis and C. Kikkert, “Design of a Long-Range Wireless SensorNode,” Proc. APCCAS, Singapore, 2006, pp. 151-154.

[5] [retrieved: August, 2012] HIH-4030 datasheet. [Online]. http://sensing.honeywell.com/index.php?ci_id=51625

[6] [retrieved: August, 2012] LM335 datasheet. [Online]. http://www.national.com/ds/LM/LM135.pdf

[7] [retrieved: August, 2012] Xbee-PRO datasheet. [Online]. http://www.digi.com/pdf/ds-xbee-pro_pkg-rf-modems.pdf

[8] RMX 232 datasheet. (2011, November) [Online]. http://www.embeddedcomms.com.au/download/rm-232bb%20supplement.pdf

[9] [retrieved: August, 2012]9-Xtend datasheet. [Online]. ftp://ftp1.digi.com/support/documentation/productmanual_xtend_oem_rfmodule.pdf

[10] [retrieved: August, 2012] DIGIMESH Protocol Specifications.[Online]. http://www.digi.com/technology/digimesh/

[11] [retrieved: August, 2012] TelosB datasheet. [Online]. http://www.willow.co.uk/TelosB_Datasheet.pdf

[12] [retrieved: August, 2012]MicaZ datasheet. [Online]. http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf .

[13] [retrieved: August, 2012]Waspmote datasheet. [Online]. http://www.libelium.com/documentation/waspmote/waspmote-datasheet_eng.pdf

[14] [retrieved: August, 2012] ExpressPCB. [Online]. http://www.expresspcb.com/

[15] [retrieved:August 2012] N.Xu. University of Southern California.[Online]. http://enl.usc.edu/~ningxu/papers/survey.pdf

[16] S. Pratomo, I. Mita , and R. Wirawan, “Design of Low cost wireless sensor network-based environmental monitoring system for developing country,” Proc. APCC, 2008, pp. 1-5.

[17] H. Wang, W. Wang, and S. Hua, “Adaptive Data Compression in Wireless Body Sensor Networks,” Proc. IEEE International Conference on Computational Science and Engineering, 2010, Hong Kong, pp. 1-5.

[18] [retrieved:August 2012] HIH-4030 datasheet. [Online]. http://sensing.honeywell.com/index.cfm?ci_id=140301&la_id=1&pr_id=145616

[19] [retrieved:August 2012] LM335 datasheet. [Online]. http://www.national.com/mpf/LM/LM335.html

[20] [retrieved:August 2012] [Online]. http://www.digi.com/technology/digimesh/

[21] [retrieved:August 2012] [Online]. http://www.expresspcb.com/.

[22] C. Perkins, E. Royer and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing,” in proc. IEEE Workshop on Mobile Computing Systems and Applications, USA, 1999, pp. 90-100.

[23] S. Hassan and M. Ingram, “SNR Estimation for a Non-Coherent Binary Frequency Shift Keying Receiver,” Proc. IEEE Globecom, 2009, USA, pp. 1-5.

[24] S. Zoican “Frequency hopping spread spectrum technique for wireless communication systems,” Proc. IEEE 5th International Symposium on Spread Spectrum Techniques and Applications, 1998, South Africa, pp. 338-341.

Data Reduction Algorithm on the Monitoring of Extreme Values in WSNs

Chun-Lung Lin
Industrial Technology
Research Institute
HsinChu, Taiwan R.O.C.
Email: chunlung@itri.org.tw

Pei-Hsuan Tsai
Institute of Manufacturing
Information and Systems
National Cheng Kung University
Tainan, Taiwan R.O.C.
Email: phtsai@mail.ncku.edu.tw

Hsiao-Chuan Liang and Jia-Shung Wang
Department of Computer Science
National Tsing Hua Univeristy
HsinChu, Taiwan R.O.C.
Email: jswang@cs.nthu.edu.tw

***Abstract*—Monitoring extreme values (maximum or minimum) is important to many applications in wireless sensor networks. A previous work, called Hierarchy Adaptive Threshold (HAT), proposed a tree-based structure to distribute queries efficiently and filter out the unnecessary data updates that are not extreme values. In this paper, a data reduction algorithm is presented to reduce energy consumption of the HAT due to network transmission. The proposed method utilizes historical information of extreme values and their corresponding node ID to adjust the reporting rate of sensors properly and eases the burden of the parent of extreme nodes by balancing the packets from extreme nodes to all their possible parents. We evaluate the performance of the proposed algorithm by NS-2 network simulator and real-world data traces. The results indicate that the overall network packets are reduced to 80% with 1% data error in comparison with HAT.**

***Keywords-sensor networks*; *extreme values*; *data reduction*.**

I. INTRODUCTION

Wireless sensor networks (WSNs) have been widely applied to many current and envisioned applications, including vehicle tracking and habit monitoring [1], [2], nuclear reactors, test areas, disaster management, combat field surveillance, factory temperature monitoring, border control and so on. A maximum (or minimum) query is a query, which continuously requests the sensor node with the maximum (minimum) sensor reading. That is, a maximum query continuously maintains the (node id, value) pair in the network. Monitoring maximum (or minimum) value is important for detecting abnormal or extreme behavior in many sensor network applications. For example, monitoring maximum temperature in a factory is essential to improve production yield, or finding out the node and its corresponding areas with the highest pollution index for the purpose of pollution control. A minimum query, on the other hand, can be requested to continuously monitor the sensor node with the least residual energy so that it can be instructed to adapt its sampling rate (or transmission rate) for extending network lifetime. Since the discussions of maximum and minimum queries are similar, in the following we will only focus on answering the continuous maximum monitoring problem.

The difficulty of answering a maximum query is unable to know the sensors that constitute high values in priori. Hierarchy Adaptive Thresholds (HAT) was proposed to monitor the maximum query efficiently [3]. HAT prevents the nodes which will not be maximum nodes from transmitting by giving constraints on nodes such as filter threshold. However, there is a penalty where additional queries are required to be issued when the sink cannot definitely decide the current maximum value. Moreover, the performance of HAT highly depends on the update strategy of filter thresholds at nodes. On the basis of HAT, a novel filter-based monitoring algorithm is proposed in this paper. According to the observations on real data, the locations and values of the maximum nodes are usually stable. That is,

1) Parts of nodes inside the network become the maximum node more often than the others. This suggests that some nodes are always likely to become the maximum node, and in contrast, some are unlikely to be the maximum one.
2) The maximum node often remains unchanged for some consecutive time intervals (i.e., temporal correlation between sensor readings).
3) When a new maximum node emerges, it is often nearby the previous maximum node (i.e., spatial correlation among nearby nodes).

In light of the phenomenon above, the proposed algorithm utilizes historical information of the past maximum values and the nodes that generate them. Specifically, the algorithm measures the probability of a node being the maximum node by using historical information. The quantitative relationship between reporting rate and filtering interval is also measured. According to the probability, nodes are dynamically instructed to re-adjust their filtering threshold so as to meet the required reporting rate (e.g., decrease the reporting rates of nodes with lower probability and increase the reporting rates of those with higher probability). The typical goal is to prolong the network lifetime by reducing unnecessary data updates.

The rest of this paper is organized as follows. The related work is given in Section 2. Section 3 presents the proposed algorithm. Simulation results are described in Section 4. Finally, the conclusion is drawn in Section 5.

II. Related Work

Monitoring aggregation functions (such as average, maximum, and minimum) in sensor networks have received extensive attention in the past few years. However, the main focus has been on how to establish the communication structure and how to apply aggregation techniques to reduce network traffic [5], [6]. Work has also been done on compressing historical sensor readings for transmission [7], [8]. These methods are applicable to archival data collection where the application wants to log historical sensor readings and analyzes them later. In contrast, we consider monitoring applications that continuously request up-to-date sensor readings. Approximate monitoring schemes have been proposed for average and sum aggregates in [9], [10]. Yoon and Shahabi [11] proposed a clustered aggregation (CAG) algorithm for approximate query processing. CAG reduces network traffic by forming clusters of nodes where the cluster heads are responsible for aggregating the readings of their children.

Another work for approximate query processing is range caching. There is a value range installed at each node and the base station caches all the values of ranges at nodes. A node updates its new reading with the sink only if the difference between the new value and the previously reported value beyond the range. The nodes are sorted by range upper bounds, and on receiving a maximum query, the sink searches each node in order one by one. This approach is costly and inefficient because it does not take aggregation into consideration. A maximum query can be a special case of top-k query, and studies on evaluating snapshot top-k queries in distributed networks are included in [12]–[15].

In this paper, we are interested in monitoring a continuous maximum query. Although continuous monitoring could be simulated by repeatedly executing a snapshot query, many snapshot queries would be costly and inefficient if the answers remain unchanged. Recent work on monitoring continuous top-k queries in distributed networks are described in [3], [16]–[19]. In [18], a filter-based approach called FILA was proposed to exploit the semantics of top-k query. The basic idea of FILA is to install a filter at each sensor node to suppress unnecessary sensor updates. Silberstein et al. [3] explored techniques for continuously monitoring a maximum query, with the objective of minimizing network traffic. They proposed a HAT approach, which use suitable constraint settings to prevent nodes unlikely to have the maximum value from transmitting. However, HAT does not take historical information of the maximum values into account.

III. The Proposed Algorithm

As mentioned above, the purpose of the proposed algorithm is to reduce network packets when running HAT. The proposed monitoring algorithm of extreme values is on the basis of HAT algorithm. The significant improvement of the proposed method over the HAT is that it utilizes historical information of the maximum values (and the maximum nodes) to measure the probability of a node being the maximum node. The quantitative relationship between reporting rate and filtering threshold is also measured. According to the probability, nodes are dynamically instructed to re-adjust their filtering thresholds so as to meet the required reporting rate.

Consider a wireless sensor network that includes n fixed location sensors. The network is rooted at the sink with a continuous power supply, and the sensor nodes are powered by battery. It is assumed that there is a tree-based communication infrastructure installed on the network by which nodes beyond the transmission range of the sink can send their data to the sink [31]. Each sensor is assumed to periodically sample the local phenomenon, such as temperature, humidity and pollution index, at a fixed rate. Without loss of generality, the period between two successive samplings is assumed to be one time unit (round). That is, we assume the query is processed repeatedly over a series of rounds, where each node generates a value in each round. Each round is long enough for all necessary messaging to occur in order to complete the query.

In this paper, we consider a maximum query that continuously queries the sensor node with the maximum sensor reading at each round. The maximum query is to continuously maintain the (node id, value) pair for the node with the maximum value in the network. The monitoring result is logged at the sink and provided to external users. The monitoring algorithm of extreme values is to control when and how sensor readings should be sent to the sink to continuously generate maximum query results.

A. Procedure of the proposed algorithm

1) Initialization: Let C_i denote the set of the immediate children of a sensor node s_i and p_i denote the parent node of s_i in the network communication infrastructure, respectively. Denote the reading of sensor i and the maximum value at the t-th round as v_t^i and v_t^{max} respectively. Initially, each sensor node sends its reading to its parent node. When an internal node s_i receives all readings of its child nodes, it obtains the maximum value in the sub-tree rooted at s_i by sorting the sensor readings, including its own reading. That is, node s_i aggregates the messages of its child nodes and itself by only passing the packet with the highest value since it is impossible for those values to be the maximum. Then the node s_i computes a filter threshold for itself and each of its child nodes. The thresholds are sent to all of its child nodes. Similarly, we denote h_t^j as the filter threshold of node s_j at the t-th round. Note that the filter threshold of a node is by known to both itself and its parent node. This message aggregation is bottom-up, beginning at leaf nodes until the sink is reached. Accordingly, the sink obtains the initial query result, i.e., v_0^{max}, by collecting the readings from all sensor nodes. Once a maximum node is obtained, the sink will also send a message to designate the node reporting the maximum value as the current maximum node, denoted by s_0^{max}.

2) Sequential Rounds: Each subsequent round proceeds in three stages: *node-initiated report*, *root-initiated query* and *reporting rate adjustment*.

Node-initiated report: If a node s_k is the designated maximum at the last round (t-th round), i.e., $s_k = s_t^{max}$, it transmits an update containing the (node id, new value) pair to the sink only if $v_{t+1}^k \neq v_t^{max}$. (i.e., the new gathered value v_{t+1}^k differs from the maximum value in the last round.) For a node s_j, $s_j \neq s_t^{max}$, if the new reading at the $(t+1)$th round is smaller than its filter threshold h_t^j, then no update is sent to its parent node. Otherwise, an update is sent to its parent node. When an internal node s_i receives any update from its child nodes, it computes the new maximum value v_i^* in the sub-tree rooted at s_i. If $v_i^* > h_t^i$, an update with the new value v_i^* is sent to its parent; otherwise, all packets in the sub-tree rooted at s_i will be filtered out to reduce network traffic. Namely, node s_i aggregates the messages of it child nodes by only passing the packet if the highest value breaks its filter threshold. Then s_i will also update its threshold h_{t+1}^i and the filter thresholds h_{t+1}^j for its child nodes that send an update by sending a threshold update packet to them. Obviously, the threshold h_{t+1}^j is known to both sensor s_j and its parent node s_i.

Root-initiated query: If the designated maximum node s_{max} in the last round do not report, it can be known in the sink that the reading of s_{max} remains unchanged in the current round. Then the stored value of v_{max} in the last round can be used to evaluate the new maximum value. Once all of the update packets are received at the sink, it determines v_{sink}^* from the set of all returned values. It can be verified easily that if the reading of s_{max} stays the same or rises, then v_{sink}^* is the maximum value since all values higher than v_{sink}^* will definitely overtake their filter threshold. On the other hand, a node can become the new maximum node without breaking its filter threshold only if the old maximum value falls. In the case, it is possible that the new maximum value can only be discovered through root-initiated querying process. More specifically, the root sends query messages containing the temporary maximum value v_{sink}^* to those of its child nodes with filter threshold greater than v_{sink}^*. In turn, each node receiving a query packet only forwards it to its child nodes with thresholds greater than v_{sink}^*. On receiving the query packet, only those nodes with values greater than v_{sink}^* send their new readings to the sink. Similarly, all reply messages are aggregated at internal nodes along the transmission path and only the maximum of them is forwarded upward. Moreover, the filter thresholds of nodes along the transmission are also updated accordingly.

Table I
GLOBAL MAXIMUM TIMES TABLE FOR 70 ROUNDS

Node	1	2	3	4	5	6	7	8
Frequency	5	0	5	0	0	0	40	20

Reporting rate adjustment: Here we assume that the sink maintains a probability value γ_t^i for each node s_i at the t-th round. (We postpone the detailed discussion of γ_t^i in the next subsection.) After deciding the maximum value in the current round, the sink will re-evaluate γ_t^i and check whether $\gamma_t^i < \gamma$ or not for all nodes in the network, where γ is a given system parameter. If $\gamma_t^i < \gamma$, then s_i is classified as an unimportant node that has very low probability to become the maximum node in the next round. Therefore, the sink sends a reporting rate adjustment packet containing γ_t^i to the nodes with $\gamma_t^i < \gamma$. On receiving an reporting rate adjustment request, node s_i raises its filter threshold by

$$h_{t+1}^i = (1 - \gamma_t^i) \cdot v_t^{max} + \gamma_t^i \cdot h_t^i \tag{1}$$

B. Classification of Important and Unimportant Nodes

In this subsection, a strategy will presented to classify the nodes as important and unimportant nodes. In real world, most scenarios, the locations where the maximum values occurred are quite stable (i.e. the distribution of monitored value would not change rapidly). For example, the maximum temperature of a factory usually stays in the same region and the maximum humidity of a rainforest may stay in the same region as well. With this observation, we can take advantage of this phenomenon. We endow the sink node with the ability to learn the importance of all nodes. In short, we deem that a node is important if it becomes maximum node frequently. Similarly, a node is classified as an important node if it has very low chance to become maximum node. Hence, the sink node will keep a *maximum times table* used to store the times of each node being the maximum node.

According to this maximum times table, the sink node could differentiate important nodes from all nodes. However, in most cases, recent observations are much more important than older observations. Therefore, an additional table is also utilized to store recent observations in the sink node. The older observations are in *global maximum times table*, and the recent observations are stored in *temporal maximum times table*. Table I and Table II show an example of global maximum times table and temporal maximum times table. By using the maximum times tables, a node could be classified by computing the important factor as follows.

$$\gamma_i = w \cdot P(G_i) + (1 - w) \cdot P(T_i), w \in [0, 1] \tag{2}$$

γ_i represents the importance of a node (the probability of this node to become maximum node). $P(G_i)$ and $P(T_i)$ denote the probabilities of node s_i being the maximum node in Global Maximum Times Table and Temporal Maximum Times Table, respectively. And w is the weight of Global

Table II
TEMPORAL MAXIMUM TIMES TABLE FOR RECENT 20 ROUNDS

Node	1	2	3	4	5	6	7	8
Frequency	0	0	0	0	0	0	15	5

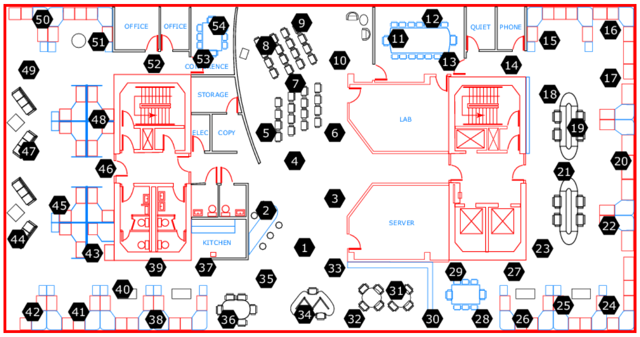


Figure 1. Sensor deployment in Intel Berkerly lab dataset.

Maximum Times Table. In practice, we would set a threshold to γ_i. A node s_i would be classified as an important node, if γ_i break this threshold. Otherwise, it would be classified as an unimportant node.

C. *Adjustment Strategy of Reporting Rate*

Three strategies will be adopted in the proposed method to dynamically adjust the reporting frequency of sensors according to their classification [21]–[30].

1) Round basis: The round basis strategy is to transmit the regulation packets periodically. Although this strategy could response the burst traffic in time, it also incurs a lot of control overheads. Furthermore, this strategy may transmit redundant control messages when the maximum value decreases.

2) Linear regression: As mentioned previously, the HAT would generate a lot of update messages when the maximum value suddenly increases since many nodes break their local filters. In other words, the sink node could send control messages only if the maximum value increases aggressively. Base on this observation, linear regression approach could be utilized to compute the trend of maximum values.

3) Cumulative Sum (CUSUM): CUSUM is a popular method used for change detection. Before change detection, the exponential weighted moving average could be applied to the historical maximum values to smooth short term fluctuations. This paper utilized CUSUM to detect if the smoothed values increase aggressively.

IV. PERFORMANCE EVALUATION

In this section, several simulation results will be presented to demonstrate the performance of proposed algorithm. The NS-2 network simulator [20] was used to simulate a wireless network environment. The temperature-data traces provided by the Intel Berkeley Lab Dataset [4] were used as the

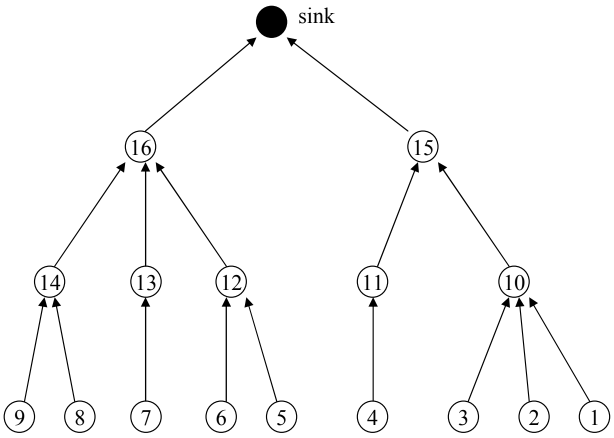


Figure 2. Network Topology.

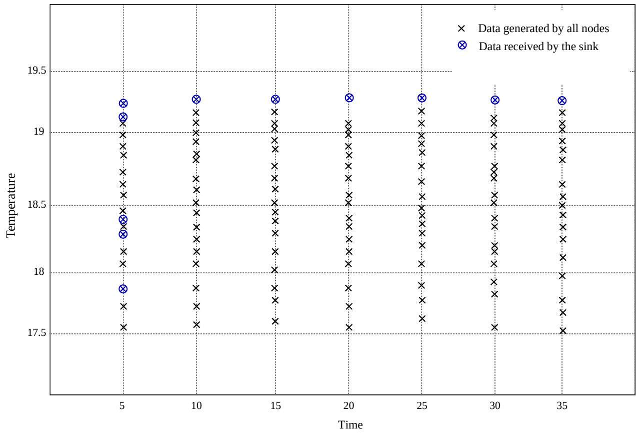


Figure 3. Performance of HAT.

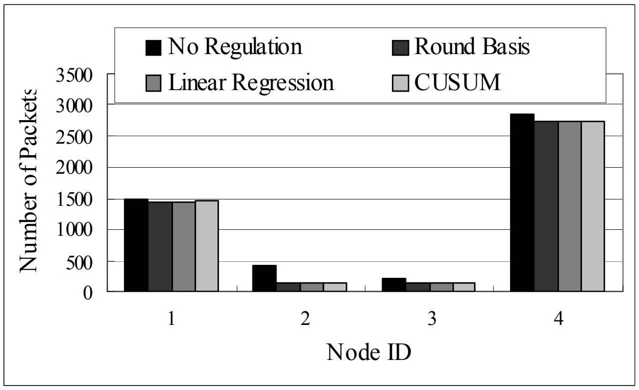


Figure 4. Number of network packets transmitted by nodes 1-4.

Table III
THE OCCURRENCE HISTOGRAM OF MAXIMUM VALUES

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Frequency	185	0	0	433	0	0	20	0	0	0	0	0	0	19	37	0

Table IV
ERROR RATE COMPARISON

Method	Round basis	Linear regression	CUSUM
Maximum error	0.102	0.102	0.06
Average error	0.04	0.05	0.04
Error rate	1.34 %	0.89 %	0.45 %

test data on which to run the proposed algorithm. The data traces consist of temperature readings, which were regularly collected from 54 nodes spread around the lab, as depicted in Fig. 1. However, there are some missing values for a few of nodes in the original data traces. Therefore, the missing values were replaced with the average value from the previous and subsequent readings. The simulated network topology was depicted in Fig. 2, which consists of the sink node and 16 sensor nodes. The 802.11 and UDP network protocols were used as the simulated MAC protocol and communication protocol, respectively. The sensor nodes were assumed to have a maximum transmission range of 25m.

In the simulations, every four seconds are called a round, and the maximum query is executed round-by-round, i.e., the maximum query is executed every four seconds. The sampling rate and reporting rate are set to two packets per second. Two data segments of Intel Berkeley Lab Data [4] were selected randomly as the input data in each simulation, and the simulator was executed 20 times to obtain the average results. Each data trace contains nearly 5500 temperature values (i.e., simulation time is about 2700 seconds). The length of Temporal Times Table is set to 10 to store ten historical values. The Global Times Table was set to contain all historical data except the data recorded in Temporal Times Table. The weight of the past data of the weighted exponential moving average is set to 0.2. If the importance of a node i is smaller than 0.2, the reporting rate of the node will be set to 2/5 packets per second as compared to the original reporting rate. If the importance of a node i is between 0.2 and 0.35, the reporting rate of the node will be set to 2/3 packets per second. Otherwise, the reporting rate keeps unchanged, i.e., two packets per second. The regulation packets will be sent once the classification of nodes has been changed since last transmission. The first 50 rounds are training stage that the initial regulation packets will be sent after the training phase. When the maximum node consistently replies their readings to the sink node, it will result in heavy loads on its parent node. We adopt a simple balance traffic mechanism which can share these traffics among all possible parent nodes of the maximum node.

Firstly, we demonstrate the basic function of HAT in Fig. 3. Hence, we set slow sampling rate and regulation rate (0.33 packet/second) with an attempt to illustrate the results clearly. In Fig. 3, we observed that at the first time unit there are only five packets received by the sink node instead of 16 packets. As mentioned previously, HAT is a double sided filter which can remove the values having no chance to compete for maximum. Thus, there are only five packets can break the thresholds along the path to the root. After the decision of maximum, the sink node sends a maximum designated packet to the node having maximum value. After receiving the maximum designated packet, the maximum node reports its readings if it has changed since last transmission. Thus, we can notice that the readings of maximum node are transmitted to the sink node continuously. Besides, after the first time unit, there are no other packets received by the sink node except the

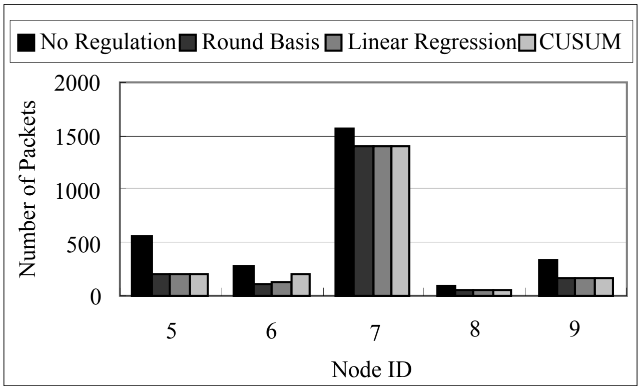


Figure 5. Number of network packets transmitted by nodes 5-9.

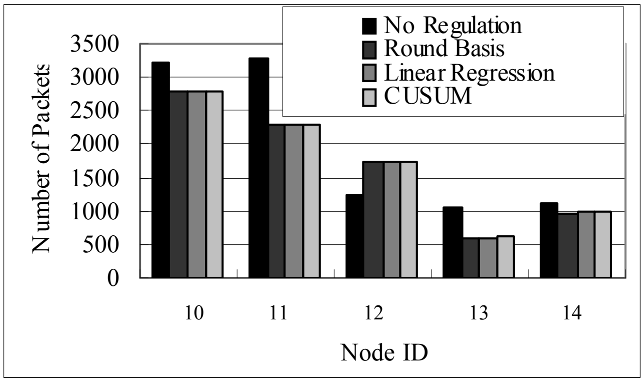


Figure 6. Number of network packets transmitted by nodes 10-14.

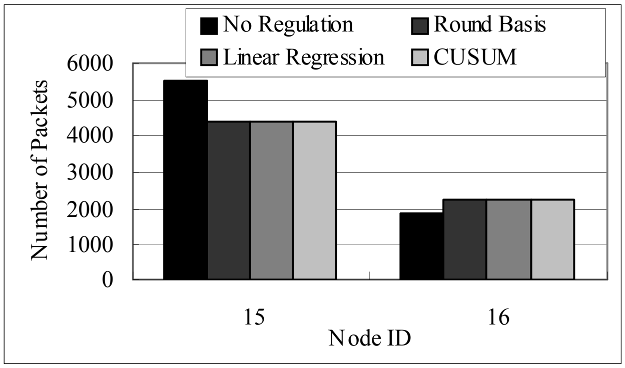


Figure 7. Number of network packets transmitted by node 15 and node 16.

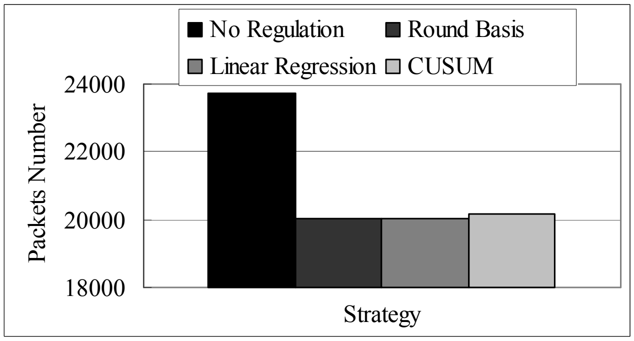


Figure 8. Total number of network packets.

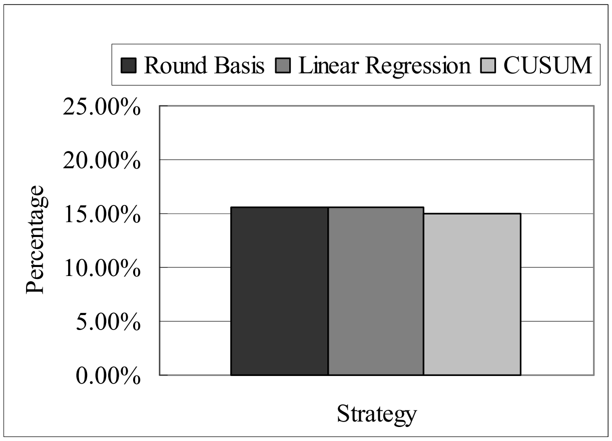


Figure 9. Network packet reduction.

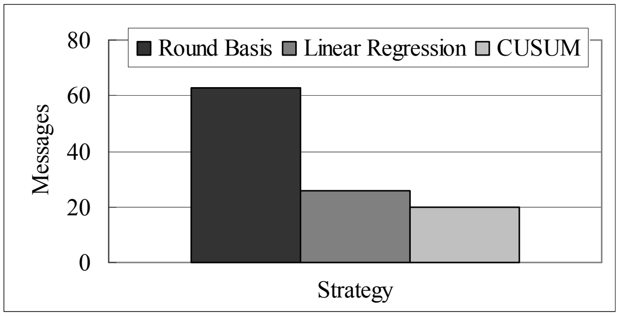


Figure 10. Control overhead.

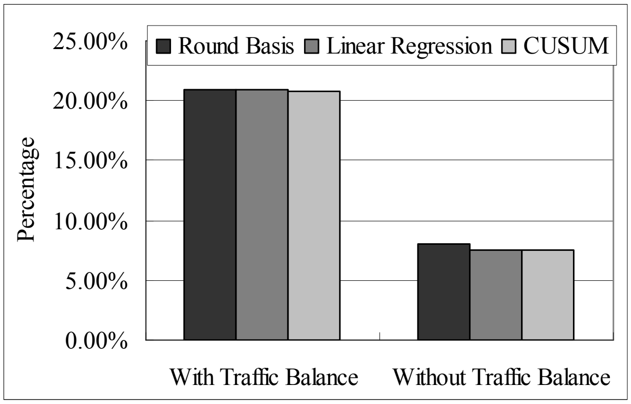


Figure 11. Packet reduction on node 15 and node 16.

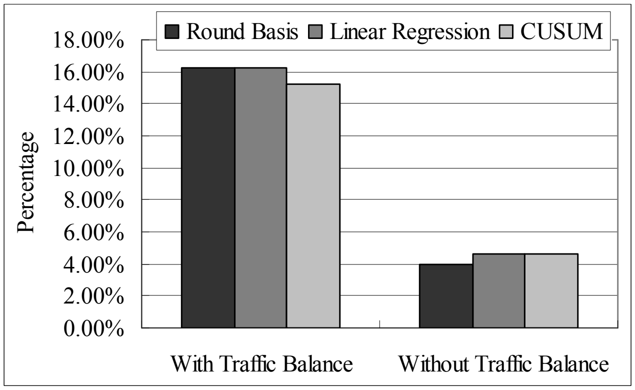


Figure 12. Packet reduction on nodes 10-14.

packets generated by the maximum node. The reason of this phenomenon is that the temperatures sensed by these nodes are decreasing. Hence, they cannot break their thresholds resulting in no other packets arrived at the sink node.

In the following, the proposed method with these three adjustment strategies of reporting rate is evaluated. In Fig. 4, it can be seen that the reduction on network traffic is nearly equal using the three adjustment strategies. This is because that these nodes become maximum node frequently, as shown in Table III. Hence, their reporting rates will be kept as normal reporting rate in order to reduce error rate. In Fig. 5, the three strategies obviously reduce the network traffic as compared with no regulation. This is because the sensed data of the nodes have little chance to become the maximum one. Accordingly, the reporting rate of the nodes will be reduced with an attempt to save energy.

In Fig. 6, the three adjustment strategies incur additional network traffic at node 12 as compared with no adjustment. The reason why our methods generated more packets at node 12 is that we adopt a traffic balance method which can share the traffic loads generated by the maximum node to all of its possible parents. For instance, when node 4 becomes the maximum node, it will transmit its data values continuously to all of its possible parents (i.e., nodes 10, 11, 12) instead of putting all loads on its direct parent (node 11). Thus, a lot of network packets could be avoided on the direct parent of the maximum node. The same phenomenon can be observed in 7. Fig. 8 shows the total number of reduced network packets. Fig. 9 suggests that the proposed strategies could obtain nearly 15% packet reduction. Among the three adjustment strategies, Round Basis has the highest percent of reduction. However, it also incurs the most overhead as well in Fig. 10.

The control overhead of Linear Regression method and CUSUM detection method is 41% and 36% as compared to Round Basis. After the illustration of packet reduction, we illustrate error rate, maximum error and average error due to packets reduction in Table IV. From Table IV, we can notice that the highest error rate is Round Basis. This is because that Round Basis reduces the most packets among the three regulation policies. Thus, it incurs the highest error rate. In addition to the error rate comparison, we further compare the method with traffic balance and without balance in Fig. 11 and Fig. 12.

In Fig. 11, node 15 and node 16 are two nodes one-hop away from the sink. In Fig. 12, nodes 10-14 are the nodes two-hop away from the sink. As can be seen, the reporting rate regulation combined with traffic balance scheme could achieve better balance than only reporting rate regulation methods. Furthermore, if we purely implement reporting rate regulation but not combined with traffic balance scheme, the extended life time of this network is limited. This is because that the nodes regulated to slow reporting rate are always the nodes with low probability to become maximum node. Hence, if we do not combine our method with a traffic balance scheme, the improvement of life time is limited.

V. CONCLUSIONS

The performance of the proposed algorithm was evaluated using NS-2 network simulator [20] and real-world data traces. The results indicate that the overall network packets are reduced to 80% with 1% data error in comparison with HAT. Besides, this paper also further keeps the error caused by reduced packets transmission below 1%.

According to our observation, a node will likely become the maximum node in the subsequent rounds once it is designed as the maximum node in the current round. This suggests that the maximum node would put heavy loading on its immediate parent due to continuous monitoring (reporting) of the maximum node. As a result, some nodes in the network would exhaust their power more quickly than the others. In this case, the balancing of communication tree should be taken into consideration in the proposed method in the future.

ACKNOWLEDGMENT

The research reported in this article was partially supported by NSC under Grant No. 100-2218-E-006-035 and by the Ministry of Economic Affairs in Taiwan under Grant No. B301AR2E10.

REFERENCES

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, ”A Survey on Sensor Networks,” IEEE communications magazine, vol. 40, pp. 102114, Aug 2002.

[2] C. Chong and S. P. Kumar, ”Sensor networks: Evolution, opportunities, and challenges,” in Proc. of IEEE, vol. 91, no. 8, pp. 12471256, Aug 2003.

[3] A. Silberstein, K. Munagala, and J. Yang, ”Energy-Efficient Monitoring of Extreme Values in Sensor Networks,” in Proc. ACM SIGMOD Intl Conf. on Management of Data, pp. 169-180, 2006.

[4] Intel Berkeley Research Lab. http://berkeley.intel-research.net/labdata/

[5] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, ”TAG: A tiny aggregation service for ad-hoc sensor networks,” in Proc. USENIX OSDI02, pp. 131146, Dec. 2002.

[6] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis, ”TiNA: A scheme for temporal coherency-aware in-network aggregation,” in Proc. ACM MobiDE03, Sep. 2003, pp. 6976.

[7] A. Deligiannakis, Y. Kotidis, N. Roussopoulos, ”Compressing Historical Information in Sensor Networks,” SIGMOD04, pp. 527-538, 2004.

[8] A. Deligiannakis, Y. Kotidis, N. Roussopoulos, ”Dissemination of compressed historical information in sensor networks,” VLDB J. vol. 16, no. 4, pp. 439-461, 2007.

[9] J. Considine, F. Li, G. Kollios, and J. Byers, ”Approximate aggregation techniques for sensor databases,” in Proc. IEEE ICDE04, pp. 449460, Mar. 2004.

[10] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson, ”Synopsis diffusion for robust aggregation in sensor networks,” in Proc. ACM SenSys04, pp. 250262, Nov. 2004.

[11] S. Yoon, C. Shahabi, ”The Clustered AGgregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks,” ACM Trans. On Sensor Networks, vol. 3, no. 1, article 3, Mar. 2007.

[12] P. Cao and Z. Wang, ”Efficient Top-k Query Calculation in Distributed Networks,” in Proc. ACM SIGACT-SIGOPS Symp. Principles on Distributed Computing (PODC 04), July 2004.

[13] R. Cheng, B. Kao, S. Prabhakar, A. Kwan, and Y.-C. Tu, ”Adaptive Stream Filters for Entity-Based Queries with Non-Value Tolerance,” in Proc. 31st Intl Conf. Very Large Data Bases (VLDB 05), 2005.

[14] R. Fagin, A. Lotem, and M. Naor, ”Optimal Aggregation Algorithms for Middleware,” in Proc. ACM Symp. Principles of Database Systems (PODS 01), Aug. 2001.

[15] U. Gu ntzer, W.-T. Balke, and W. Kie ling, ”Optimizing Multi-Feature Queries for Image Databases,” in Proc. Intl Conf. Very Large Data Bases (VLDB 00), 2000.

[16] B. Babcock and C. Olston, ”Distributed Top-k Monitoring,” in Proc. ACM SIGMOD Intl Conf. Management of Data (SIGMOD 03), pp. 28-39, June 2003.

[17] C. Olston, J. Jiang, and J. Widom, ”Adaptive Filters for Continuous Queries over Distributed Data Streams,” in Proc. ACM SIGMOD Intl Conf. Management of Data (SIGMOD 03), pp. 563- 574, June 2003.

[18] M. Wu, J. Xu, X. Tang, and W.-C. Lee, ”Top-k Monitoring in Wireless Sensor Networks,” IEEE Trans. on Knowledge and Data Engineering, vol. 19, no. 7, July 2007.

[19] A. Silberstein, R. Braynard, C. Ellis, K. Munagala, and J. Yang, ”A Sampling-Based Approach to Optimizing Top-k Queries in Sensor Networks,” in Proc. Intl Conf. Data Eng. (ICDE 06), Apr. 2006.

[20] The Network Simulator NS-2. http://www.isi.edu/nsnam/ns.

[21] S. Makridakis, S. C. Wheelwright, and R. J. Hyndman, Forecasting: Methods and Applications -3rd edition, John Wiley and Sons, Inc., 1998

[22] C. C. Zou, W. Gong, D. Towsley, and L. Gao, ”The Monitoring and Early Detection Internet Worms,” in IEEE/ACM Trans. on Networking, Oct. 2005.

[23] H. Wang, D. Zhang, and K.G. Shin, ”Change-Point Monitoring for Detection of DOS Attacks,” in IEEE Trans. on Dependable and Secure Computing, vol. 1, no. 4, 2004.

[24] Z. Chen, L. Gao, and K. Kwiat, ”Modeling the Spread of Active Worm,” in Proc. IEEE Intl Conf. on Computer Communications (INFOCOM), 2003.

[25] J. O. Kephart, and S. R. White, Measuring and Modeling Computer Virus prevalence, in Proc. IEEE Symp. on Security and Privacy, pp. 2-15, 1993.

[26] T. Bu, A. Chen, S. V. Wiel, and T. Woo, ”Design and Evaluation of a Fast and Robust Worm Detection Algorithm,” in Proc. IEEE Intl Conf. on Computer Communications (INFOCOM), 2006.

[27] C.T. Ee and R. Bajcsy, ”Congestion Control and Fairness for May-to-One Routing in Sensor Networks,” in Proc. Intl Conf. on Embedded Networked Sensor Systems (ACM SenSys), Nov. 2004.

[28] C. Wang, B. Li, K, Sohraby, M. Daneshmand, and Y. Hu, ”Upstream Congestion Control in Wireless Sensor Networks Through Cross-Layer Optimization,” IEEE journal on Selected Areas in Communications, vol. 25, no. 4, May 2007.

[29] B. Hull, K. Jamieson, and H. Balakrishnan, ”Mitigating Congestion in Wireless Sensor Networks,” in Proc. Intl Conf. on Embedded Networked Sensor Systems (ACM SenSys), 2004.

[30] K. Karenos, V. Kalogeraki, and S. V. Krishnamurthy, ”Cluster-based Congestion Control for Supporting Multiple Classes of Traffic in Sensor Networks,” in Proc. 2nd IEEE workshop on Embedded Networked Sensors, 2005.

[31] J. Gao, and L. Zhang, ”Load Balanced Shortest Path Routing in Wireless Networks,” in Proc. IEEE Intl Conf. on Computer Communications (INFOCOM), 2004.

Certificates Shared Verification Key Management for SurvSec Security Architecture

Mohamed Helmy Megahed
School of Information Technology and Engineering
University of Ottawa
Ottawa, Canada
mmega080@uottawa.ca

Dimitrios Makrakis
School of Information Technology and Engineering
University of Ottawa
Ottawa, Canada
dimitris@site.uottawa.ca

Hisham Dahshan
Communications Department
Military Technical College
Egyptian Armed Forces
Cairo, Egypt
hishamdahshan@yahoo.com

***Abstract*—SurvSec is a novel security architecture for reliable network recovery from base station BS failure of surveillance wireless sensor network (WSN) in hostile environment. Key management is the fundamental security mechanism in WSN which is needed for secure localization, secure clustering, secure data aggregation, secure authenticated broadcasting and secure routing. In this paper, a novel hybrid and dynamic key management scheme was proposed. This new scheme established secret keys between sensor nodes for SurvSec security architecture with high security level, high performance and low setup time. Hybrid key management provides high security level in the hostile environment however previous work assumed heterogeneous network utilizes high end sensor nodes (HSNs) with high power for high computations of certificates verification. This assumption provides attackers the best chance to destroy the network by targeting the HSNs. Also, HSN is connected to large number of nodes and there is no backup node for it. In addition, if the attackers target HSNs, then the connectivity and scalability will be affected where these nodes are points of failure. Moreover, previous work did not explain how to revoke compromised HSN. Furthermore, increasing the number of HSNs will increase the network deployment cost. Finally, if HSN is destroyed, nodes cannot have rekeying or addition of new nodes or revocation of compromised nodes. This paper proposed a hybrid scheme with homogenous network that uses some sensor nodes named as security managers (SMs) with a proposed novel mechanism called certificates shared verification to verify the certificates of group of nodes with distributed computations to overcome the absence of HSNs. This paper presents analytical evaluation and extensive simulation. The simulation results showed that at the cost of increasing communication overhead, the certificates shared verification mechanism was developed. Also, simulation results showed that the proposed scheme has lower computation overhead at SM side and lower setup time than HSN model. Both schemes have the same storage overhead.**

***Keywords-Key Management*; *Dynamic*; *Hybrid*; *Certificate Shared Verification*.**

I. INTRODUCTION

Researchers have investigated WSNs key management schemes and divided them into three categories. The category based on encryption has three classes which are: (i) symmetric key based key management [1-8]; (ii) asymmetric key based key management [9, 10] and (iii) hybrid key management [11-13]. The category based on location produced location based key management [14-16]. The category based on dynamicity and it has two classes which are static key management and dynamic key management [17, 18].

Hybrid key management combines the advantages of symmetric key and public key and it is the best solution for the hostile environment. Previous researches for hybrid key management [11-13] suggested using heterogeneous network with HSNs and low end sensor nodes (LSNs), where HSNs are used to perform high power calculations such as certificate verification, exponentiation, elliptic curve scalar multiplications and additions and modular multiplications.

HSNs are the best targets for the attackers to destroy the network where HSN is connected to large number of nodes. Also, HSN verifies certificates one by one within its connected nodes, which takes large time. Our scheme uses security managers to process a certificate shared verification process in distributed manner, with lower time for the same number of nodes, as shown in section 6. Moreover, HSN scheme does not provide backup for HSN where our scheme provides backup for SMs. Furthermore, no node can revoke HSN if it is compromised but in our scheme BKSM can revoke compromised SM. Besides, connectivity and scalability is affected by a compromised HSN, while our scheme provides BKSM to maintain high connectivity and scalability if SM is compromised. Destroying HSN in the middle of a branch results in cut communications in the branch. Also, each node underneath HSN needs three certificates verification for beacon nodes which is high cost for large number of nodes while our scheme assumes beacon nodes certificates verification once for the whole cluster. Finally, if HSN is destroyed, nodes cannot have rekeying or addition of new nodes or revocation of compromised nodes.

The proposed key management scheme has four types of nodes, which are SM, BKSM, initiator node and sensor nodes. The key management scheme assumes seven phases which are: key predistribution, key establishment, secure localization, secure clustering, rekeying, keys revocation and addition of new nodes. The protocol has four algorithms. The first algorithm is used for certificates verification and keys distribution. The second algorithm is used for initiator nodes to initiate key management process. The third algorithm is used for secure localization. The fourth algorithm is used for secure clustering. Sensor nodes near the BS are the first layer SMs. SMs are located every two layers. First, SMs near the BS verify the certificate of BS and the BS verifies the certificates of the first layer SMs then they share a symmetric link keys. Second, first layer SMs determine their locations from their neighbour beacon nodes after receiving the neighbour beacon nodes certificates and then send them to BS for verification. Third, SMs broadcast their certificates to their neighbour nodes underneath and these nodes verify the certificate of SMs. Fourth, neighbour nodes underneath SMs broadcast their certificates to SMs which in turn send these certificates to BS for verification then SMs and neighbour nodes underneath share a symmetric keys. Fifth, neighbour nodes underneath SMs determine their locations from their neighbour beacon nodes after receiving the neighbour beacon nodes certificates and then send them to SMs then to BS for verification. Sixth, SMs and their neighbour nodes underneath form secure clustering then SMs select BKSMs according to maximum connectivity between BKSM and sensor nodes in the cluster. Finally, lower layer SMs send certificates of their neighbour nodes underneath and beacon nodes to higher layer SMs for verification.

Our scheme proposed to deploy an initiator node every predefined number of nodes to start the process of key management in distributed manner and to finish it in controlled efficient time where these nodes are SMs. These nodes collect the certificates of their underneath nodes for verification and execute our proposed second algorithm. Finally, every initiator node communicates with its higher layer node and its upper layer SM.

In this paper, we proposed a new hybrid and dynamic key management in homogenous network that uses a novel idea of certificates shared verification to avoid using HSN and our scheme has BKSM for every cluster to replace the SM if it is compromised.

The proposed scheme provided secure clustering algorithm to choose backup security managers (BKSMs). In addition, the proposed scheme can revoke the compromised SM by the BKSM. Moreover, BKSM will maintain the network scalability and connectivity if the SM is compromised. Furthermore, the proposed scheme provided secure localization algorithm with certificates shared verification to lower computation overheads and to verify certificates of beacon nodes once for the whole cluster. The proposed dynamic key management uses certificates shared verification to reduce computations overheads and setup time for rekeying and addition of new nodes. The proposed scheme used initiator nodes every predefined number of nodes to start key management process for its underneath nodes to overcome absence of HSN. The proposed scheme can distribute link keys in time lower than HSN model.

A. Contributions

1- We designed a homogenous network that utilizes SMs, BKSMs and initiators to implement the distributed security concept instead of using HSNs which is the best target for the attackers.

2- We designed the certificates shared verification mechanism to distribute the high power computations of certificates verification among sensor nodes in the cluster.

3- We designed an integrated key management scheme that combines hybrid key management; and dynamic key management to resist attacks in the hostile environment.

4- We designed a secure localization algorithm that employs the certificates shared verification scheme with low computation overhead through verifying beacon nodes certificates only once for the cluster where previous scheme assumes that each sensor node verifies certificates of three beacon nodes.

5- We designed a secure clustering algorithm that chooses BKSM to replace and revoke the SM if it is compromised. Also, BKSM will maintain high connectivity and high scalability if SM is compromised.

6- We designed the network with low setup time, and low cost compared to network with HSNs. The computation overhead at SM is lower than that at HSN.

7- We designed our key management to be dynamic to provide rekeying, revocation of compromised sensor nodes and addition of new nodes using certificates shared verification.

B. Outline of the Paper

Section 2 presents the related work. Section 3 describes the network assumptions and threat model. Section 4 describes the proposed hybrid and dynamic key management scheme along with certificates shared verification. Section 5 presents security analysis of the proposed scheme. Section 6 presents the performance analysis. Section 7 presents simulation results. Section 8 presents comparison with previous works. Finally, Section 9 concludes the paper.

II. RELATED WORK

In this section, we present related work to our proposed scheme.

A. SurvSec Security Architecture

Surveillance Security (SurvSec) is a new designed security architecture for reliable network recovery from single BS failure of surveillance WSN with single BS [19]. SurvSec relies on a set of sensor nodes serve as SMs for management and storage of the security related data of all sensor nodes. SurvSec has three components: (1) Sensor nodes serve as SMs, (2) Data Storage System, (3) Data Recovery System.

SurvSec is used for securing surveillance WSN during the time between the BS failure and the new mobile BS deployment which is the perfect time for attackers to compromise many legitimate nodes then destroy the security of the whole network. Also, SurvSec describes how the new BS will verify the trustworthiness of the deployed WSN otherwise a new WSN must be deployed.

B. WSN Key Management Schemes

1. Static versus Dynamic Key Management

Static key management schemes assume that once administrative keys are predeployed in the nodes, they will not be changed. Most static schemes use the overlapping of administrative keys to determine the eligibility of neighbouring nodes to generate a direct pair-wise communication key.

Basically, dynamic key management schemes change administrative keys periodically, or on demand or on detection of node capture. The major challenge in dynamic keying is to design a secure yet efficient rekeying mechanism. A proposed solution to this problem is using exclusion-based systems (EBSs); a combinatorial formulation of the group key management problem developed in [17, 18]. A drawback of the basic EBS-based solution is that a small number of nodes may collude and collectively reveal all the network keys.

2. *Key Management based on Encryption Key*

Symmetric key based key management schemes are widely used because these schemes consume less computation time and power than other schemes, which are suitable for the limited resource characteristics. Based on the key distribution, key discovery and key establishment in the schemes, we can divide these schemes into eight categories: entity based key management schemes [1], pairwise key pre-distribution schemes [2], pure probabilistic-based schemes [3], polynomial-based key pre-distribution schemes [4], matrix- based key pre-distribution schemes [5, 25], tree-based key pre-distribution schemes [6], combinatorial design-based key pre-distribution schemes [7] and exclusion basis systems EBS-based key pre-distribution schemes [8].

Public key based key management schemes have many advantages such as low communications overhead, low storage overhead, high scalability. It can provide simpler solution with much stronger security strength. Public key based schemes have been categorized into three types: RSA-based asymmetric encryption system, ECC-based asymmetric encryption system and ID-based key agreement schemes. Several research groups have successfully implemented the public-key in WSNs [9, 10]. Asymmetric key based key management requires higher computations and energy cost than symmetric key based key management.

Hybrid key establishment schemes are proposed by several research groups [11-13]. The motivation is the needs for high security level and to exploit the difference among the BS, the HSNs and the sensors, and place the cryptographic burden on the BS or the HSNs. Sensors have limited computational power and energy resources. On the other hand, the BS and HSNs have much more computational power and other resources. Previous hybrid key establishment schemes reduce the high computational cost on the sensors by placing them on the HSN side and assume certificates verification for large number of nodes at HSN. Huang et al. [11] proposed a hybrid authenticated key establishment scheme, which is based on a combination of elliptic curve cryptography (ECC) and symmetric-key operations. The hybrid key establishment protocol reduces the high cost elliptic curve random point scalar multiplications at the sensor side and replaces them with low cost and efficient symmetric-key based operations.

3. *Key Management based on Location*

Recently researchers have suggested utilizing the location of sensor nodes [14- 16] after node deployment to improve the security of key management. Location based key management protocols are very efficient methods in terms of key connectivity and storage overhead. Location-aware key management is resilient against node capture attacks in large-scale sensor networks.

III. NETWORK MODEL&THREAT MODEL

A. Network Model

We consider a hierarchical WSN consisting of a BS, sensor nodes which are grouped into clusters and beacon sensors equipped with GPS called beacons. Each node has a unique ID, unique location and unique certificate. The assumptions of model are as follows:

1- We assume sensor nodes are static and some nodes continuously store the detected security threats and all other security data related to nodes where these nodes are SMs.

2- Nodes near the BS have the public key of the BS.

B. Threat Model

We consider an adversary that tries to uncover the keys of the network through capturing some nodes.

IV. PROPOSED SCHEME

The proposed scheme has seven phases which are key predistribution phase, key establishment phase, secure localization phase, secure clustering phase, key revocation phase, rekeying phase and add new node phase. The proposed scheme has four types of sensors: SMs, BKSMs initiators and sensor nodes.

A. Key Predistribution Phase

The key predistribution phase consists of acquiring the sensors certificate from the certificate authority CA. ECC is used in this protocol to perform security functions on sensors with limited computing resources. The protocol uses the elliptic curve explicit certificate scheme instead of X.509 because of the resulting low storage overhead, low communication overhead, which is a dominant factor for low bit transmission channels in WSN.

The certificate generation processes for any sensor node *U* is performed offline before it joins the network.

1- An elliptic curve E defined over *GF(p)* (*p* is the characteristic of the base field) with suitable coefficients and a base point *P* of large order *n* is selected and made public to all users.

2- CA selects a random integer q_{CA} as its static private key, and computes the public key $Q_{CA} = q_{CA} X P$.

3- To obtain a certificate and private-public key pair, the sensor U randomly selects a key pair (q_U, Q_U) where $Q_U = q_U X P$ and sends Q_U and q_U to CA.

4- CA verifies U's identity and private-public key pair.

5- The implicit certificate for U is the concatenation of CA's public key Q_{CA}, the device identity ID_U, the U public key Q_U and the certification expiration date t_U , i.e., the

ISBN: 978-1-61208-207-3

certificate is (Q_{CA}, ID_U, Q_U, t_U) signed by the CA private key using Elliptic Curve Digital Signature Algorithm ECDSA.

B. Key Establishment Phase

Certificates Verification &Keys Distribution

Power of the signature verification for ECDSA is about 1000 times more than the power of the signature transmission [20]. Each node in HSN model performs four times certificate verification for three beacon nodes and for HSN certificates. With the same number of certificates verification at each node, we developed our proposed certificates shared verification scheme. Each node in our scheme verifies four certificates only with the cost of increasing the communication overhead with four messages for every node. These verifications are: first verification for SM certificate, two verifications for two nodes underneath that node, and one verification for beacon node certificate. We assume that there are nodes named as security managers SMs and these nodes are located every two layers. We assume that there are nodes named as initiators every predefined number of nodes such as 30, 20 or 10 nodes to start the operation of key management process.

We explain our scheme in the form of two algorithms.

Algorithm 1: Certificates Verification and Keys Distribution

1: BS → n : {BS (Q_{CA}, ID_{BS}, Q_{BS}, t_{BS}) }

The BS broadcasts its certificate to nodes near BS at layer n and the nodes verify certificate of BS. These nodes are SMs.

2: n → BS : {n (Q_{CA}, ID_U, Q_U, t_U) }

The nodes near the BS at layer n broadcast their certificates to the BS and the BS verifies the certificates of these nodes.

3: n : selects (*k*), calculates (d_U), encrypts (d_U)

Each node near BS at layer n selects a *k*-bit random number c_U of 160 bits to produce its link key contribution with the BS.

Each node at n calculates the value of $d_U = H(c_U \parallel ID_U)$ where H is a cryptographic hash function. Each node at n encrypts d_U with BS public key Q_{BS}. To encrypt and send a message d_U to BS, each node at n chooses a random positive integer x and produces the ciphertext C_m consisting of the pair of points which are: $C_m = (x\, P,\ d_U + x\, Q_{BS})$.

4: n → BS : { C_m }

Each node near BS at layer n sends its encrypted link key contribution with the BS which is C_m.

5: BS : decrypts (C_m), selects (*k*), calculate (d_{BS}), encrypts (d_{BS})

BS decrypts C_m for every node at n. BS multiplies first point in the pair by BS's private key and subtracts result from second point: $d_U + x\, Q_V - q_V\,(x\, P) = d_U + x\,(q_V\, P) - q_V\,(x\, P) = d_U$.

BS selects a *k*-bit random number c_{BS} of 160 bits for each node near BS to produce its link key contribution with nodes near BS. BS calculates the value of $d_{BS} = H(c_{BS} \parallel ID_{BS})$ for every node near BS where H is a cryptographic hash function.

BS encrypts d_{BS} for every node near BS using symmetric key encryption under key d_U, generating value $y = E_{du}(\ ID_{BS} \parallel d_{BS})$.

6: BS → n : { *y* }, {hash {*K*}}

BS sends *y*, the encrypted link key contribution of BS, to every node near BS. BS generates the link key with every node near the BS at n by calculating $K = H\ (d_u \parallel ID_U \parallel d_{BS} \parallel ID_{BS})$ then $H(K)$ where H is a cryptographic hash function. BS sends $H(K)$ of every node at n to its participant to achieve correctness.

7: n : decrypts (*y*), calculates (*K*)

Every node at n decrypts the received message *y* using symmetric key encryption under key d_U to obtain the value d_{BS}.

Every node at n generates the link key with BS by calculating $K = \mathrm{H}(d_u \parallel ID_U \parallel d_{BS} \parallel ID_{BS})$.

8: n → BS : {*z*}

Every node at n calculates $z = \mathrm{H}(K)$ and sends *z* to BS. BS checks if $z = H(K)$. If yes, the link key is established correctly. Otherwise, the protocol is terminated.

9: n → n-1 : {n (Q_{CA}, ID_{SM}, Q_{SM}, t_{SM}) }

Each SM at layer n broadcasts its certificate to nodes at layer n-1 and nodes at n-1 verify the certificate of its SM. Each node at layer n-1 verifies SM certificate.

10: n-1 → n : {n-1 (Q_{CA}, ID_U, Q_U, t_U) }

Each node at layer n-1 sends its certificate to its SM at layer n

10: n → BS : all certificates {n-1 (Q_{CA}, ID_U, Q_U, t_U) }

Every SM at layer n sends the certificates of its nodes at layer n-1 to BS for verification because SM will lose high power and consume large time for verifying certificates of at least four nodes connected to it.

11: BS → n : {valid certificates or invalid certificates}

BS sends to each SM an encrypted message indicating that its certificates from layer n-1 are valid or not.

Then SMs at layer n executes steps from 3 to 8 to share symmetric link keys with nodes at layer n-1.

12: n-1 → n-2 : {n-1 (Q_{CA}, ID_U, Q_U, t_V) }

Every node at layer n-1 sends its certificate to its neighbour node at layer n-2 and the node at layer n-2 verifies the certificate of node at layer n-1. Nodes at layer n-2 are SMs.

13: n-2 → n-1, n : {n-2 (Q_{CA}, ID_{SM}, Q_{SM}, t_{SM}) }

Every node at layer n-2 sends its certificate to its connected node at layer n-1 then to the SM at layer n. The node at layer n-1 verifies the certificate of node at layer n-2 and node at layer n-2 verifies certificate of node at layer n-1.

14: n-2 → n, n-1 : {share link keys }

SM at layer n-2 executes steps from 3 to 8 to share symmetric link keys with node at layer n-1 and SM at layer n.

15: n-2 → n-3 : {n-2 (Q_{CA}, ID_{SM}, Q_{SM}, t_{SM}) }

Every node at layer n-2 which is a SM broadcasts its certificate to nodes at layer n-3 and nodes at n-3 verify the certificate of its SM.

16: n-3 → n-2 : {n-3 (Q_{CA}, ID_U, Q_U, t_U) }

Each node at layer n-3 sends its certificate to its connected SM at layer n-2.

17: n-2 → n : all certificates {n-3 (Q_{CA}, ID_U, Q_U, t_U) }

Every SM at layer n-2 sends the certificates of its nodes at layer n-3 to its SM at layer n for verification.

18: n → n-1 : all certificates {n-3 (Q_{CA}, ID_U, Q_U, t_U) }

SM at layer n sends the certificates of nodes at layer n-3 to its downstream nodes at layer n-1 for verification.

18: n-1 → n : {valid certificates or invalid certificates}

Every node at layer n-1 sends to its SM indicating that the checked certificate from layer n-3 is valid or not.

19: n → n-2 : {valid certificates or invalid certificates}

SM at layer n sends to the SM at layer n-2 indicating that the checked certificates from layer n-3 are valid or not. Then SMs at layer n-2 executes steps from 3 to 8 to share symmetric link keys with nodes at layer n-3. **Finally**, lower layer SMs send certificates of their neighbour nodes underneath to higher layer SMs for verification.

Discussion

The bottleneck of algorithm 1 is the number of SMs near the BS because if the number of these nodes increases, this will reduce setup time for the nodes underneath the SMs. Therefore, if the number of SMs near BS is more than three, SMs near BS execute algorithm 2.

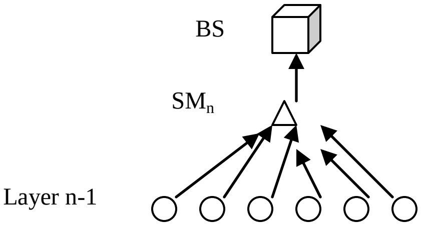


Fig. 1.a Certificates Verification for layer n-1

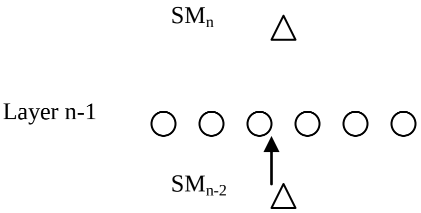


Fig. 1.b Certificates Verification for layer n-2

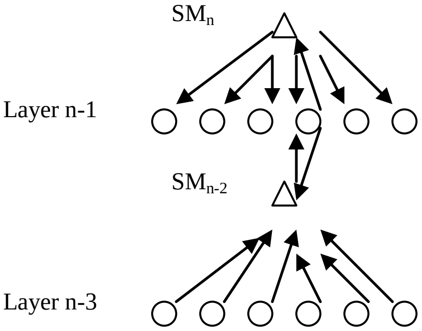


Fig. 1.c Certificates Verification for layer n-3

Fig. 1 shows certificates shared verification process in three layers using the first algorithm. Initiator nodes start the process of key management in distributed manner where these nodes are predetermined every number of nodes such as 30, 20 or 10 nodes. Initiator nodes work as HSN to control the setup time for the key management.

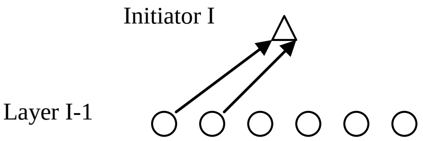


Fig. 2.a Certificates Verification using Initiator for 2 nodes

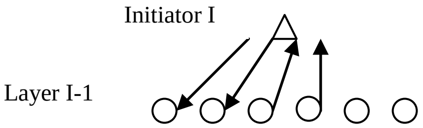


Fig. 2.b Certificates Verification using Initiator for 2 nodes

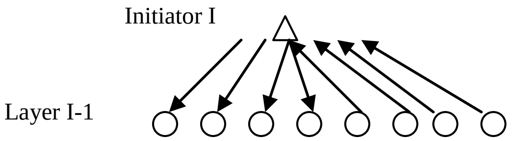


Fig. 2.c Certificates Verification using Initiator for 4 nodes

Fig. 2 shows certificates shared verification process for one layer using algorithm 2. SM verifies certificates of first two nodes then it sends the certificates of the second two nodes to the first two nodes then it sends certificates of other four nodes to the verified four nodes. Algorithm 2 is efficient in terms of distribution of power consumption among sensor nodes in the cluster and it can be used with all SMs in their clusters. Algorithm 1 provides high speed for certificates verification but its drawback is that the cluster nodes between an initiator and its upper layer SM are not involved in the process of certificates verification. Therefore, there is a trade off between high speed certificates verification using algorithm 1 and distributed power consumption using algorithm 2.

Algorithm 2: Initiator nodes to start key management process

1: I → n : { I (Q_{CA}, ID_{SM}, Q_{SM}, t_{SM}) }

Each initiator node broadcasts its certificate to its underneath nodes at layer n to verify it. The nodes at layer n verify the certificate of the initiator.

2: n → I : { n (Q_{CA}, ID_U, Q_U, t_U) }

The initiator node receives the certificates of its underneath nodes for verification. We assume there are n nodes underneath the initiator node. First, the initiator node verifies the certificate of the first two nodes.

3: I → $n_{1,2}$: { share link keys }

The initiator node shares link keys with node 1 and node 2 as steps from 3 to 8 in algorithm 1.

4: I → $n_{1,2}$: { $n_{3,4}$ (Q_{CA}, ID_U, Q_U, t_U) }

The initiator node sends to node 1 and node 2 underneath the certificates of node 3 and node 4 for verification.

5: $n_{1,2}$ → I : { valid certificates or invalid certificates }

Node 1 and node 2 send to the initiator node two messages indicating that certificates of nodes 3 and 4 are valid or not.

6: I → $n_{3,4}$: { share link keys }

The initiator node shares link keys with node 3 and node 4 as steps from 3 to 8 in algorithm 1.

7: I → $n_{1,2,3,4}$: { $n_{5,6,7,8}$ (Q_{CA}, ID_U, Q_U, t_U) }

The initiator node sends to node 1, node 2, node 3 and node 4 underneath the certificates of node 5, node 6, node 7 and node 8 for verification and nodes 1, 2, 3, 4 respond with valid certificate or not.

8: I → $n_{5,6,7,8}$: { share link keys }

The initiator node shares link keys with node 5, node 6, node 7 and node 8 as steps from 3 to 8 in algorithm 1. **Finally**, the process of the initiator continues to verify all of its underneath nodes then its underneath nodes use algorithm 1 to share link keys with their underneath nodes and so on.

1. Certificates shared verification between SM near BS and BS needs two messages but it needs four messages between SM at lower layer and SM at upper layer.
2. Each SM establishes a link key with its nodes underneath in ten messages but SM near BS establishes a link key with its nodes underneath in eight messages.
3. After the SMs and the sensor nodes establish link keys, they determine their locations using our proposed secure localization scheme with certificates shared verification.

C. Secure Localization Phase

A number of secure localization algorithms [21] have been reported. Different researchers have different strategies to categorize them. These strategies can be divided into direct and indirect localization, centralized localization and distributed localization, range-based localization and range-free localization, absolute localization and relative localization. We propose to get the location information from the followings approach:

The indirect approaches of localization were introduced to overcome some of the drawbacks of the GPS-based direct localization techniques while retaining some of its advantages. In this approach, a small subset of nodes in the network, called the beacon nodes, are equipped with GPS receivers to compute their location. Beacon nodes send beams of signals providing their location to all nodes in their vicinity. Using the transmitted signal containing location information, nodes compute their location. Each node needs three beacon nodes to locate its position.

Our proposed scheme depends on SM and certificates shared verification for secure localization. We assume that each cluster has three beacon nodes. Sensor nodes in the cluster send the beacon nodes certificates to SM then SM sends these certificates to its upper layer SM for verification to insure one verification time for beacon nodes certificates for the whole cluster. The upper layer SM sends these certificates to its underneath nodes for verification. Verification power is 1000 times more than communication power.

Algorithm 3: Secure Localization

1: $Beacons_{1,2,3}$ → SM_n : {$Beacons_{1,2,3}$ (Q_{CA}, ID_B, Q_B, t_B) }

The beacon nodes near BS broadcast their certificates and locations to SMs near BS. We need three beacon nodes to locate the position.

2: SM_n → BS : { $Beacons_{1,2,3}$ (Q_{CA}, ID_B, Q_B, t_B) }

The SMs near BS at layer n send the certificates of the beacon nodes to BS for verification.

3: BS → SM_n : {valid certificates of $Beacons_{1,2,3}$ }

BS sends to SMs at layer n that beacon nodes certificates are valid.

4: SM_n → $Beacons_{1,2,3}$: { $Key_{1,2,3}$ }

Every SM at layer n shares a link key with the three beacon nodes in four steps.

5: SM_n : calculates (x, y) position

Every SM at layer n calculates its position.

6: $Beacons_{1,2,3}$ → n-1 : {$Beacons_{1,2,3}$ (Q_{CA}, ID_B, Q_B, t_B) }

The beacon nodes near BS broadcast their certificates and locations to nodes at layer n-1.

7: n-1 → SM_n : { $Beacons_{1,2,3}$ (Q_{CA}, ID_B, Q_B, t_B) }

The nodes at layer n-1send the certificates of beacon nodes to SMs at layer n for verification. If the beacon nodes certificates are previously verified, it is ok but if there are new beacon nodes certificates, then SMs at layer n send the new beacon nodes certificate to BS for verification.

8: SM_n → n-1 : { $\mathrm{Key}_{1,2,3}$ }

Every SM at layer n sends its link keys with the beacon nodes to its connected nodes at layer n-1.

9: n-1 : calculates (x, y) position

Every node at layer n-1 calculates its position.

10: $\mathrm{Beacons}_{4,5,6}$ → SM_{n-2} :{$\mathrm{Beacons}_{4,5,6}$ (Q_{CA}, ID_B, Q_B, t_B) }

The beacon nodes near SMs at layer n-2 broadcast their certificates and locations to SMs at layer n-2.

11: SM_{n-2} → SM_n : { $\mathrm{Beacons}_{4,5,6}$ (Q_{CA}, ID_B, Q_B, t_B) }

The SMs at layer n-2 send the certificates of the beacon nodes to SMs at layer n for verification.

12: SM_n → n-1 : { $\mathrm{Beacons}_{4,5,6}$ (Q_{CA}, ID_B, Q_B, t_B) }

The SMs at layer n send the certificates of the beacon nodes to nodes at layer n-1 for verification.

13: n-1 → SM_n : { valid certificates of $\mathrm{Beacons}_{4,5,6}$ }

The nodes at layer n-1 send to SMs at layer n that beacon nodes certificates are valid.

14: SM_n → SM_{n-2} : { valid certificates of $\mathrm{Beacons}_{4,5,6}$ }

The SMs at layer n send to SMs at layer n-2 that beacon nodes certificates are valid.

15: SM_{n-2} → $\mathrm{Beacons}_{4,5,6}$: { $\mathrm{Key}_{4,5,6}$ }

Every SM at layer n-2 shares a link key with the three beacon nodes in four steps.

16: SM_{n-2} : calculates (x, y) position

Every SM at layer n-2 calculates its position.

17: $\mathrm{Beacons}_{4,5,6}$ → n-3 : {$\mathrm{Beacons}_{4,5,6}$ (Q_{CA}, ID_B, Q_B, t_B) }

The beacon nodes near nodes at layer n-3 broadcast their certificates and locations to nodes at layer n-3.

18: n-3 → SM_{n-2} : { $\mathrm{Beacons}_{4,5,6}$ (Q_{CA}, ID_B, Q_B, t_B) }

The nodes at layer n-3 send the certificates of beacon nodes to SMs at layer n-2 for verification. If the beacon nodes certificates are previously verified, it is ok but if there are new beacon nodes certificates, then SMs at layer n-2 send the new beacon nodes certificate to SMs at layer n for verification.

19: SM_{n-2} → n-3 : { $\mathrm{Key}_{4,5,6}$ }

Every SM at layer n-2 sends its link keys with the beacon nodes to its connected nodes at layer n-3.

20: n-3 : calculates (x, y) position

Every node at layer n-3 calculates its position. **Finally**, lower layer SMs send certificates of beacon nodes to higher layer SMs for verification.

1. Certificates shared verification for beacon nodes certificates between SM at lower layer and SM at higher layer will reduce setup time and reduce computations complexity at the cost of increasing only four messages.
2. Certificates verification for beacon nodes is done only one time at the SM not multiple times at each node underneath the SM to reduce computations complexity.
3. Sensor nodes underneath SM will use the shared keys between the SM and the beacon nodes which will reduce the setup time, computations and storage overhead.
4. After the SMs and the sensor nodes determine their locations, they form secure clustering.

D. Secure Clustering Phase

SMs can form secure clustering [22] with their nodes underneath and SM can choose BKSM to replace it if the SM is compromised.

Algorithm 4: Secure Clustering

1: BS → n : {req SM_msg }

BS sends to nodes near BS at layer n that these nodes are SMs using its shared symmetric key with these nodes.

2: SM_n → n-1 : { adv cluster_msg }

Every SM at layer n sends an encrypted advertise message to nodes at layer n-1 to form a cluster.

3: n-1 → SM_n : { join cluster_msg }

Every node at layer n-1 sends an encrypted message to its SM at layer n to join the cluster.

4: SM_n → n-1 : {choose BKSM }

The SM at layer n chooses BKSM according to maximum connectivity between the BKSM and the nodes in the cluster where BKSM must be connected to all nodes in the cluster.

5: BKSM_n → n-1 : { BKSM (Q_{CA}, ID_{BKSM}, Q_{BKSM}, t_{BKSM}) }

The BKSM sends its certificate to the nodes at layer n-1 where SM at layer n verifies this certificate. Also, the BKSM sends its certificate to its upper layer node to establish a link key with it to reroute data if SM is compromised.

6: n-1 → n-2 : { req SM_msg }

The nodes at layer n-1 send to nodes at layer n-2 an encrypted message that these nodes are SMs.

7: SM_{n-2} → n-3 : { adv cluster_msg }

Every SM at layer n-2 sends an encrypted advertise message to nodes at layer n-3 to form a cluster.

8: n-3 → SM_{n-2} : { join cluster_msg }

Every node at layer n-3 sends an encrypted message to its SM at layer n-2 to join the cluster.

9: SM_{n-2} → n-3 : {choose BKSM }

The SM at layer n-2 chooses BKSM according to maximum connectivity between the BKSM and the nodes in the cluster where BKSM must be connected to all nodes in the cluster.

10: BKSM_{n-2} → n-3 : { BKSM (Q_{CA}, ID_{BKSM}, Q_{BKSM}, t_{BKSM}) }

The BKSM sends its certificate to the nodes at layer n-3 where SM at layer n-2 verifies this certificate. Also, the BKSM sends its certificate to its upper layer node to establish a link key with it to reroute data if SM is compromised. **Finally**, the steps of forming the secure clustering are performed until the last layer of SM.

1. Our proposed secure clustering scheme assumes hybrid key management protocol to achieve high security level.
2. Our proposed scheme chooses BKSM to solve the problem of compromised SM and to sign the message of revoked SM.
3. Our scheme achieves secure clustering in four messages.

E. Key Revocation Phase

The first component of our dynamic based key management scheme is the keys revocation of the compromised sensor nodes. SurvSec security architecture has a compromised nodes detection algorithm at the first stage to be able to detect compromised nodes but it is not discussed in this paper.

When a sensor node is compromised by an adversary, all the session keys used by this sensor node will be revoked. The SM will broadcast a revocation message containing the identification of the compromised node to all the nodes underneath. A digital signature is computed over the message by utilizing Elliptic Curve Digital Signature Algorithm ECDSA at [23] with SMs private key. When a node receives the revocation message, it checks the message by verifying the digital signature. This prevents an adversary from sending a fake revocation message. If SM is compromised, it is revoked by the BKSM.

F. Rekeying Phase

The second component of our dynamic based key management scheme is rekeying after compromised nodes detection or rekeying can be done periodically. Rekeying is used when the SM is compromised. The BKSM will share a link key with its upper layer SM then the BKSM will use our novel scheme of certificates shared verification with its upper layer SM to verify the certificates of the cluster nodes. Finally, BKSM will share link keys with its lower SM and its nodes in the cluster.

G. Add New Node Phase

When a new node joins the network, it tries to find its nearest SM by broadcasting a Hello message contains the new node certificate.

To support the addition of new nodes, the SM verifies the certificate of the new nodes using our novel scheme of certificates shared verification.

V. SECURITY ANALYSIS

Security analysis of our protocol focuses on resilience to node compromise attack, and collusion attack.

A. Compromised Node Attack

1- If an attacker compromises one ordinary node, therefore, the number of insecure link is $P_{insec} = 1 / N$ where N is the number of nodes at the network. For n compromised ordinary nodes, number of insecure links is $P_{insec} = n / N$.

2- If the attacker compromises one SM, therefore, the number of insecure links is $P_{insec} = (n_s + 3) / N$ where n_s is the number of nodes in the cluster of the SM and 3 represents the links with upper SM, lower SM and SM upper node. For n compromised SMs, the number of insecure links is $\mathrm{P_{insec}} = \mathrm{n\ (n_s + 3) / N}$.

3- Suppose that in a network of N nodes, there are m SMs and BKSMs. The probability to compromise one SM or one BKSM is P(com) = 2m / N, so the probability of at least *k* nodes from the SMs and BKSMs are captured is:

$$p(k) = 1 - \sum_{i=0}^{k}\binom{n}{i}p(com)^{i}\,(1-p(com))^{n-i} \qquad (1)$$

The probability that all SMs and BKSMs are captured is:

$$p(k) = 1 - \sum_{i=0}^{2m}\binom{n}{i}p(com)^{i}\,(1-p(com))^{n-i} \qquad (2)$$

4- Our proposed key management assumes compromised node detection at the first stage and compromised nodes revocation. Therefore, SM will revoke the ordinary compromised node and the BKSM will revoke the SM to eliminate the insecure links.

B. Collusion Attack

Two nodes can collude when they share their keys with each other. Our designed protocol is resistant to collusion attack because each sensor node communicates only with a SM therefore; compromised nodes cannot discover themselves.

VI. PERFORMANCE ANALYSIS

The performance analysis is measured in Computation complexity, communication complexity and storage complexity. We assume that the network is secure during setup time which depends on number of initiators.

A. Computation Complexity

Our proposed hybrid key management scheme using certificates shared verification has much lower computations overhead at SM side rather than computations at HSN in heterogeneous network. For algorithm 1, our scheme assumes each sensor node in each cluster verifies four certificates for the keys distribution and localization which are the certificate of its SM, two certificate from its underneath nodes and one beacon node certificate. SM verifies one certificate which is its upper node. For algorithm 2, our scheme assumes each sensor node in each cluster verifies at most four certificates for the keys distribution and localization which are the certificate of its initiator, two certificates from the nodes of its cluster and one beacon node certificate. Initiator node verifies three certificates which are two certificates from its underneath nodes and one certificate for its upper node.

Each sensor node and SM performs three times hash to generate one link key. The sensor node encrypts its part of the link key with the SM's public key using ECC 160 bits scalar multiplication and addition. Also, the SM decrypts the received message from the sensor node with its private key. The SM encrypts its part of the link key using symmetric key under the key from the sensor node. The sensor node decrypts the message from the SM using symmetric key. Our scheme has less computation overhead at SM than the scheme uses HSNs at HSN.

In our scheme:

Each node performs at most 4 verifications and shares key with SM or initiator for keys distribution and localization.

SM or initiator performs at most 3 verifications and shares keys with n nodes for keys distribution and localization where n nodes are ranged from 4 to 8 nodes in the cluster.

In HSN scheme:

Each node performs 4 verifications and shares key with HSN for keys distribution and localization.

HSN performs n verifications and shares keys with n nodes where n nodes are ranged from 10 to 30 nodes underneath HSN.

Our scheme has lower computations than HSN scheme.

B. Communication Complexity

Communication complexity is the number and size of packets sent and received by a sensor node. In our protocol, the number of messages sent and received to establish a key between one sensor node and a SM is ten messages and we need six messages to establish link key between lower layer SM and upper layer SM. Device ID is 64 bits, expiration time is 64 bits, random number is 160 bits and L the sensor location is 64 bits. The certificate is 56 bytes from 20 bytes CA public key, 8 bytes node ID, 20 bytes node public key and 8 bytes validity time. Our scheme has higher communication overhead than HSN model with 4 messages to establish link key for every node.

In our scheme:

For algorithm 1:

Communication overhead = 6 N_{SM} + 10 m N_{SM},, N_{SM} is number of SMs and m is the number of nodes underneath SM within its cluster.

For algorithm 2:

Communication overhead = I (12 + 8 (m – 2)) + 6 I, I is the number of initiator nodes, m is the number of nodes underneath initiator. We have 2 nodes needs 12 messages and other nodes in the cluster need 8 messages and 6 represents the communication between the initiator and its upper node.

For algorithm 1 and 2: Total communication overhead is C_{com}.

$C_{com} = N_{SM} (6 +10 m) + I (2 + 8 m)$.

We found that communication overhead for algorithm 2 is lower than communication overhead for algorithm 1.

In HSN scheme:

For one HSN every 30 nodes: communication overhead is C_{com}.

$C_{com} = N_{HSN} (6 + 6 n_0 +8 n_1 + 10 n_2 + 12 n_3)$. Where N_{HSN} is the number of HSNs, n_0 is the number of first layer nodes underneath HSN, n_1 is the number of second layer nodes underneath HSN, n_2 is the number of third layer nodes underneath HSN, n_4 is the number of fourth layer nodes underneath HSN and 6 represents the communication between the HSN and its upper node.

For one HSN every 20 nodes: communication overhead is C_{com}.

$C_{com} = N_{HSN} (6 + 6 n_0 +8 n_1 + 10 n_2)$.

For one HSN every 10 nodes: communication overhead is C_{com}.

$C_{com} = N_{HSN} (6 + 6 n_0)$.

Our model has lower communication overhead than HSN model for one HSN every 30 but our model has higher communication overhead than HSN model for one HSN every 20 or 10 nodes.

C. Storage Complexity

Storage complexity is the amount of memory units required to store security credentials. Each sensor node stores its public key, private key, BKSM public key and the link key shared with the SM. The SM stores all of the shared keys with each sensor node underneath plus its public, private key, link key with upper SM, link key with the lower SM and link key with its upper node. Our scheme has the the same storage overhead as HSN scheme.

In our scheme:

Total SMs storage overhead = (N_S+5) N_{SM}, N_S is the number of nodes underneath SM and N_{SM} is the number of security managers.

Sensor nodes storage overhead = 3 N_S.

In HSN scheme:

Total HSNs storage overhead = (N_S+5) N_{HSN}.

Sensor nodes storage overhead = 3 N_S.

D. Setup Time

We assume that verification using ECDSA takes about 4 sec [24], share link key takes about 1 sec [11] and certificate transmission takes about 0.1 sec [11]. The transmission time is dominant factor but on the other hand, the bottleneck will be the certificate verification operation time. Setup time is equal to verification time plus communication time plus share link key time.

In our scheme:

For algorithm 1 the setup time is T.

T = 4S + n X 1S + (5 n + 2) X 0.1S, verification is done in parallel where upper layer SM sends to its underneath nodes the certificates of the nodes underneath its lower layer SM which is n nodes. Therefore, we need one verification time and n times to share link keys and (5n+2) messages to send all certificates to the verifiers and have the result.

For algorithm 2 the setup time is T.

T = m X 4S + n X 1S + (12 + 8 (n - 2)) X 0.1S, verification is done m times, share link keys is done n times and we need number of messages equal to (12 + 8 (n-2)).

Setup time for algorithm 1 is lower than setup time for algorithm2.

In HSN scheme:

Setup time = n X 4S + n X 1S + 6n X 0.1S, where n is the number of nodes underneath the HSN and 6n is the number of messages between nodes and HSN.

Our proposed scheme with algorithm 1 has much lower setup time than HSN model where we perform parallel verification but HSN model performs sequential verification.

Our proposed scheme with algorithm 2 has lower setup time than HSN model where we perform parallel verifications but HSN model performs sequential verification. Our proposed model combines both algorithm 1 and algorithm 2.

E. Scalability

In our scheme:

BKSM will replace the SM if it is compromised and this insures high scalability to extend the network.

In HSN scheme:

If a HSN is compromised in a branch, the scalability of the branch cannot be achieved because there is no backup HSN.

F. Connectivity

In our scheme:

BKSM will replace the SM if it is compromised and this insures high connectivity with its underneath nodes.

In HSN scheme:

If a HSN is compromised in a branch, the connectivity for the nodes underneath the HSN cannot be achieved because there is no backup HSN.

VII. SIMULATION RESULTS

In this section, we evaluate the communication overhead, the computations overhead and the network setup time under different number of nodes *N* for our proposed model and HSN model.

We built our proposed model and HSN model and we implemented a simulator in MATLAB that can scale to thousand of nodes. In this simulator, sensors can send and receive data from each other's. The simulation verifies the correctness and the feasibility of our security architecture. It is our future work to implement SurvSec in some sensor network testbeds with all its ingredients. Our simulation scenarios include *N* nodes distributed randomly. We choose *N* 1000, 2000 and 3000 sensor nodes.

In the simulations, these parameters are given as follows:

1- The number of sensor nodes *N* is varied from 1000, 2000 and 3000 sensor nodes.
2- The simulation is done for HSN or initiators every 30 nodes, 20 nodes and 10 nodes.

The communication overhead for security manager to exchange a key with a node is according to algorithm 1 or algorithm 2 or both as shown in section 6.

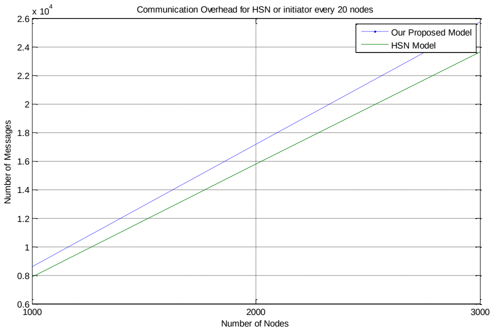


Fig. 3 Communication overhead every HSN or Initiator every 10 nodes

Fig. 3 shows the communication overhead for HSN model and our proposed model for one HSN node every 10 nodes and one initiator every 10 nodes. Our proposed model has higher communication overhead than HSN model with 20%.

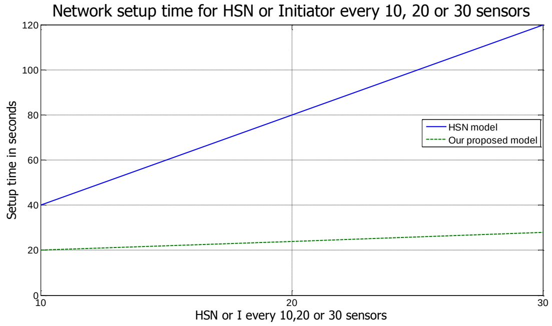


Fig. 4 Network Time Setup for HSN or Initiator every 30 nodes, 20 nodes, and 10 nodes

Fig. 4 shows the network setup time for HSN model and our proposed model for one HSN node or one initiator every 30 nodes, 20 nodes and 10 nodes. Our proposed model has at least half the network setup time than HSN model.

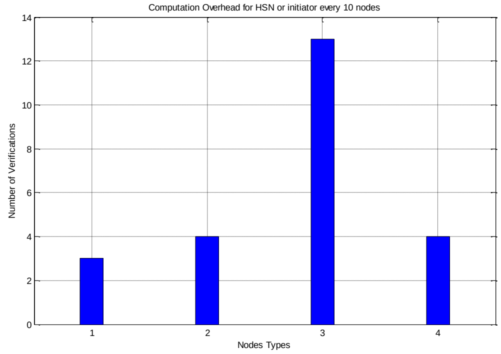


Fig. 5 Computation Overhead of Certificates Verifications for HSN or Initiator every 10 nodes

Fig. 5 shows the computation overhead for certificates verifications for HSN model and our proposed model for one HSN node or one initiator every 10 nodes. Number 1 at x-axis is the number of certificates verification at the SM which is 3 verifications for key establishment and secure localization. Number 2 at x-axis is the number of certificates verification at every node in our proposed model which is 4 verifications for key establishment and secure localization. Number 3 at x-axis is the number of certificates verification at the HSN which is 13 verifications for key establishment and secure localization. Number 4 at x-axis is the number of certificates verification at every node in HSN model which is 4 verifications for key establishment and secure localization. Our proposed model has lower computation overhead than HSN model. Our scheme has one quarter lower certificates verifications overhead then HSN model at SM side and one half lower certificates verification overhead in total. Finally, for HSN or Initiators every 10 nodes we increase communication overhead by 20% and we decrease the computation overhead to one half where power of certificates verification using ECDSA is 1000 times more than power of communication.

VIII. COMPARISON with others WORKS

Now, we compare between our proposed model and HSN model.

TABLE 1, COMPARISON between OUR MODEL and HSN MODEL.

	Property	**HSN Model [11-13]**	**Our Model**
1	Computation overhead for key establishment and secure localization	N verification at HSN and 4 verifications at node	3 verifications at SM and 4 verifications at node
2	Storage overhead	3 keys at node (n+5) at HSN	3 keys at node (n+5) at SM
3	Communication overhead for key establishment	6 or 8 or 10 or 12 messages for each node according to HSN every 30 or 20 or 10 nodes	8 messages for algorithm 2 or 10 messages for algorithm 1 for each node
4	Communication overhead for secure localization	No	3 messages from each node to SM and one verification message from SM to each node plus 6 messages for one time verification
5	Computation overhead for secure localization	3n verifications for the cluster	3 verifications for the whole cluster
6	Setup time	n verifications time	parallel verifications executes in 1/n time of HSN model for algorithm1 and n/2 time of HSN model for algorithm 2
7	Scalability	Affected by compromised HSN	High
8	Connectivity	Affected by compromised HSN	High
9	Backup node	No	BKSM
10	Secure localization	High cost at each node for 3 verifications	Low cost for 3 verifications for the whole cluster
11	Rekeying	High cost at HSN	Low cost at SM
12	Addition of new nodes	High cost at HSN	Low cost at SM
13	Probability of insecure links	High with compromised HSN	Low after compromised SM revocation
14	Effect of compromised nodes	No	Affect certificates shared verification
15	Nodes revocation	Cannot revoke HSN	BKSM revokes SM
16	Cost	High	Low

Our proposed scheme distributes certificate verification at nodes underneath SM rather than verifies certificates at SM. Also, our scheme verifies beacon nodes certificates once for the whole cluster. Our scheme has higher connectivity and scalability than HSN model. Our scheme can revoke compromised SM through BKSM and has lower network cost than HSN scheme. Our scheme has lower network setup time than HSN scheme and it has same storage overhead. Our scheme has lower computations overhead than HSN scheme.

IX. CONCLUSION

In this paper, we proposed a novel hybrid and dynamic key management scheme for WSNs which utilizes a novel scheme of certificates shared verification to verify the certificates of nodes in distributed computations and eliminate the usage of high end sensor nodes which are the best targets for the attackers. Our scheme is based on some sensor nodes called security managers which are chosen every two layers. We proposed a secure localization scheme with low computation overhead. Also, we proposed a secure clustering algorithm to choose backup security manager for every cluster to replace and revoke the security managers if it is compromised. Our proposed scheme can distribute link keys in lower setup time than the model uses high end sensor nodes. Our proposed scheme has higher communication overhead, lower computation overhead at the security manager node and lower energy cost at the security manager node than scheme uses high end sensor node. Both schemes have the same storage overhead. Our proposed scheme has low cost than model used high end sensor nodes. Our proposed scheme connectivity and scalability are not affected if the security manager is compromised.

REFERENCES

[1] Perrig A., Szewczyk R., Wen V., Cullar D., and Tygar J.D. “SPINS: security protocols for sensor networks”. In: Proceedings of the 7th annual ACM/IEEE international conference on mobile computing and networking, July 2001, pp. 189–199.

[2] Chan H., and Perrig A., “Random key predistribution schemes for sensor networks”. In: Proceedings of the 2003 IEEE symposium on security and privacy, May 2003, pp. 197–213.

[3] L. Eschenauer and V. Gligor, “A Key Management Scheme for Distributed Sensor Networks,” Proc. 9th ACM Conf. Comp. and Commun. Sec., Nov. 2002, pp. 41-47.

[4] Liu D., and Ning P. “Establishing pairwise keys in distributed sensor networks”. In: Proceedings of 10th ACM conference on computer and communications security (CCS03). July 2003. pp. 41–47.

[5] Yu Z., and Guan Y. A “Robust group-based key management scheme for wireless sensor networks”. In: Proceedings of IEEE wireless communications and networking conference (WCNC 2005), New Orleans, LA USA. IEEE Press; July 2005. pp. 13–17.

[6] Lee J., and Stinson D.R. “Deterministic key predistribution schemes for distributed sensor networks”. In: Proceedings of ACM symposium on applied computing 2004, Lecture notes in computer science, vol. 3357, 2005, Waterloo, Canada, August 2004. pp. 294–307.

[7] Camtepe S.A., and Yener B. “Combinatorial design of key distribution mechanisms for wireless sensor networks”. IEEE/ACM Transactions on Networking (TON) 2007;15(2)pp:346–358.

[8] M. Eltoweissy , “Combinatorial Optimization of Key Management in Group Communications,” J. Network and Sys. Mgmt., Special Issue on Network Security, Mars. 2004, pp 33-50.

[9] N. Gura, A. Patel, A.Wander, H. Eberle, and S.C. Shantz. “Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs”. In CHES, Cambridge, MA, August 2004.

[10] Gaubatz G., Kaps J.P., and Sunar B. “Public key cryptography in sensor networks”. In: 1st European workshop on security in ad-hoc and sensor networks (ESAS 2004), Mars 2004.

[11] Qiang Huang, Johnas Cukier, Hisashi Kobayashi, Bede Liu and Jinyun Zhang, “Fast authenticated key establishment protocols for self-organizing sensor networks”, Proceedings of the 2nd ACM international conference on WSNs and applications, pp 141-150.

[12] Gang Han, Rui Zhou, and Hua Yang, “A SPT-Routing Key Management Scheme for Heterogeneous Wireless Sensor Networks Based on ECC”, International Conference on Uncertainty Reasoning and Knowledge Engineering, IEEE 2011, August 2011, pp. 243-247.

[13] Rui Zhou and Hua Yang, “A Hybrid Key Management Scheme for Heterogeneous Wireless Sensor Networks Based on ECC and Trivariate Symmetric Polynomial”, International Conference on Uncertainty Reasoning and Knowledge Engineering, IEEE 2011, August 2011, pp. 251-255.

[14] Cungang Yang, Celia Li, and Jie Xiao, “Location-based design for secure and efficient WSNs”, Elsevier 2008, pp. 3119–3129.

[15] Katerina Simonova, Alan C. H., Ling, X., and Sean Wang, “Location-aware Key Predistribution Scheme for Wide Area Wireless Sensor Networks”, SASN’06, ACM 2006, pp. 157-168.

[16] Yanchao Zhang, Wei Liu, Wenjing Lou and Yuguang Fang, “Securing Sensor Networks with Location-Based Keys, IEEE 2005, Mars 2005, pp. 1909-1914.

[17] Eltoweissy M, and Mukkamala R. “Dynamic key management in sensor networks”. IEEE Comm. Magazine 2006;April: pp. 122–130.

[18] M. Eltoweissy, “Group Key Management Scheme for Large-Scale WSNs” Ad Hoc Networks, 2005, pp.796-802.

[19] Mohamed Megahed, and Dimitrios Makrakis, “SurvSec: A New Security Architecture for Reliable Network Recovery from Base Station Failure of Surveillance WSN”, 2nd International Conference on Ambient Systems, Networks and Technologies, ANT 2011, August 2011, pp. 141-148.

[20] Krzysztof Piotrowski, Peter Langendoerfer and Steffen Peter, “How Public Key Cryptography Influences Wireless Sensor Node Lifetime”, Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks, ACM 2006, pp. 169-176.

[21] Qi Mi, John A. Stankovic, and Radu Stoleru, “Practical and secure localization and key distribution for wireless sensor networks”, Journal of Adhoc Networks, Elsevier 2012, August, pp. 946-961.

[22] Leonardo B. Oliveira, Hao C. Wong, M. Bern, Ricardo Dahab, and A. A. F. Loureiro, “SecLEACH – A Random Key Distribution Solution for Securing Clustered Sensor Networks”, Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications NCA 06, IEEE 2006, pp. 145-154.

[23] Chunguang Ma, Guining Geng, Huiqiang Wang, and Guang Yang, “Location-aware and secret share based dynamic key management scheme for WSN”, Networks Security, Wireless Communications and Trusted Computing Conference, IEEE 2009, April, pp. 770-773.

[24] Erik Dahmen and Christoph Krau, “Short Hash-Based Signatures for Wireless Sensor Networks”, 8th International Conference on Cryptology and Network Security, ACM 2009, pp. 463-476.

[25] Di Pietro, L. V. Mancini, and A. Mei, "Efficient and resilient key discovery based on pseudo-random key pre-deployment", IEEE Workshop on Wireless, Mobile, and Ad Hoc Networks, April 2004, pp. 2132-2140.

ISBN: 978-1-61208-207-3

Distributed Compromised Nodes Detection Scheme at First Stage for SurvSec Security Architecture

Mohamed Helmy Megahed
School of Information Technology and Engineering
University of Ottawa
Ottawa, Canada
mmega080@uottawa.ca

Dimitrios Makrakis
School of Information Technology and Engineering
University of Ottawa
Ottawa, Canada
dimitris@site.uottawa.ca

Hisham Dahshan
Communications Department
Military Technical College
Egyptian Armed Forces
Cairo, Egypt
hishamdahshan@yahoo.com

***Abstract*—SurvSec is a novel security architecture for reliable network recovery from base station BS failure of surveillance Wireless Sensor Network WSN in hostile environment. Compromised nodes detection is a very important security mechanism in surveillance WSN to detect compromised nodes before they destroy the security of the WSN. Node compromise attack is a multi-stage attack which consists of three stages: physically capturing and compromising sensor nodes; redeploying the compromised nodes back to network and compromised nodes rejoining the network. Only two protocols detect compromised nodes at first stage. The first protocol can be easily broken by targeting couple of nodes at the same time and the second protocol has high overheads and it is based on the distribution of one key list for all nodes which is not secure if one node is compromised. In this paper, a new compromised nodes detection algorithm that detects compromised nodes at first stage for SurvSec security architecture was proposed. The proposed scheme was based on four algorithms. First algorithm provided the network with key management. Second algorithm provided the network with secure localization. Third algorithm provided the network with secure clustering. Fourth algorithm built overlapped groups from clusters. Each cluster has a security manager (SM) and backup security manager (BKSM) to manage security issues. From the locations of nodes in the cluster, the nodes can form a group by sending and receiving from their right and left neighbours in the cluster. Each group forms overlapped group with its neighbour groups. The groups resemble interconnected rings in a chain and if attackers capture one group in the chain, the chain will be cut and its overlapped groups will discover the compromised group. Each node in the cluster sends an encrypted "Hello" message to its neighbours in the cluster every 15 seconds. If a node does not respond to the "Hello" message, this means it is compromised and its neighbours will send to the SM that the node is compromised and if the SM is compromised, its neighbours will send to the BKSM that the SM is compromised then to BS. The proposed protocol was designed to be resistant against large number of compromised nodes by collaborative work of attackers. Extensive simulation results were given to demonstrate the high detection rate of the proposed scheme besides the low overheads with high security level for the protocol.**

***Keywords-Overlapped Groups*; *Node Compromise Attack*; *First Stage*.**

I. INTRODUCTION

A node compromise attack is a three stage attack. In the first stage, the attacker captures some sensor nodes from the network and then compromises these nodes. In the second stage, these compromised nodes are redeployed into the network. In the third stage, the attacker will use these compromised nodes to launch various security attacks. Much work has tackled the node compromise attack [1, 2]. However, all of them address the node compromise attack either in the second stage based on node redeployment detection [2] or in the third stage based on node misbehavior detection [3-6]. We believe that group of attackers will launch node compromise attack to jeopardize the whole network in few minutes. Therefore, early detection of node compromise attack can lead to high security level.

Several protocols have been proposed for detecting compromised nodes at the second and third stage. Some protocols rely on the assumption that compromised node will change its location or its signal strength will alter after it is compromised.

Xiaodong [7] made the first attempt to detect node compromise in the first stage. He described a new couple based compromised node detection protocol to build couples of sensor nodes in ad-hoc pattern to detect node compromise attack at the first stage. The nodes within the same couple can monitor each other. This protocol assumes each sensor node can detect that it is connected to a programming board during the attack. After that the node will send a message to its couple to identify the other node that it is compromised. This protocol cannot be used against collaborative work of attackers to compromise large number of nodes where attackers can collect the couples at the same time. Also, it is not secure to depend on a message from the compromised node to its couple indicating that it is compromised.

The protocol [8] requires high storage overhead for one key list for the whole network, high communication overhead to broadcast "Hello" message to all neighbours then receive the same message from the neighbours, and high energy cost. Also, if one node is captured, then the key list is known to the attacker and the protocol is no longer secure.

In this paper, we developed a new overlapped groups based node compromise detection scheme. Compared with previously reported schemes, the proposed scheme detects

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-207-3

the node compromise attack in the first stage against large number of attackers with low overheads and with high security level. Specifically, after sensor nodes are deployed, first, they share link keys using our key management scheme; second, they perform secure localization using our secure localization scheme; third, they perform secure clustering so that each cluster has SM and BKSM; fourth, they perform overlapped groups for compromise node detection. Each node sends and receives from its interconnected nodes in the cluster to form a group. Every group is overlapped with other groups to form overlapped groups.

A. Contributions

1- We designed a homogenous network utilizing security managers SM and backup security managers BKSM to implement the distributed security concept.

2- We proposed to combine our key management with our secure localization and our secure clustering to develop high security level for the network before forming the overlapped groups.

3- We proposed a novel compromised nodes detection scheme at first stage based on formation of overlapped groups with low communication overhead, storage overhead and power cost. Each node in a group monitors its neighbours in the group. Each group is overlapped with other groups.

4- We designed the protocol such that our network is a chain and each group in the network is a ring in the chain and rings are interconnected therefore, if one ring is compromised, its interconnected rings will discover this.

B. Outline of the Paper

Section 2 presents the related work. Section 3 describes the assumptions and threat model. Section 4 describes the four algorithms to develop our overlapped groups based compromised nodes detection scheme at the first stage. Section 5 presents the security analysis. Section 6 presents the performance analysis. Section 7 presents the simulation results. Section 8 presents the comparison with others works. Section 9 concludes the paper.

II. RELATED WORK

In this section, we present related work to our proposed scheme.

A. SurvSec Security Architecture

Surveillance Security (SurvSec) is a new designed security architecture for reliable network recovery from single BS failure of surveillance WSN with single BS [18]. SurvSec relies on a set of sensor nodes serve as SMs for management and storage of the security related data of all sensor nodes. SurvSec has three components: (1) Sensor nodes serve as SMs, (2) Data Storage System, (3) Data Recovery System.

SurvSec is used for securing surveillance WSN during the time between the BS failure and the new mobile BS deployment which is the perfect time for attackers to compromise many legitimate nodes then destroy the security of the whole network. Also, SurvSec describes how the new BS will verify the trustworthiness of the deployed WSN otherwise a new WSN must be deployed.

B. Compromised Nodes Detection Schemes

We need an effective security scheme to identify compromised nodes in a timely manner because compromised nodes in surveillance WSN represent uncovered areas. A node compromise attack involves three stages. From [1-6], the authors proposed many protocols to detect compromised nodes based on location, signal strength, reputation, weighted trust, intrusion detection and MAC layer misbehavior. However, these approaches are not effective since they can detect compromised nodes on the second or the third stage and they depend on node’s misbehavior or node’s location, which means a node may be compromised but behaves well until a programmed time. In [7], a couple based compromised node detection protocol at first stage is proposed to build couples of sensor nodes where the couple can monitor each other but this scheme cannot be used against collaborative work of attackers to compromise large number of nodes because attackers can collect the couples at the same time. Also, we cannot depend on a message from the compromised node to its monitored node. Two protocols [8] are proposed based on four messages. Each sensor node broadcasts a “Hello” message to his neighbors which receive this message and reply to it. If the node did not send for three times, it is marked as compromised and the compromised node neighbors flood the network with the node is compromised message. This protocol uses one key list for the whole network which is insecure in addition to large communication overhead, storage overhead, and high power cost.

Also, software-based attestation techniques [9-17] have been proposed to verify the contents of the code running on nodes where the node’s free memory space is filled with incompressible random noise known to the attester.

For the detection in the second stage: In [2], Song et al. made the first attempt to detect compromise node in the second stage. They assume that an adversary will not be able to precisely deploy the compromised sensors back into their original positions.

For the detection in the third stage: In [1], Carl et al. demonstrate the case in which compromised nodes can be detected in the third stage and they show exactly what information can be obtained and how it can be used to disrupt, falsify data within, or eavesdrop on sensor networks. They suggest that sensor nodes in hostile environment would be desirable not to respond to the standard on chip debugging and if a node can detect its own movement by either accelerometers or GPS then it can preemptively delete important information stored in SRAM.

In [3], Kyasanur and Vaidya propose modifications to IEEE 802.11 MAC protocol to simplify misbehaviour detection. Once the sensor nodes are compromised, they will launch false data injection attack. Thus, several en-route filtering schemes [4, 5] have been proposed to drop the false data en-route before they reach the sink. Nevertheless, these schemes only mitigate the threats. Thus Ye et al. [6] propose a probabilistic nested marking scheme to locate colluding compromised nodes in false data injection attacks. Recently, several software-based attestation schemes for node compromise detection in sensor networks have been proposed [11].

III. NETWORK MODEL&THREAT MODEL

A. Network Model

We consider a hierarchical WSN consisting of a BS, sensor nodes which are grouped into clusters and beacon nodes equipped with GPS called beacons. Each node has a unique ID, unique location and unique certificate. The assumptions of model are as follows:

1- We assume sensor nodes are static and some nodes continuously store the detected security threats and all other security data related to nodes where these nodes are SMs.

2- Nodes in the cluster form a group and every group is overlapped with other groups.

B. Threat Model

We consider a group of attackers that try to uncover the keys of the network through capturing some nodes then redeploy the compromised nodes in the network again.

IV. PROPOSED SCHEME

The proposed scheme has four phases which are key management phase to distribute keys among nodes, secure localization phase to determine nodes locations, secure clustering phase to choose BKSM to revoke SM if it is compromised, and forming overlapped groups phase for the overlapped groups based compromised nodes detection protocol at first stage. The proposed scheme has four types of sensors: SMs, BKSMs, initiators and sensor nodes.

A. Key Management Phase

We propose a novel hybrid and dynamic key management protocol utilizing our novel scheme of certificates shared verification to eliminate the needs for high end sensor nodes HSNs which have high power for intensive calculation of public key operations. High end sensor nodes are the best targets for the attackers in the hostile environment. Our proposed key management scheme has two steps which are: key predistribution and key establishment. HSN is the nodes cluster head.

i. Key Predistribution

The key predistribution step consists of acquiring the sensors certificate from the certificate authority CA. ECC is used in this protocol to perform security functions on sensors with limited computing resources. The protocol uses the elliptic curve implicit certificate scheme [19] instead of X.509 because of the resulting low storage overhead, low communication overhead, which is a dominant factor for low bit transmission channels in WSN.

The certificate generation processes for any sensor node U is performed offline before it joins the network.

1- CA selects a random integer q_{CA} as its static private key, and computes the public key $Q_{CA}= q_{CA}\ X\ P$. q Multiplied by P.

2- To obtain a certificate and private-public key pair, the sensor U randomly selects a key pair (q_U, Q_U) where $Q_U= q_U\ X\ P$ and sends Q_U and q_U to CA. U sends its public and private key to CA so that CA can verify the pair. CA is not on the network, so it works off line.

3- CA verifies U's identity and private-public key pair.

4- The implicit certificate for U is the concatenation of CA's public key Q_{CA}, the device identity ID_U, the U public key Q_U and the certification expiration date t_U , i.e., the certificate is (Q_{CA}, ID_U, Q_U, t_U) signed by the CA private key using ECDSA.

ii. Key Establishment

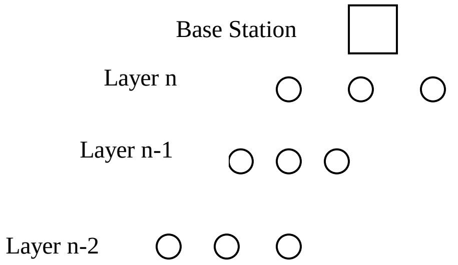


Fig. 1 Network Topology

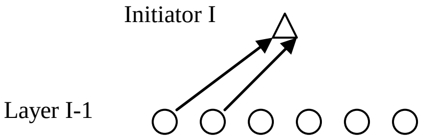


Fig. 2.a Certificates Verification using Initiator for 2 nodes

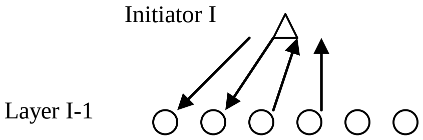


Fig. 2.b Certificates Verification using Initiator for 2 nodes

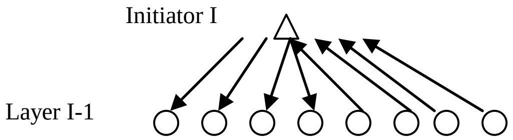


Fig. 2.c Certificates Verification using Initiator for 4 nodes

Figure 1 shows the network topology and Figure 2 shows the certificates shared verification process for one layer using algorithm 1 that uses initiator nodes to start the process of key establishment protocol. We assume that there are nodes named as security managers SMs and these nodes are located every two layers. We assume that there are nodes named as initiators every predefined number of nodes such as 30, 20 or 10 nodes to start the operation of key management process. Initiator node verifies certificates of first two nodes then it sends the certificates of the second two nodes to the verified first two nodes then it sends certificates of other four nodes to the verified four nodes. Algorithm 1 is efficient in terms of distribution of power consumption among sensor nodes in the cluster and it can be used with all SMs in their clusters. The nodes under the initiator are ordinary nodes and the nodes under the ordinary nodes are SMs and so on until we reach another initiator.

Each node in HSN model performs four times certificate verification for three beacon nodes and for HSN certificates. With the same number of certificates verification at each node, we developed our proposed certificates shared verification scheme. Each node in our scheme verifies four certificates only with the cost of increasing the communication overhead with four messages for every node. These verifications are: first verification for SM certificate, two verifications for two nodes in the cluster, and one verification for beacon node certificate.

We explain our scheme in the form of an algorithm.

Algorithm 1: Initiator nodes to start key management process

1: I → n : { I (Q_{CA}, ID_{SM}, Q_{SM}, t_{SM}) }

Each initiator node broadcasts its certificate to its underneath nodes at layer n to verify it. The nodes at layer n verify the certificate of the initiator. The initiator node is a SM.

2: n → I : { n (Q_{CA}, ID_U, Q_U, t_U) }

The initiator node receives the certificates of its underneath nodes for verification. We assume there are n nodes underneath the initiator node. First, the initiator node verifies the certificate of the first two nodes.

3: I → $n_{1,2}$: { share link keys }

The initiator node shares link keys with node 1 and node 2 as steps from 4 to 9.

4: n : selects (k), calculates (d_U), encrypts (d_U)

Each node underneath initiator I at layer n selects a k-bit random number c_U of 160 bits to produce its link key contribution with the BS. Each node at n calculates the value of $d_U = H(c_U \parallel ID_U)$ where H is a cryptographic hash function. Each node at n encrypts d_U with I public key Q_I. To encrypt and send a message d_U to I, each node at n chooses a random positive integer x and produces the ciphertext C_m consisting of the pair of points which are: $C_m = (x\ P,\ d_U\ + x\ Q_I)$. We add the number d_U to both values of the point $x\ Q_I$.

5: n → I : { C_m }

Each node underneath I at layer n sends its encrypted link key contribution with the I which is C_m.

6: I : decrypts (C_m), selects (k), calculate (d_{BS}), encrypts (d_{BS})

I decrypts C_m for every node at n. I multiplies first point in the pair by I's private key and subtracts result from second point: $d_U + x\ Q_I - q_I\ (x\ P) = d_U\ + x\ (q_I P) - q_I\ (x\ P) = d_U$.
I selects a k-bit random number c_I of 160 bits for each node near I to produce its link key contribution with nodes near I. I calculates the value of $d_{BS} = H(c_I \parallel ID_I)$ for every node near I where H is a cryptographic hash function.
I encrypts d_{BS} for every node near I using symmetric key encryption under key d_U, generating value $y = E_{du}\ (ID_I \parallel d_{BS})$.

7: I → n : { y }, {hash {K}}

I sends y, the encrypted link key contribution of I, to every node near I. I generates the link key with every node near the I at n by calculating $K = H\ (d_u \parallel ID_U \parallel d_I \parallel ID_I)$ then $H(K)$ where H is a cryptographic hash function. I sends $H(K)$ of every node at n to its participant to achieve correctness.

8: n : decrypts (y), calculates (K)

Every node at n decrypts the received message y using symmetric key encryption under key d_U to obtain the value d_I.
Every node at n generates the link key with I by calculating the $K = \mathrm{H}(d_u \parallel ID_U \parallel d_I \parallel ID_I)$.

9: n → I : {z}

Every node at n calculates $z = \mathrm{H}(K)$ and sends z to I. I checks if $z = H(K)$. If yes, the link key is established correctly. Otherwise, the protocol is terminated.

10: I → $n_{1,2}$: { $n_{3,4}$ (Q_{CA}, ID_U, Q_U, t_U) }

The initiator node sends to node 1 and node 2 underneath the certificates of node 3 and node 4 for verification.

11: $n_{1,2}$ → I : { valid certificates or invalid certificates }

Node 1 and node 2 send to the initiator node two messages indicating that certificates of nodes 3 and 4 are valid or not.

12: I → $n_{3,4}$: { share link keys }

The initiator node shares link keys with node 3 and node 4 as steps from 4 to 9 in algorithm 1.

13: I → $n_{1,2,3,4}$: { $n_{5,6,7,8}$ (Q_{CA}, ID_U, Q_U, t_U) }

The initiator node sends to node 1, node 2, node 3 and node 4 underneath the certificates of node 5, node 6, node 7 and node 8 for verification and nodes 1, 2, 3, 4 respond with valid certificate or not.

14: I → $n_{5,6,7,8}$: { share link keys }

The initiator node shares link keys with node 5, node 6, node 7 and node 8 as steps from 4 to 9 in algorithm 1. **Finally**, the process of the initiator continues to verify all

```
    of its underneath nodes then its underneath nodes use algorithm 1 to share link keys with their underneath nodes and so on.
```

1. The second layer after the initiator is SMs and so on until the initiator layer because initiators are defined every 30 or 20 or 10 nodes.
2. After the SMs and the sensor nodes establish link keys, they determine their locations using our proposed secure localization scheme with certificates shared verification.

B. Secure Localization Phase

A number of secure localization algorithms [20] have been reported. Different researchers have different strategies to categorize them. These strategies can be divided into direct and indirect localization, centralized and distributed localization, range-based and range-free localization, absolute and relative localization. We propose to get the location information to form the group from the followings approach:

The indirect approaches of localization were introduced to overcome some of the drawbacks of the GPS-based direct localization techniques while retaining some of its advantages. In this approach, a small subset of nodes in the network, called the beacon nodes, are equipped with GPS receivers to compute their location. Beacon nodes send beams of signals providing their location to all nodes in their vicinity. Using the transmitted signal containing location information, nodes compute their location. Each node needs three beacon nodes to locate its position.

Our proposed scheme depends on SM and certificates shared verification for secure localization. We assume that each cluster has three beacon nodes. Sensor nodes in the cluster send the beacon nodes certificates to SM then SM sends these certificates to its underneath nodes for verification to insure one verification time for beacon nodes certificates for the whole cluster. This is done because Verification power is 1000 times more than communication power [21]. SM assures that certificate verification for beacon nodes is done only once for the whole cluster to reduce the power of verification. Each node needs to verify three beacon nodes with total of 3n verifications but with certificate shared verification this is done once. SMs are clusterheads.

Algorithm 2: Secure Localization

```
1: Beacons$_{1,2,3}$ → SM$_n$ : {Beacons$_{1,2,3}$ ($Q_{CA}$, $ID_B$, $Q_B$, $t_B$) }
    The beacon nodes near BS broadcast their certificates and locations to SMs near BS. We need three beacon nodes to locate the position.
2: SM$_n$ → BS : { Beacons$_{1,2,3}$ ($Q_{CA}$, $ID_B$, $Q_B$, $t_B$) }
    The SMs near BS at layer n send the certificates of the beacon nodes to BS for verification.
3: BS → SM$_n$ : {valid certificates of Beacons$_{1,2,3}$ }
    BS sends to SMs at layer n that beacon nodes certificates are valid.
4: SM$_n$ → Beacons$_{1,2,3}$ : { Key$_{1,2,3}$ }
    Every SM at layer n shares a link key with the three beacon nodes in four steps.
5: SM$_n$ : calculates ($x$, $y$) position
    Every SM at layer n calculates its position.
6: Beacons$_{1,2,3}$ → n-1 : {Beacons$_{1,2,3}$ ($Q_{CA}$, $ID_B$, $Q_B$, $t_B$) }
    The beacon nodes near BS broadcast their certificates and locations to nodes at layer n-1.
7: n-1 → SM$_n$ : { Beacons$_{1,2,3}$ ($Q_{CA}$, $ID_B$, $Q_B$, $t_B$) }
    The nodes at layer n-1send the certificates of beacon nodes to SMs at layer n for verification. If the beacon nodes certificates are previously verified, it is ok but if there are new beacon nodes certificates, then SMs at layer n send the new beacon nodes certificate to BS for verification.
8: SM$_n$ → n-1 : { Key$_{1,2,3}$ }
    Every SM at layer n sends its link keys with the beacon nodes to its connected nodes at layer n-1.
9: n-1 : calculates ($x$, $y$) position
    Every node at layer n-1 calculates its position.
10: Beacons$_{4,5,6}$ → SM$_{n-2}$ : {Beacons$_{4,5,6}$ ($Q_{CA}$, $ID_B$, $Q_B$, $t_B$) }
    The beacon nodes near SMs at layer n-2 broadcast their certificates and locations to SMs at layer n-2.
11: SM$_{n-2}$ → SM$_n$ : { Beacons$_{4,5,6}$ ($Q_{CA}$, $ID_B$, $Q_B$, $t_B$) }
    The SMs at layer n-2 send the certificates of the beacon nodes to SMs at layer n for verification.
12: SM$_n$ → n-1 : { Beacons$_{4,5,6}$ ($Q_{CA}$, $ID_B$, $Q_B$, $t_B$) }
    The SMs at layer n send the certificates of the beacon nodes to nodes at layer n-1 for verification.
13: n-1 → SM$_n$ : { valid certificates of Beacons$_{4,5,6}$ }
    The nodes at layer n-1 send to SMs at layer n that beacon nodes certificates are valid.
14: SM$_n$ → SM$_{n-2}$ : { valid certificates of Beacons$_{4,5,6}$ }
    The SMs at layer n send to SMs at layer n-2 that beacon nodes certificates are valid.
15: SM$_{n-2}$ → Beacons$_{4,5,6}$ : { Key$_{4,5,6}$ }
    Every SM at layer n-2 shares a link key with the three beacon nodes in four steps.
16: SM$_{n-2}$ : calculates ($x$, $y$) position
    Every SM at layer n-2 calculates its position.
17: Beacons$_{4,5,6}$ → n-3 : {Beacons$_{4,5,6}$ ($Q_{CA}$, $ID_B$, $Q_B$, $t_B$) }
    The beacon nodes near nodes at layer n-3 broadcast their certificates and locations to nodes at layer n-3.
18: n-3 → SM$_{n-2}$ : { Beacons$_{4,5,6}$ ($Q_{CA}$, $ID_B$, $Q_B$, $t_B$) }
    The nodes at layer n-3 send the certificates of beacon nodes to SMs at layer n-2 for verification. If the beacon nodes certificates are previously verified, it is ok but if there are new beacon nodes certificates, then SMs at
```

ISBN: 978-1-61208-207-3

layer n-2 send the new beacon nodes certificate to SMs at layer n for verification.

19: SM$_{n-2}$ → n-3 : { Key$_{4,5,6}$ }

Every SM at layer n-2 sends its link keys with the beacon nodes to its connected nodes at layer n-3.

20: n-3 : calculates (*x*, *y*) position

Every node at layer n-3 calculates its position. **<u>Finally</u>**, lower layer SMs send certificates of beacon nodes to higher layer SMs for verification.

1. Certificates shared verification for beacon nodes certificates between SM and its underneath nodes will reduce setup time and reduce computations complexity at the cost of increasing only four messages.
2. Certificates verification for beacon nodes is done only one time not multiple times at each node underneath the SM to reduce computations complexity.
3. Sensor nodes underneath SM will use the shared keys between the SM and the beacon nodes which will reduce the setup time, computations and storage overhead.
4. After the SMs and the sensor nodes determine their locations, they form secure clustering.

C. Secure Clustering Phase

SMs can form secure clustering [22] with their nodes underneath and SM can choose BKSM to replace it if the SM is compromised.

Algorithm 3: Secure Clustering

1: BS → n : {req SM_msg }

BS sends to nodes near BS at layer n that these nodes are SMs using its shared symmetric key with these nodes.

2: SM$_n$ → n-1 : { adv cluster_msg }

Every SM at layer n sends an encrypted advertise message to nodes at layer n-1 to form a cluster.

3: n-1 → SM$_n$: { join cluster_msg }

Every node at layer n-1 sends an encrypted message to its SM at layer n to join the cluster.

4: SM$_n$ → n-1 : {choose BKSM }

The SM at layer n chooses BKSM according to maximum connectivity between the BKSM and the nodes in the cluster where BKSM must be connected to all nodes in the cluster.

5: n-1 → n-2 : { req SM_msg }

The nodes at layer n-1 send to nodes at layer n-2 an encrypted message that these nodes are SMs.

6: SM$_{n-2}$ → n-3 : { adv cluster_msg }

Every SM at layer n-2 sends an encrypted advertise message to nodes at layer n-3 to form a cluster.

7: n-3 → SM$_{n-2}$: { join cluster_msg }

Every node at layer n-3 sends an encrypted message to its SM at layer n-2 to join the cluster.

8: SM$_{n-2}$ → n-3 : {choose BKSM }

The SM at layer n-2 chooses BKSM according to maximum connectivity between the BKSM and the nodes in the cluster where BKSM must be connected to all nodes in the cluster.

9: BKSM$_{n-2}$ → n-3 : { BKSM (Q_{CA}, ID_{BKSM}, Q_{BKSM}, t_{BKSM}) }

The BKSM at layer n-2 sends its certificate to the nodes at layer n-3 where SM at layer n-2 verifies this certificate. **<u>Finally</u>**, the steps of forming the secure clustering are performed until the lower layer of SMs.

1. Our proposed secure clustering scheme assumes hybrid key management protocol to achieve high security level.
2. Our proposed scheme chooses BKSM to solve the problem of compromised SM and to sign the message of revoked SM.
3. Our scheme achieves secure clustering in four messages.

D. Forming Overlapped Groups Phase

Each node in a cluster sends its location to its SM. From the nodes locations at the SM, the SM starts the process to form a group. Assume each cluster has n nodes and the SM builds the overlapped group from the nodes in the cluster as shown in algorithm 4.

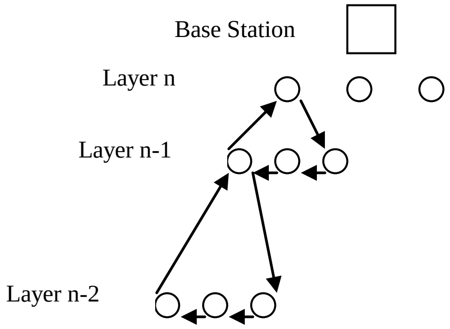


Fig. 3 Overlapped Groups Formation

Figure 3 shows the overlapped groups formation. Algorithm 4 represents forming a group from seven nodes which are labeled from n to n-6. The SM sends a message to its nearest node containing the sequence of sending and receiving messages in the cluster to form a group according to each node neighbours. The SM chooses a group key for the cluster and sends it to all nodes in the cluster. A group is a cluster.

Algorithm 4: Forming Overlapped Groups

1: SM$_n$ → n, n-6 : { join group_msg }

SM at layer n sends an encrypted message to node n and node n-6 to form the group and the message contains the interconnections of all nodes in the cluster to form the group. The used key is the group key between the SM and the nodes in the cluster. The sent message includes what every node is connected to in the cluster to form a closed loop.

2: n → n-1, SM$_n$: { join group_msg }

Node n sends an encrypted message to node n-1 and SM to complete the process of forming a group. The message contains the interconnections of all nodes in the cluster. The used key is the group key between the SM and nodes in the cluster plus one. The sent message is the join group message.

3: n-1 → n, n-2 : { join group_msg }

Node n-1 sends an encrypted message to node n and node n-2 to complete the process of forming a group. The message contains the interconnections of all nodes in the cluster. The used key is the group key between the SM and nodes in the cluster plus two.

4: n-2 → n-1, n-3 : { join group_msg }

Node n-2 sends an encrypted message to node n-1 and the node n-3 to complete the process of forming a group. The message contains the interconnections of all nodes in the cluster. The used key is the group key between the SM and nodes in the cluster plus three.

5: n-3 → n-2, n-4 : { join group_msg }

Node n-3 sends an encrypted message to node n-2 and the node n-4 to complete the process of forming a group. The message contains the interconnections of all nodes in the cluster. The used key is the group key between the SM and nodes in the cluster plus four.

6: n-4 → n-3, n-5 : { join group_msg }

Node n-4 sends an encrypted message to node n-3 and the node n-5 to complete the process of forming a group. The message contains the interconnections of all nodes in the cluster. The used key is the group key between the SM and nodes in the cluster plus five.

7: n-5 → n-4, n-6 : { join group_msg }

Node n-5 sends an encrypted message to node n-4 and the node n-6 to complete the process of forming a group. The message contains the interconnections of all nodes in the cluster. The used key is the group key between the SM and nodes in the cluster plus six.

8: n-6 → n-5, $\mathbf{SM_n}$: { join group_msg }

Node n-6 sends an encrypted message to node n-5 and the SM at layer n to complete the process of forming a group. The message contains the interconnections of all nodes in the cluster. The used key is the group key between the SM and nodes in the cluster plus seven.

Finally, the "Hello" message is sent from one node to two neighbour nodes in the cluster and the two nodes respond to the "Hello" message. If the node is compromised, it will not send the "Hello" message and therefore, the recipient nodes will mark it as compromised and they will send to the SM to revoke that node. If the SM is compromised, its monitored nodes will send to the BKSM to revoke the SM.

1- Our proposed compromised nodes detection scheme is based on the overlapped groups to discover the compromised group. If a node is compromised in a group, it will be detected by its neighbours who will send to SM that this node is compromised. If a SM is compromised, its neighbour nodes will send to the BKSM to revoke it.

2- Each node sends at first time with key K then next time with key K+n+1 and next time with key K+2n+1 and so on.

3- Each node sends a "Hello" message and receive two messages from its neighbours in 15 seconds.

4- Each group forms an overlapped group with its upper group and its lower group.

5- We designed the compromised nodes detection protocol at first stage such that our network resembles a chain and each cluster in the network forms a group and each group is a ring in the chain and rings are interconnected therefore, if one ring is compromised, its interconnected rings will discover this.

V. SECURITY ANALYSIS

Security analysis of our protocol focuses on resilience to node compromise attack, collusion attack and impersonation attack.

A. Compromised Node Attack

1- If an attacker compromises one regular node, therefore, the probability of insecure link is $P_{insec} = 1/N$ where N is the number of nodes at the network. For n compromised regular nodes the probability of insecure links is $P_{insec} = n/N$.

2- If the attacker compromises one SM, therefore, the probability of insecure links is $P_{insec} = (n_s + 4) /N$ where n_s is the number of nodes in the cluster of the SM. For n compromised SMs the probability of insecure links is $P_{insec} = n (n_s + 4) /N$.

3- Our proposed key management assumes compromised node detection at the first stage and compromised nodes revocation. Therefore, SM will revoke the regular compromised node and the BKSM will revoke the SM to eliminate the insecure links.

B. Collusion Attack

Two nodes can collude when they share their keys with each other. Our designed protocol is resistant to collusion attack because each sensor node communicates only with a SM therefore; compromised nodes cannot discover themselves.

C. Impersonation Attack

Each node has a certificate to join the key management process and to join the network. This prevents the attacker from impersonating any legitimate node. Also, knowing the public key of the SM will not reveal the private key for the SM because this needs the attacker to solve the elliptic curve discrete logarithmic problem ECDLP which is a hard problem.

VI. PERFORMANCE ANALYSIS

The performance analysis is measured in computation complexity, communication complexity and storage complexity. We assume that the network is secure during setup time which depends on number of initiators.

A. Computation Complexity

SM generates a group key and sends it encrypted with the shared link key with every node in the cluster to use it in the process of compromised nodes detection. Each sensor node decrypts the message sent with group key with its shared link key with the SM.

Our scheme has lower computation overhead than the scheme that uses couples to detect compromised nodes at the first stage. Our scheme has the same computation overhead compared to the scheme uses distributed compromised nodes detection at first stage. Our scheme has low computation overhead to generate the group key and to send it encrypted to all nodes in the group.

B. Communication Complexity

Communication complexity is the number and size of packets sent and received by a sensor node. In our protocol, the number of messages sent is one message every 15 seconds and there are two messages received every 15 seconds with total of three messages sent and received every 15 seconds to establish the compromised nodes detection protocol. Our scheme has lower communication overhead than the other two schemes that detects compromised nodes at the first stage.

C. Storage Complexity

Storage complexity is the amount of memory units required to store security credentials. Each sensor node stores the group key with the SM and other nodes in the cluster. Our scheme has the same storage overhead as the scheme uses couples to detect compromised nodes at the first stage but it has lower storage overhead than the scheme uses distributed compromised nodes detection at first stage. Our scheme needs to store only one key which is the group key between the SM and the nodes in the group

VII. SIMULATION RESULTS

We built a model for the proposed design and we implemented a simulator in MATLAB that can scale to thousand of nodes. In this simulator, sensors can send and receive data from each other's. The simulation verifies the correctness and the feasibility of our security architecture. It is our future work to implement SurvSec in some sensor network testbeds with all its ingredients. Our simulation scenarios include N nodes distributed randomly. We choose N as 1000 sensor nodes.

The followings are the built models for simulation:

1- Network setup model for the overlapped groups.
2- Compromised nodes detection protocol.

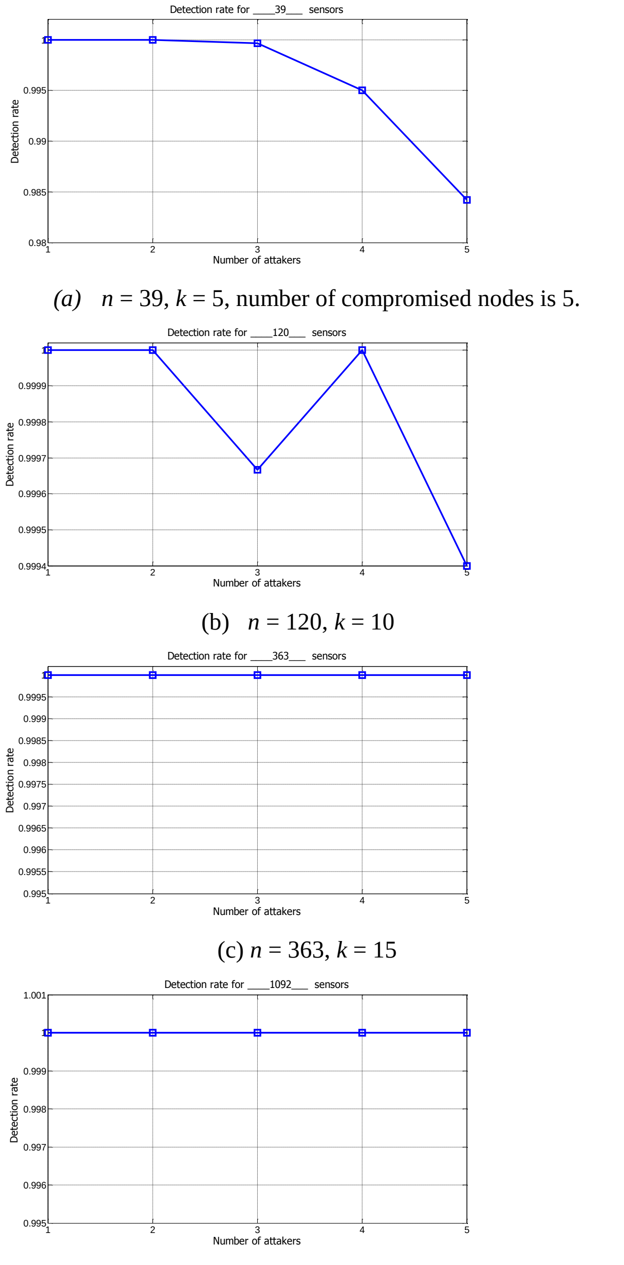
In the simulations, these parameters are given as follows:

1- The number of sensor nodes *n* is varied from 39 to 1000 sensor nodes.
2- The interval of beacon information is set to 15 seconds.
3- The time of an adversary to successfully compromise a sensor node is varied from 30 seconds to 60 seconds.

In this section, we evaluate the detection rate under different n.

The detection rate is equal to the detected compromised sensor nodes over all compromised nodes.

In the proposed adversary model, we assume that an adversary can simultaneously compromise *k* sensor nodes, where *k*<*n*.



(a) *n* = 39, *k* = 5, number of compromised nodes is 5.

(b) *n* = 120, *k* = 10

(c) *n* = 363, *k* = 15

(d) *n* = 1092, *k* = 25

Fig. 4. Detection rate varies with number of compromised nodes under different n =39, 120, 363, 1092, Interval = 15 Sec.

ISBN: 978-1-61208-207-3

Thus, we first evaluate the detection rate under different parameters *n*, *k* and beacon interval and the results are shown in Figure 4. From Figure 4, we can see the detection rate does not increase linearly with *k*. When n = 363 or n = 1092, the detection rate reaches the maximum. Due to this observation, when the number of sensor nodes increase, we found that the proposed scheme has high resiliency against node compromise attack by collaborative work of attackers at the same time for large hierarchical WSN.

VIII. COMPARISON WITH OTHER WORKS

Now, we compare between our proposed model and previous works that detects compromised nodes at first stage.

TABLE 1, COMPARISON BETWEEN OUR MODEL AND OTHER MODELS.

	Property	**CAT [7]**	**Distributed Detection [8]**	**Our Model**
1	Detect compromised nodes for group of attackers	No	Yes	Yes
2	Detection rate	Less than 100%	Near 100%	Near 100%
3	Communication overhead	14 messages every 15 sec for beacon every 2 sec High overhead	At least 6 messages every 15 sec moderate overhead	At least 3 message s every 15 sec low overhead
4	Computation overhead	Low	Low	Low
5	Storage overhead	Low to store one key	High to store key list	Low to store one key
6	Setup time	Low	Low	Low
7	Power cost	High	High	Low

Our proposed model can be used against collaborative work of attackers to compromise large number of nodes at the same time. Also, the detection rate is near 100%. Our model has low communication overhead and low computation overhead and low storage overhead. Our model has low power cost since it sends and receives only three messages every 15 sec which is lower than the other two schemes. Key predistribution time is equal to the time that is needed from the initiator to distribute keys with its underneath nodes because initiators work separately.

IX. CONCLUSION

In this paper, we proposed the overlapped groups-based compromised nodes detection scheme to early detect the node compromise attack in the first stage. Concretely, the simulation results showed that by building groups among neighboring sensor nodes in a local area, physical node compromise attack can be detected immediately. Also, the simulation results showed that the proposed detection scheme has high detection rate. This work is an initial work to form overlapped groups for detecting compromise attack at the first stage and we do not expect that the proposed scheme will solve all the problems in the node compromise nodes attack. Our future work will continue to build more overlapped groups to early detect compromise nodes attack.

REFERENCES

[1] C. Hartung, J. Balasalle, and R. Han, “Node compromise in sensor networks: the need for secure systems,” in *Technical Report CU-CS-990-05, Dept. of Comp Sci, Univ of Colorado at Boulder*, Jan 2005.

[2] H. Song, L. Xie, S. Zhu, and G. Cao, “Sensor node compromise detection: the location perspective,” in *IWCMC’07*, Honolulu, Hawaii, USA, Aug. 2007.

[3] P. Kyasanur and H. Vaidya, “Detection and handling of mac layer misbehavior in wireless networks,” in *IEEE DSN*, 2003.

[4] S. Zhu, S. Setia, S. Jajodia, and P. Ning, “An interleaved hop-by hop authentication scheme for filtering of injected false data in sensor networks,” in *IEEE Symposium on Security and Privacy’04*, 2004.

[5] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh, “Toward resilient security in wireless sensor networks,” in *ACM MobiHoc’05*, 2005.

[6] F. Ye, H. Yang, and Z. Liu, “Catching moles in sensor networks,” in *IEEE ICDCS’07*, Jun, 2007.

[7] Xiaodong Lin, “*CAT: Building Couples to Early Detect Node Compromise Attack in Wireless Sensor Networks*”, IEEE "GLOBECOM" 2009.

[8] Wei Ding, Yingbing Yu, and Sumanth Yenduri, “Distributed First Stage Detection for Node Capture”, IEEE Globecom 2010.

[9] A. Seshadri, M. Luk, E. Shi, A. Perrig, L. van Doorn, and P. Khosla, “Pioneer: verifying integrity and guaranteeing execution of code on legacy platforms,” in *SOSP*, Oct. 2005.

[10] D. Spinellis, “Reflection as a mechanism for software integrity verfication,” in *ACM Trans. Inf. Syst. Secu.*, Vol, 3, No, 1, 2000.

[11] Yi Yang, Xinran Wang, Sencun Zhu, and Guohong Cao, “*Distributed Software-based Attestation for Node Compromise Detection in Sensor Networks*”, 26th IEEE International Symposium on Reliable Distributed Systems, IEEE 2007.

[12] Taejoon Park, and Kang G. Shin, “*Soft Tamper-Proofing via Program Integrity Verification in Wireless Sensor Networks*”, IEEE Transactions on Mobile Computing, Vol. 4, No. 3, May/June 2005, IEEE 2005.

[13] Xiaojiang Du, “Detection of Compromised Sensor Nodes in Heterogeneous Sensor Networks”, IEEE "ICC" 2008.

[14] Arvind Seshadri, Adrian Perrig, Leendert van Doorn, and Pradeep Khosla, “*SWATT: SoftWare-based ATTestation for Embedded Devices*”, In IEEE Symposium on Security and Privacy (2004), IEEE Computer Society 2004.

[15] Tamer AbuHmed, Nandinbold Nyamaa, and DaeHun Nyang, “*Software-Based Remote Code Attestation in Wireless Sensor Network*”, IEEE "GLOBECOM" 2009.

[16] Jun-Won Ho, Matthew Wright, and Sajal K. Das, “*ZoneTrust: Fast Zone-Based Node Compromise Detection and Revocation in Sensor Networks Using Sequential Analysis*”, 2009 28th IEEE International Symposium on Reliable Distributed Systems, IEEE 2009.

[17] J. Deng, R. Han, and S. Mishra, “*Secure Code Distribution in Dynamically Programmable Wireless Sensor Networks*”, In Proc.

International Conference on Information Processing in Sensor Networks, pp. 292–300, ACM 2006.

[18] Mohamed Megahed, and Dimitrios Makrakis, “SurvSec: A New Security Architecture for Reliable Network Recovery from Base Station Failure of Surveillance WSN”, 2nd International Conference on Ambient Systems, Networks and Technologies, ANT 2011

[19] Rene Struik and Gregg Rasor, “Mandatory ECC Security Algorithm Suite”, IEEE 2002, Wireless Personal Area Networks, March 2002.

[20] C. Savarese, J. Rabay and K. Langendoen. “Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks”. USENIX Technical Annual Conference, Monterey, CA, June 2002.

[21] Krzysztof Piotrowski, Peter Langendoerfer and Steffen Peter, “How Public Key Cryptography Influences Wireless Sensor Node Lifetime”, Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks, ACM 2006.

[22] Leonardo B. Oliveira, Hao C. Wong, M. Bern, Ricardo Dahab, and A. A. F. Loureiro, “SecLEACH – A Random Key Distribution Solution for Securing Clustered Sensor Networks”, Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications NCA 06, IEEE 2006.

Range Free Localization of Wireless Sensor Networks Based on Sugeno Fuzzy Inference

Mostafa Arbabi Monfared
Department of Electrical & Electronic Engineering
Eastern Mediterranean University
Famagusta, via Mersin10, Turkey
mostafaarbabi@hotmail.com

Reza Abrishambaf, Sener Uysal
Department of Electrical & Electronic Engineering
Eastern Mediterranean University
Famagusta, via Mersin10, Turkey
{reza.abrishambaf, sener.uysal}@emu.edu.tr

Abstract—**One of the challenges in wireless sensor networks is to determine the location of sensor nodes based on the known location of other nodes. This paper identifies an intelligent localization method, which is based on range free localization to estimate the location of the unknown nodes. In the proposed method, the anchor nodes are connected to the sensor nodes and then each sensor node receives a signal from the anchor node. The Received Signal Strength Indicator is then calculated by the node. The RSSIs are calculated based on the distance of the sensor node to each anchor node. The RSSIs are, then, fed to the Sugeno fuzzy inference system to calculate the weights to be used in the centroid relation. The centroid technique is proposed to estimate the location of the unknown sensor nodes. Both analytical and experimental results are discussed in this paper. The results show that with increasing the membership functions, the error decreases and that is because of the RSSI graph, which better fits the corresponding simulation result.**

Keywords-Range-free Localization; Received Signal Strength Indicator (RSSI); Centroid localization; Fuzzy Logic Systems; Wireless Sensor Networks.

I. INTRODUCTION

Wireless Sensor Networks consist of unique nodes, which are small, battery powered devices that can compute and communicate different signals in a target environment. The WSNs have many applications in building, air traffic control, manufacturing, automation, environment monitoring, other industrial and security applications [1].

The recent developments of micro electro mechanical systems (MEMS), communication technology and computing have motivated the use of massive distributed wireless sensor networks, which consist of hundreds or thousands of nodes. Thus every node is able to sense the environment, compute one or more tasks and communicate with central unit or other sensors [2].

Wireless Sensor Networks are particularly attractive in risky environments, specifically in a large deployment. In WSN applications, one of the important problems is the location of the unknown sensor nodes for the base service. The design of efficient localization algorithms depends on a successful localization technique to compute the correct sensor position in some coordinate system.

There are two kinds of nodes in WSNs, namely anchor nodes, and unknown sensor nodes. Some sensor nodes have pre-determined, known positions, which are called anchors or beacons. However, unknown sensor nodes don't have those specifications. One of the most significant problems in WSNs is the localization of these unidentified sensor nodes for the location based service and plays an important role in different applications scenarios in WSNs [3].

There are some works about localization in WSNs which can be divided into two classes: range based and range free schemes which are different in the information used for localization. The range based schemes require either node to node distance or the angles for estimating positions. They need more sophisticated hardware to estimate node range such as time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA) and received signal strength indicator (RSSI) [4].

On the other hand, the range free localization also has a drawback that it is not able to estimate the exact point to point distance. Nevertheless, the simplicity of the hardware made range free methods very appealing and advantageous for localization in wireless sensor networks. Although the results in range free schemes are not as precise as the range based, the low cost and simpler estimates are important reasons that the range free method has increased popularity in the recent years [5].

Fuzzy Logic (FL) is a multivalued logic which permits intermediate values to be defined between conventional evaluations such as yes or no, high or low, true or false, which has two different meanings. In the narrow sense fuzzy logic is a logic system of an extension of multivalued logic. FL has difference in both substance and concept of traditional multivalued systems in the narrow description. On the other hand, in a wide sense FL is synonymous with the theory of fuzzy sets that theory relates to classes of objects with limitations [6].

In this paper, the Sugeno fuzzy [10] inference is used for simulation to estimate the location of each sensor nodes.

ISBN: 978-1-61208-207-3

The reason for using Sugeno Fuzzy inference is the membership functions which are more suitable in order to model the RSSI graph. We increase the number of Fuzzy membership function in order to better fit the RSSI graph versus distance. In fact, all the RSSIs are fed to the fuzzy system to achieve the weights to be used in centroid relation in order to estimate the location of the sensor nodes.

The range free method has different techniques to estimate the position of sensor nodes in a specific region. In this paper Centroid localization based on RSSI has been utilized. RSSI is used in the estimation of the distance between each anchor and the sensor node.

.

II. Centroid Localization

In centroid algorithms, the locations of the unknown nodes are estimate from the coordinates of their neighboring reference nodes. In fact, centralized localization is mainly based on transferring of the inter node ranging and connectivity data to a powerful central base station. After that, the computed positions are transferred back to the respective nodes.

The main advantage of the centralized localization method is that it omits the problem of computation in every node. The centroid localization scheme is simple and easy to implement. A simple centroid localization algorithm is needed to calculate a node's location based on the positions of many reference nodes which is simple but the estimation error might be high because of the simplicity of the centroid formula. However, using appropriate weights for the reference nodes decreases the localization error [7].

A. Fundamental Centroid

The range free algorithm based on proximity uses the location of anchor nodes (reference nodes) (x_i, y_i) to estimate the nearest unknown node [7]. The task of the centroid algorithm is, to take several nodes around the unknown nodes as polygon vertices and the unknown node as the centroid of the polygon, which is indicated in Figure 1.

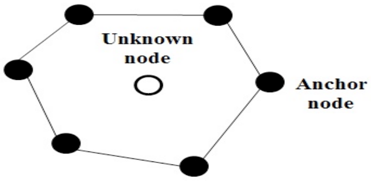


Figure 1. Centroid Localization Algorithm.

After receiving the message, the following relationship is used for estimating the coordinates of the unknown node:

$$(X_{est}, Y_{est}) = (\frac{X_1 + ... + X_N}{N}, \frac{Y_1 + ... + Y_N}{N}) \quad (1)$$

where (X_{est}, Y_{est}) indicates the estimated position of the sensor node and *N* is the number of the anchor nodes which is connected to the sensor node. This algorithm is simple but is not sufficient enough for estimating the unknown position. Therefore, the use of some weights on the reference nodes is required to solve this problem.

The method to improve the results of Eq. (1), where the anchor nodes are weighted in terms of their proximity to the sensor nodes, is given by this formula [8]:

$$(X_{est}, Y_{est}) = (\frac{x_1 w_1 + ... + x_n w_n}{\sum_{i=1}^{n} w_i}, \frac{y_1 w_1 + ... + y_n w_n}{\sum_{i=1}^{n} w_i}) \quad (2)$$

This method also has a weakness due to the choice of the weights $(w_1, w_2, ..., w_n)$ and the performance depends on the design of the weights.

Soft computing is an important tool to solve the problem of using low-cost, simple hardware and it is very appropriate for systems with uncertainties and nonlinearities. It is used to achieve tractability, robustness and low solution cost. Soft computing technique in fuzzy logic plays a crucial role in this paper [9].

III. Localization Using Fuzzy Logic

The fuzzy logic provides a distinct way to achieve control or classification in a system. This method is focused on what the system must do more than trying to model how it works and also it can concentrate on solving a problem more than the mathematical modeling of the system. The fuzzy logic is an area of research, which is fascinating and reaches a trade off between significance and exactness. Fuzzy logic is a reasonable way to map an input space to an output space where mapping is the starting point for everything [6].

One of the concepts in fuzzy logic is an if-then rule, which used in artificial intelligence that deals with fuzzy antecedents or consequents. Indeed, fuzzy logic solution is an interpretation of human thinking. On the other hand, FL can model nonlinear functions of optional complexity to a sufficient degree of exactness. Fuzzy logic is a simple way to model a multi input and multi output system [10].

IV. Simulation Results

Estimating the location of each sensor node is done by the centroid method. Hence, the weights are the main variable in the centroid relation, which are the outputs of the fuzzy system in the simulation. In fact, the Sugeno fuzzy system receives RSSIs as inputs to map the outputs, which are weights of each anchor node to the sensor node.

A WSN consists of sets of anchor nodes and sensor nodes with anchor nodes are located at known positions as $[(X_1, Y_1), (X_2, Y_2), ..., (X_N, Y_N)]$ and transmit signal strengths containing their respective locations.

The anchor nodes in this implementation are located at (0,0), (10,0), (10,10) and (0,10), where the coordinates are in meters. Several sensor nodes are distributed randomly in the specific region and receive signal strengths from the anchor nodes to estimate their location. The main responsibility of the sensor node is to compute the RSSI information sent by the anchor nodes.

The implementation is been done by Sugeno type fuzzy inference method. The Sugeno fuzzy inference is similar to the Mamdani method and the main difference between them is the membership functions where the output in the Sugeno method is constant or linear.

The input membership function of the Sugeno method is the RSS from the anchor nodes, which are decomposed into nine triangular membership function such as very very low (VVL), very low (VL), low (L), medium low (ML), medium (M), medium high (MH), high (H), very high (VH), very very high (VVH) that it shown in Figure 2. The input membership functions take value *$[RSS_{min}, RSS_{max}]$,* where RSS_{min} and RSS_{max} are the minimum and maximum RSS respectively, which are received by each sensor from each of the anchor node.

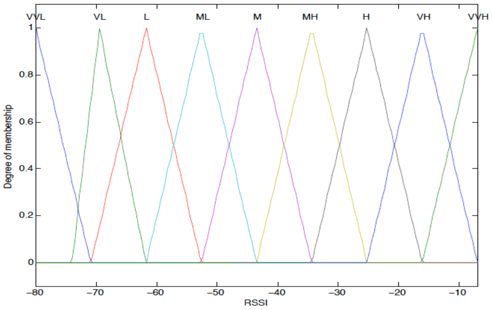


Figure 2. Input membership functions.

On the other hand, the output membership function of the Sugeno fuzzy inference is the weight of each anchor node for a given sensor node which takes a value $[0, w_{max}]$, where w_{max} is the weight with a maximum value of one. The output membership function distributes into nine linear functions such as VVL, VL, L, ML, M, MH, H, VH, VVH.

To find the range of output of each membership function, the logarithm of each RSSI at different distances should be mapped to linear variable between [0,1]. On the other hand, the RSSIs should be mapped among [0,1] in nine variable with weights.

The rules are considered for this Sugeno fuzzy method are in terms of the power of RSS. If the anchor node receives a high power from the sensor node, it indicates that the sensor node is near to the anchor node. On the other hand, if the senor node receives a low power, it shows that the sensor node is far from the anchor node. Table 1 shows the rules of Sugeno fuzzy system.

TABLE I. FUZZY LOGIC RULES

RULES	IF: RSSI IS	THEN: WEIGHT
RULE 1	V V LOW	V V LOW
RULE 2	V LOW	V LOW
RULE 3	LOW	LOW
RULE4	MEDIUM LOW	MEDIUM LOW
RULE 5	MEDIUM	MEDIUM
RULE 6	MEDIUM HIGH	MEDIUM HIGH
RULE 7	HIGH	HIGH
RULE 8	V HIGH	V HIGH
RULE 9	V V HIGH	V V HIGH

Figure 3 indicates the surface of the fuzzy system, which shows the weights corresponding to the RSSI values.

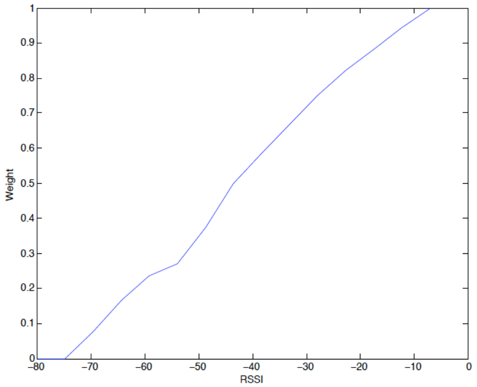


Figure 3. RSSI vs. Weight (Surface).

A. Calculating RSSI

The sensor nodes are distributed randomly in a square region with a side length of 10 metres. The first step is to estimate the RSSI by the following formula:

$$RSSI = -(10n\log_{10}(d) + alpha) \qquad (3)$$

where d is the distance of each sensor node to the anchor nodes, $n=3.25$ is path loss exponent, which may take different values because of ambient conditions may differ in different directions. Alpha is a constant and is the RSSI value of the sensor node that is located at 1-meter distance from the anchor node; so, in this paper, alpha is considered to be -40dBm for this implementation.

B. Using Localization Algorithm

The centroid method is the scenario considered in this paper. Therefore, for estimating the coordinates of the sensor nodes, the centroid formula given in Eq. (2) is used, where N is the number of connected adjacent reference nodes.

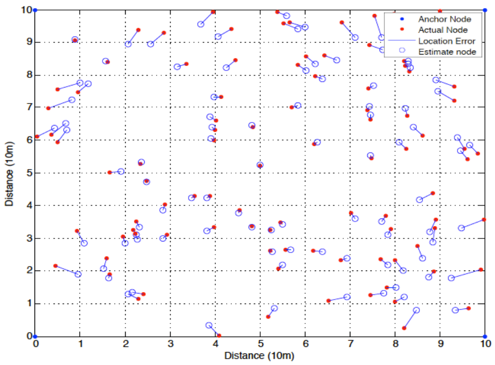


Figure 4. Simulation result of localization by Sugeno fuzzy method.

As seen in Figure 4, the region is 10 square meters, where the anchor nodes are located at (0,0), (10,0), (10,10) and (0,10). 100 sensor nodes are randomly deployed in the region. Each sensor node receives four different RSSI from each of the four anchor nodes, therefore, RSSIs reflect the distance of the sensor node to each of the anchor nodes. After estimating the RSSIs, each sensor node has four weights that are estimated by Sugeno fuzzy system.

The solid red dots shown in Figure 4 are the exact locations of the randomly deployed sensor nodes and the empty circles shown as connected to the solid circles are the estimated locations of the sensor nodes. The length of the line between them is the error of location.

C. *Performance Evaluation*

The location error between actual and estimated nodes is calculate by the following relation:

$$\textit{Location error} = \sqrt{(x_{est} - x_a)^2 + (y_{est} - y_a)^2} \qquad (4)$$

In order to estimate the position errors for all the estimated and actual nodes, the following relation is used

$$\textit{Average Location Error} = \frac{\sum\sqrt{(x_{est} - x_a)^2 + (y_{est} - y_a)^2}}{N} \qquad (5)$$

where *N* is the total number of sensor nodes.

After estimating the location of the sensor nodes in the simulation, Additive White Gaussian Noise (AWGN) is added to the RSSI with Signal to Noise Ratio (SNR) equal to 10. Figure 5 indicates the result simulation of localization with AWGN.

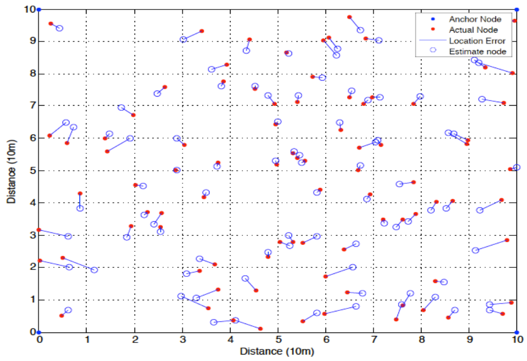


Figure 5. The location of sensors with AWGN.

Table 2 indicates minimum, maximum and average error locations of sensor nodes in both centroid method and fuzzy are shown. Hence, the result of centroid method error location comparing to the fuzzy is very high. For fuzzy, the result error location for both with and without AWGN also shown. The result of average errors in both areas are close to each other. According to this table and comparing the minimum and maximum of both results of error location in the areas, we see that this implementation has the highest accuracy in different environments.

TABLE II. THE COMPARISON RESULTS

	Min Error Location	Max Error Location	Average Error Location
Centroid Method without Fuzzy	0.10	4.35	2.95
Fuzzy	0.005	0.70	0.26
Fuzzy + AWGN	0.01	0.79	0.30

.

V. EXPERIMENTAL RESULTS

The experiment is done in a square region with 10 meters side length. The RSSIs taken from each node in this experiment have slightly different values compared to the RSSIs obtained in the simulation. Figure 8 shows the result of RSSIs for both simulation and experiment. The solid line in this figure indicates the result of RSSI simulated in Matlab^{TM} and the other line is the result of RSSI taken from the experiment (Figure 6).

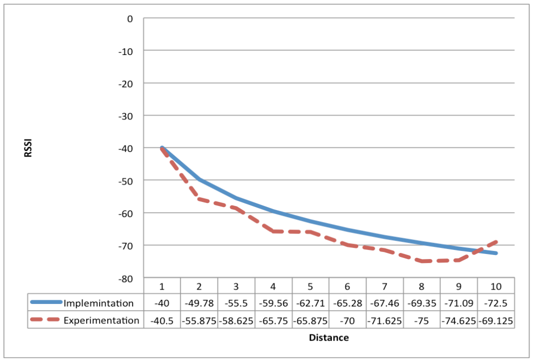


Figure 6. RSSIs in Simulation and Experimentation.

In the experiment, some nodes are located randomly in the region in order to obtain their RSSI from the anchor nodes. Each RSSI is then fed to the fuzzy system to get the corresponding weights.

This experiment is repeated 6 times for 6 different positions of the sensor nodes. Figure 11 shows the position of one of the random nodes located at (1,1).

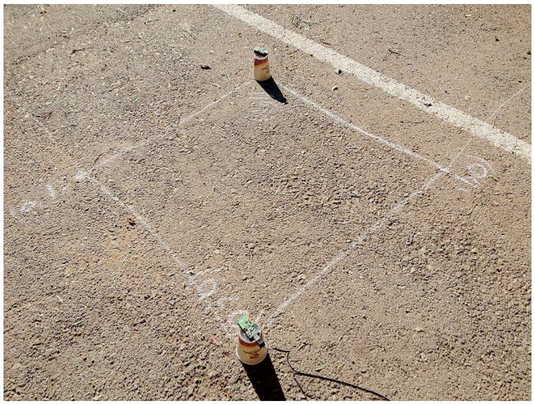


Figure 7. Sensor node in position (1,1).

As can be seen from Figure 7, the sensor node is closer to the first anchor (0,0), so it receives the highest RSSI in comparison with the other anchor nodes. The centroid relation for the sensor nodes is given in Table 3.

As can be seen from Table 3, the node at (1,1) receives the highest RSSI from the anchor node at (0,0), which is the nearest anchor node. On the other hand, that node receives the minimum RSSI from the anchor node at (10,10) so it is located at the farthest distance from that anchor node. The difference between the measured and simulated results is very low.

TABLE III. THE RESULTS OF THE EXPERIMENT. THE POWER VALUES ARE IN dbm AND THE LOCATION IN meters.

(X_a,Y_a)	RSSI from (0,0)	RSSI from (0,10)	RSSI from (10,0)	RSSI from (10,10)	(X_{est},Y_{est})	Error Location
(1,1)	-41	-71.5	-68.5	-75.5	(1.37,0.74)	0.45
(4,3)	-57	-69	-63	-68	(4.67,2.98)	0.67
(2,7)	-67	-65	-75	-70	(2.10,6.56)	0.44
(5,5)	-69	-71	-70	-68	(5.52,4.98)	0.52
(7.5,7.3)	-75	-68	-67	-55	(7.89,7.54)	0.46
(9,5)	-73	-74	-64	-65	(9.21,4.34)	0.69

Table 4 represents the comparison results of error location in both implementation and experimentation.

TABLE IV: COMPARISON RESULTS OF ERROR LOCATION

	Average Error in Simulation	Average Error in Experiment
S. Yun, J. Lee, W. Chung, E. Kim, S. Kim (2009)	0.78	0.80
V. Kumar, A. Kumar, S. Soni (2011)	0.59	0.94
Proposed Method in this Paper	0.26	0.53

It can be seen that, the result of error location obtained in this paper by Sugeno fuzzy is less than the other existing methods. This is due to the enhancement in the input membership functions in the Sugeno fuzzy system which has been increased in order to better mapping between input and output.

VI. CONCLUSION

The node localization is a big challenge in wireless sensor networks. The range free localization method is very simple and does not require a complicated hardware. The range free method has different techniques to estimate the position of a sensor node in a specific region. In this paper Centroid localization has been used. The estimation of each sensor node's location has been implemented by RSSI. The Sugeno fuzzy inference is used to simulate for estimating the location of each sensor node. The RSSIs are fed to the fuzzy system to compute the weights to be used in the centroid relation in order to estimate the location of the sensor nodes. The weights are the main parameter in the centroid relation, which are the outputs of the fuzzy system. The Sugeno fuzzy system receives RSSIs as inputs to map to the outputs, which are the weights of each anchor node with respect to the sensor node. The simulated results are checked with an experimental setup. The experiment is repeated 6 times for 6 different sensor nodes in the region. The agreement between the simulated and measured results is very good.

REFERENCES

[1] E. D. Elson, “A Bridge to the Physical World”, Journal of Sensor Networks, Springer, 2004.

[2] W. Su, Y. Sankarasubramaniam, E. Cayirci, and I. Akyildiz, "A survey on sensor networks," in Communications Magazine, 2002, pp. 112-114.

[3] U. Shaha, U.B Desai, S.N. Merchant, and M.M. Patil, "Localization in Wireless Sensor Networks using Three Masters," in International Conference on Personal Wireless Communications ICPWC, 2005.

[4] X. Li, Y. Shang, D. Ma, and H. Shi, "Cramer-Rao Bound Analysis of Quantized RSSI Based Localization in Wireless Sensor Networks," in International Conference on parallel and distributed systems (ICPAD’05), 2005.

[5] J. Heidemann, D. Estrin, and N. Bulusu, "GPS-less Low Cost Outdoor Localization for Very Small Devices," Personal Communications Magazine, vol. 7, pp. 28-34, 2000.

[6] D. H. Prade, “Fuzzy Sets and Systems: Theory and Applications”, New York, Academic Press, 1980.

[7] C. Huang, B. Blum, J. Stankovic, A. T. Abdelzaher, and T. He, "Range-Free localization schemes in large scale sensor networks," in International Conference on Mobile Computing and Networking (Mobicom), 2003.

[8] G. Sarigiannidis, “Localization for Ad Hoc Wireless Sensor Networks. Netherlands”, Technical University Delft, August 2006.

[9] L. Jaehun and W. Y. Sukhyun, "A soft computing approach to localization in wireless sensor networks," vol. 36, pp. 7552–7561, 2009.

[10] M. Sugeno, "Fuzzy measures and fuzzy integrals: a survey," Fuzzy Automata and Decision, pp. 89-102, 1977.

Review of Capacitive Atmospheric Icing Sensors

Umair Najeeb Mughal[1], Muhammad Shakeel Virk[2], Mohamad Yazid Mustafa[3]
High North Technology Center, Department of Technology
Narvik University College
Narvik, Norway
Email: unm@hin.no[1], msv@hin.no[2], myfm@hin.no[3]

***Abstract*—The application of capacitive sensing technique is widely distributed in different physical domains primarily because of the diversity in dielectric permittivity and due to its minimum loading error and inertial effects. Atmospheric ice is a complex mixture of water, ice and air which is reflected in its complex dielectric constant. There are many existing atmospheric icing sensors but only few are based on their complex dielectric permittivity measurements. This technique is very suitable because the capacitive variation in this mixture is due to the reorientation of water dipole in the electromagnetic radiation's oscillating field. Depending on the frequency, the dipole may move in time to the field, lag behind it or remain apparently unaffected. This variation is clearly reflected on the cole cole diagram which is a measure of the relaxation frequency. This paper is a review of some capacitive sensing technique in general but based upon dielectric variations and some existing capacitive based atmospheric ice sensing techniques. It is emphasized that the capacitive method proposed by Jarvenin provides maximum atmospheric icing parameters hence future atmospheric icing sensors may utilize the proposed technique with some modifications to further reduce the loading errors.**

Keywords-Atmospheric ice; Sensor; Polar molecule; Dielectric.

I. Introduction

A. Atmospheric icing

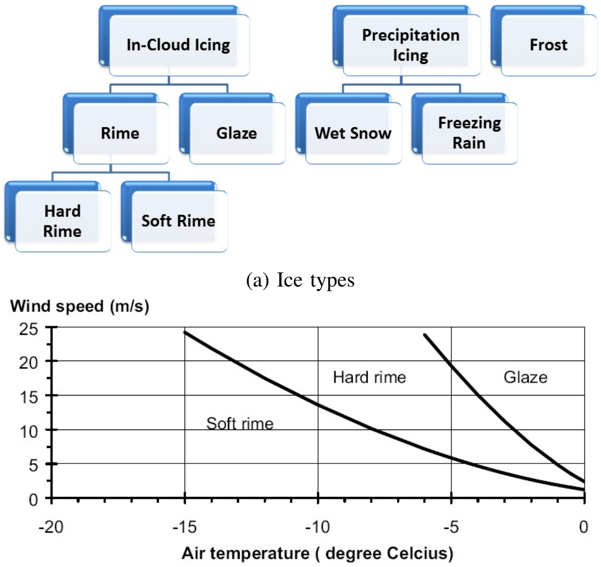
Atmospheric icing is the term used to describe the accretion of ice on structures or objects under certain conditions. This accretion can take place either due to freezing precipitation or freezing fog. It depends mainly on the shape of the object, wind speed, temperature, liquid water content (amount of liquid water in a given volume of air) and droplet size distribution (conventionally known as the median volume diameter).

The major *effects of atmospheric icing* on structure are the static ice loads, wind action on iced structure and dynamic effects.

Generally an icing event is defined as period of the time when the temperature is below 0°C and the relative humidity is above 95%. Ice accretion can be defined as, *any process of ice build up and snow accretion on the surface of objects exposed to the atmosphere'* [6]. Atmospheric icing is traditionally classified according to two different processes [6] and are shown in Fig. 1(a).

Table I: Typical properties of accreted atmospheric ice [5]

Type of ice	**Density (kg/m^3)**	**General appearance**	
		Colour	***Shape***
Glaze	900	transparent	evenly distributed/icicles
Wet snow	300-600	white	evenly distributed/eccentric
Hard rime	600-900	opaque	eccentric, pointing windward
Soft rime	200-600	white	eccentric pointing windward



(a) Ice types

(b) Ice as a function of wind speed and air temperature

Figure 1: Ice types and their dependence [6]

(a) Precipitation icing (including freezing precipation and wet snow),
(b) In-cloud icing (also called rime/glaze, including fog),

Fig. 1(b) shows the type of accreted ice as a function of wind speed and temperature. In this figure, the curve shifts to the left with the increasing liquid water content and with decreasing object size. A classification of atmospheric ice is shown in Table I.

B. Atmospheric icing sensors

A robust technique to detect icing and ice accretion rates has not yet been reported in the published literature. It is a challenging task to devise a measurement technique that can deal with both rime and glaze icing and can measure icing rate, load and duration without being affected by the icing event. Currently, all the ice detectors available are capable of measuring either one or both phenomenon such as detecting the icing event and measuring the rate of icing. As ice sensors can be integrated with ice mitigation systems, it is important for these sensors to deliver the necessary information timely enough so as to be able to operate anti-icing and de-icing mitigation strategies effectively. To distinguish between snow and ice can be considered to be an important factor for the determination of deicing power requirements. Hence, measurement of an icing event or related phenomena bounds a set of requirements which include the ability of a sensor/probe to detect icing with high sensitivity without being influenced by the icing incident. Icing measurement techniques can be classified into direct and indirect methods as follows:

Indirect Methods: The indirect methods of ice detection involve measuring weather conditions such as humidity, and temperature that lead to icing or detecting the effects of icing, for example, reduction in the power generated by the wind turbine, reduction in the speed of anemometers or measuring the variables that cause icing or variables that correlate with the occurrence of icing, such as cloud height and visibility [5] . Empirical or deterministic models are then used to determine when icing is occurring. Also Homola et. al. [7] have outlined five indirect measurement methods. The reduction in the speeds of anemometers method of Craig and Craig [2] and the noise generation frequency method of Seifert [14] are typical examples of indirect methods.

Direct Methods: The direct methods of ice and snow detection are based on the principle of detecting property changes caused by accretion such as mass, dielectric constants, conductivities, or inductance. Although Homola et. al. [7] outlined twenty four direct measurement methods but they still need to be more categorized for further exploration. The categorization of these direct methods can be,

1) Capacitive techniques
2) Microwave techniques
3) Inductance techiques
4) Ultrasonic techniques
5) Acoustic techniqes
6) Infrared techniques
7) Resonance techniques

II. Capacitive sensing technique - In general

From the above categories, the capacitive technique is the main focus of this review. The capacitance depends on the geometrical arrangement of the conductors and on the dielectric material between them, $C = C(\epsilon, G)$. For example, for a capacitor formed by n equal parallel plane plates having a geometry G depending upon area A, with a distance d between each pair, and an interposed material with a relative dielectric constant ϵ_r, the capacitance is

$$C = \epsilon_\circ \epsilon_r \frac{A}{d}(n-1)$$

where $\epsilon_\circ = 8.85pF/m$ is the dielectric constant for vacuum mentioned by Pallace and Webster [13]. Therefore, any measurand producing a variation in ϵ_r, A, or d will result in the change in the capacitance C and can be in principle sensed by that device.

A. Dielectric constant from electronic polarization

The electron orbiting a nucleus is like a harmonic oscillation with a natural frequency ω_o mentioned in Kao [9]. The dynamic equation can be defined as,

$$m\frac{d^2\Delta x}{dt^2} = -\gamma \Delta x - ZqF_{loc} \tag{1}$$

where Δx is the electrons displacement, m is the electron mass, q is the electronic charge, Z is the number electrons involved, F_{loc} is the local field acting on the atoms, and γ is the force constant. Also the natural oscillation frequency is given as $\omega_o = \sqrt{\frac{\gamma}{m}}$. Also oscillating electron is equivalent to an electric dipole and would radiate energy according to electromagnetic theory of radiation. This energy can be taken as a damping mechanism and $\beta\frac{dx}{dt}$ is a retarding force, hence our dynamic equation is,

$$m\frac{d^2\Delta x}{dt^2} + m\omega_o^2 \Delta x = -\beta\frac{dx}{dt} - ZqF_{loc} \tag{2}$$

From Bohr's Model, we have the potential of electron given as,

$$E = \hbar\omega_o = \frac{mq^4}{(4\pi\varepsilon_o)^2\hbar^2} \tag{3}$$

where $\hbar = \frac{h}{2\pi}$ and h is plank's constant. Also when $Z = 1$ we have electronic polarization, $\alpha_e = 4\pi\varepsilon_o R^3$ where R is radius of the ground state orbit of Bohr's atom.
Similarly, electronic susceptibility and dielectric constant is given as,

$$\begin{gathered} \chi_e = \tfrac{N\alpha_e}{\varepsilon_o} = \tfrac{N}{\varepsilon_o}\left[\tfrac{(Zq)^2}{m\omega_o^2}\right] \\ \varepsilon_r = 1 + \chi_e = 1 + \tfrac{N}{\varepsilon_o}\left[\tfrac{(Zq)^2}{m\omega_o^2}\right] \end{gathered} \tag{4}$$

B. Complex dielectric constant

When a time varying electric field is applied across a parallel plate capacitor with the plate area of one unit and a separation of d between the plates, then the total current is given by,

$$J_T = J + \frac{dD}{dt} = J + \varepsilon^* \frac{dF}{dt} \tag{5}$$

where J is the conduction current and ε^* is defined as complex permittivity which is introduced to allow for dielectric losses due to friction accompanying polarization and orientation of electric dipoles. This may be written as,

$$\varepsilon^* = \varepsilon - j\varepsilon^{'} = \left(\varepsilon_r - j\varepsilon^{'}_r\right)\varepsilon_o \tag{6}$$

where ε_r is dielectric constant and $\varepsilon_r^{'}$ is the loss factor. Also loss tangent is defined as, $\tan\delta = \frac{\varepsilon_r}{\varepsilon_r^{'}}$ where δ is loss angle. We can use the instantaneous energy absorbed per second per cm^3 is given by $J_T(t)F(t)$. Thus, on average, the amount of energy per cm^3 per second absorbed by the material is

$$W = \frac{\omega\varepsilon_r^{'}\varepsilon_o F_m^2}{2} \tag{7}$$

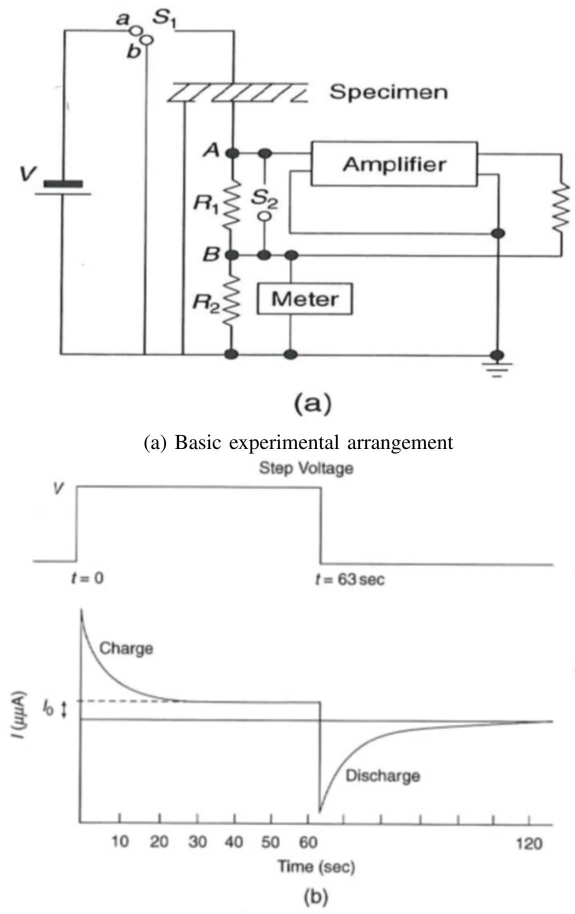
The discrete nature of matter, and the behavior and interaction of those particles, can be manifested through their response to time varying electric fields with wavelengths comparable the distances between the particles. To measure the dynamic response, we can use either use,

Time Domain Approach: We measure the time dependent polarization immmediately after the application of a step function electric field or we meausre the decay of the polarization from an initial steady state value to zero after the sudden removal of an initial polarizing field. This decay is generally referred to as *dielectric relaxation*.

Frequency Domain Approach: We mainly measure the dielectric constant at various frequencies of alternating excitation fields. From the viewpoint of measuring techniques, the time domain approach is simpler than the frequency domain approach, but from the viewpoint of data analysis, the time domain approach is more complex. However, both approaches should be intimately connected and should yield, in principle, the same results.

1) Dielectric relaxation - Time Domain Approach: This is a time domain approach which provides conspicuous information abouth the nonlinearity of the dielectric behavior simply by varying the amplitude of the applied step function held. Experimental arrangement for the measurements of the time domain response (i.e., the transient charging or discharging current, resulting from the application or the removal of a step DC voltage) is given in Fig. 2.

In this circuit, the switch S_1 has 2 positions: one for turning on the step DC voltage to start the flow of charging current, the other for short circuiting the specimen to allow the discharging current to flow after the specimen has been fully charged to a steady state level. The switch S_2 is used to short circuit R_1 to provide a path for surge currents for a very short period of time to protect the circuit; it also gives a chance to adjust the amplifier to a null position before recording the transient current. It is important to make the time constant of the amplifier which depends on the stray capacitance in shunt with R_1, much smaller than the time during which the transient current is flowing. The specimen has the guard and the guarded electrodes, the outer guard electrode being connected to ground to eliminate surface leakage currents from the specimen. The charging or discharging current is measured as a voltage appearing across R_1 by means of a DC amplifier. The voltage drop from point A to ground is made zero by a negative feedback in the amplifier circuit, which produces a voltage across R_2 equal and opposite to that across R_1 thus making the applied step voltage across the specimen only. The step voltage and the charging and discharging current as a function of time are also shown in Fig. 2(b) in which I_o is the steady DC compoent of the charging current and the width of the step voltage is 63 seconds.



(a) Basic experimental arrangement

(b) The step voltage and responses

Figure 2: Setup for the measurements of the charging and the discharging current from the application and removal of a step voltage [9]

2) Frequency Domain Approach: No material is free of dielectric losses and therefore no material is free of absorption and dispersion which reflects that no material is frequency independent ε_r and $\varepsilon_r^{'}$. Now, using Debye Equations for a varying electric field $F_m e^{j\omega t}$ we have the relationships as,

$$\varepsilon_r = \varepsilon_{r\infty} + \frac{\varepsilon_{rs} + \varepsilon_{r\infty}}{1 + \omega^2\tau_0^2} \tag{8}$$

$$\varepsilon_r^{'} = \frac{(\varepsilon_{rs} - \varepsilon_{r\infty})\,\omega\tau_0}{1 + \omega^2\tau_0^2} \tag{9}$$

$$\tan\delta = \frac{\varepsilon_r^{'}}{\varepsilon_r} = \frac{(\varepsilon_{rs} - \varepsilon_{r\infty})\,\omega\tau_0}{\varepsilon_{rs} + \varepsilon_{r\infty}\omega^2\tau_0^2} \tag{10}$$

Eq(s). 8, 9, 10 equations can also be written as,

$$\frac{\varepsilon_r - \varepsilon_{r\infty}}{\varepsilon_{rs} - \varepsilon_{r\infty}} = \frac{1}{1+\omega^2\tau_0^2} \tag{11}$$

$$\frac{\varepsilon_r^{'}}{\varepsilon_{rs} - \varepsilon_{r\infty}} = \frac{\omega\tau_0}{1+\omega^2\tau_0^2} \tag{12}$$

Now, the Eqn(s). 11, 12 are the parametric equations of a circle in the $\varepsilon_r - \varepsilon_r^{'}$ plane. By eliminating $\omega\tau_0$ from Eq. 11 and 12 we obtain,

$$\left(\varepsilon_r - \frac{\varepsilon_{rs} + \varepsilon_{r\infty}}{2}\right)^2 + \varepsilon_r^{'2} = \left(\frac{\varepsilon_{rs} - \varepsilon_{r\infty}}{2}\right)^2 \tag{13}$$

Only the semicircle of Eqn(s). 13 over which $\varepsilon_r^{'}$ is positive has physical significance. In this Argand Diagram shown in Fig. 3(b), frequency is not explicitly shown. The variation of ε_r and $\varepsilon_r^{'}$ due to the variation of ω is shown in Fig. 3(a) which illustrates schematically the typical dispersion behavior for polarization in the relaxation regime. Also, the Eq(s). 8, 9, 10 are based on the following assumptions for simplicity: the local field is the same as the applied field F; the conductivity of the materials is negligible; all dipoles have only one identical relaxation time τ_o. For more details on the mathematical description of the various forms of Debye relations for the detection of atmospheric ice, see Mughal et. al. [11].

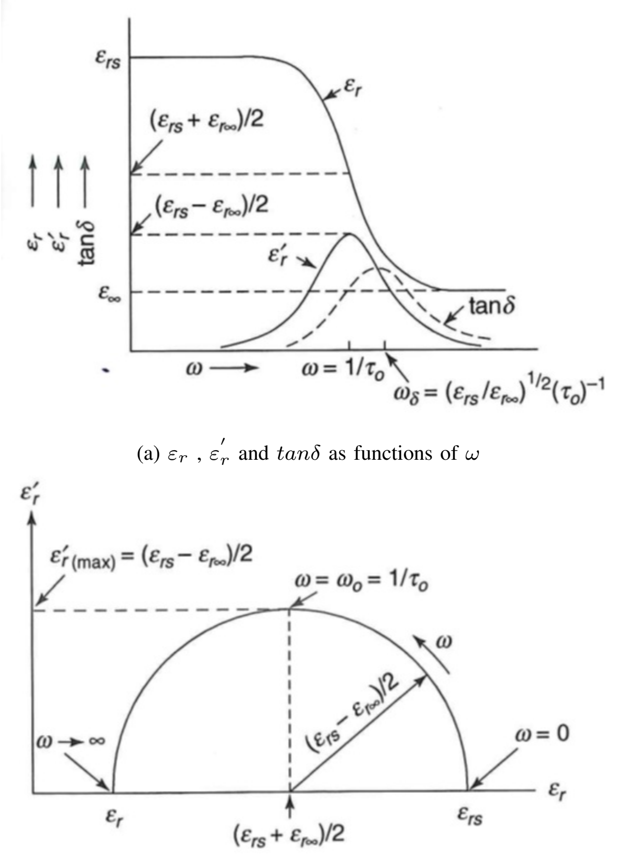
III. Capacitive icing sensors

Capacitive ice sensors generate an electric field to detect the presence of dielectric materials. Such electric field radiates outward around the probe and a dielectric material in close proximity of the field affects the measured capacitance, Mughal et. al. [10]. This attribute enables non-invasive measurements. In Tiuri et. al. [15], the results indicate that the complex dielectric constant is practically independent of the structure of snow. It is also mentioned that for dry snow, the dielectric constant is determined by the density and for wet snow, the imaginary part and the increase of the real part due to liquid water have the same volumetric wetness dependence. The static dielectric constants, ε_0 of both polycrystalline and single crystals of ice have been carefully determined Auty and Cole [1]. Also, application electrical properties to the measurement of ice thickness, temperature, crystal orientations are presented in Evanes [4]. Weinstein [16], Kwadwo [12] and Jarvinen [8] proposed three different capacitive based ice detection methods, Mughal et. al. [10], which are discussed in the following sections.

A. Capacitive ice detector by Weinstein

This ice sensor proposed by Weinstein [16] as given in Fig(s). 4a(a) can be used for the determination of the thickness of ice (22) on the outer surface (12) of an object independent of temperature and the composition of the ice (22). First capacitive guage (16), second capacitive guage (18), and the temperature guage (20) are embedded in embedding material (14) located within a hollow portion of outer surface (12). First capacitive guage (16), second capacitive guage (18), and temperature guage (20) are respectively connected to first capacitance measurement circuit (24), second capacitance measurement circuit (26), and temperature measuring circuit (28). The geometry of first and second capacitive guages (16) and (18) is such that the ratio of voltage outputs of first and second capacitive guages (24) and (26) is proportional to the thickness of ice (22), regardless of ice temperature or composition. This ratio is determined by offset and dividing circuit (29). First capacitance measuring circuit (24) and second capacitance measuring circuit (26) are connected to offset the dividing circuit (29). The output voltage V_{out} of this offset and dividing circuit (29) for ice conditions is determined by the relation,

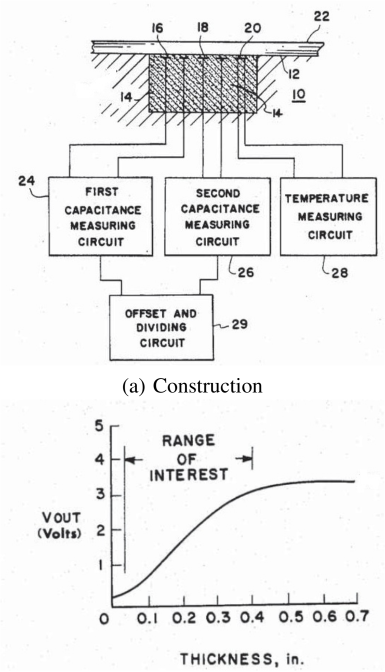
$$V_{out} = \frac{(V-V_o)_2}{(V-V_o)_1} \tag{14}$$



(a) ε_r , $\varepsilon_r^{'}$ and $tan\delta$ as functions of ω

(b) Argand diagram of $\varepsilon_r - \varepsilon_r^{'}$ relations for cases with one one relaxation time τ_o

Figure 3: Frequency domain approach for measurement of dielectric constants [9]


16
18
20
22
12
10
14
14
24
FIRST CAPACITANCE MEASURING CIRCUIT
SECOND CAPACITANCE MEASURING CIRCUIT
TEMPERATURE MEASURING CIRCUIT
26
28
OFFSET AND DIVIDING CIRCUIT
29
5
4
3
2
1
RANGE OF INTEREST
VOUT (Volts)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
THICKNESS, in.

(a) Construction

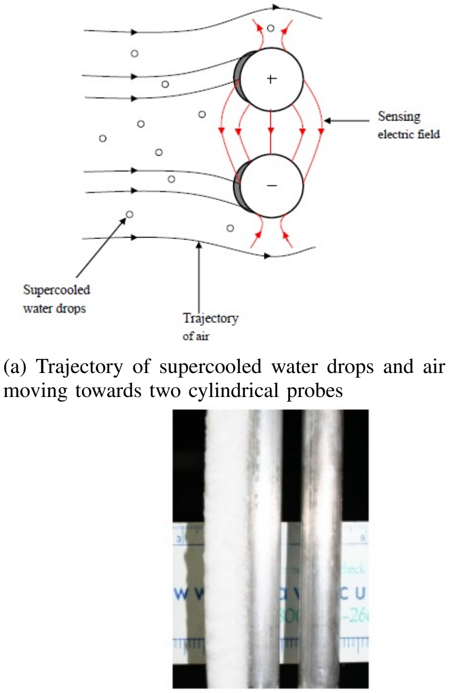
(b) Ratio of capacitance gauge as a function of thickness

Figure 4: Weinstein ice detector [16]

where V is the voltage output for the ice conditions and V_o is the initial voltage for no ice conditions. Subscripts (1) and (2) refer respectively to capactive measurements from first capacitance measuring circuit (24) and second capacitance measurement circuit (26). V_{out} is independent of both temperature and ice decomposition since both effects results in identical scaling factors for both $\frac{V-V_o}{2}$ and $V - V_{o1}$, thereby resulting no changes in Eq(s). 14. The variation of capacitance as a function of thickness is shown in Fig(s). 4a(b). *This sensor is capable of predicting ice and its thickness on a planar surface.*

B. Two cylinder capacitive icing probe

Kwadwo [12] has used two-cylinder probes to act as a capacitive ice sensor, based on the principle that as ice accretes on two electrically charged parallel-arranged cylindrical probes, the measured capacitance increases, while the resistance decreases. As the super cooled water droplets collide with the cylindrical probes and stick on the surface, they freeze and ice begins to grow as shown in Fig 5b. The accreted ice affects the electric field generated by the electrically charged cylindrical probes resulting in an increase in the capacitance due to the higher dielectric constant of the accreted ice compared to air. The electric field originating from the polarization charges on the surface of the ice partly shields the external electric field generated by the charged cylindrical probes leading to a reduction in the overall electric field. The overall voltage decreases simultaneously, because the electric field is directly proportional to the voltage. The resistance between the cylindrical probes is large at the start of the icing event because of the air gap between the cylinders. However, as ice builds up on the cylindrical probes, the air gap between the cylindrical probes decreases and the resistance begins to decrease exponentially. The rate of decrease is sensitive to the presence of water on the surface of the ice formed on the cylindrical probes and this phenomenon is used to distinguish between different types of ice .



;

(a) Trajectory of supercooled water drops and air moving towards two cylindrical probes

(b) Ice formation at the windward side of the cylindrical probes

Figure 5: Cylindrical capacitive sensor

C. Total impedance and complex dielectric property ice detection system

In this sensor, Jarvinen [8] used the method for detecting the presence and the accretion of ice by first measuring the properties of the contaminant layer overlying the ice sensor. The contaminant layer's temperature, thermal conductivity and variation of total impedance versus ice sensor electrical excitation frequency are measured. The complex dielectric property subsystem monitors the dielectric property locus in dielectric space as the excitation frequency is varied from near dc to higher frequencies (using Cole-Cole plot) and compares the measured results for magnitude and shape with laboratory property data taken at the same temperature and stored in the processor. It double checks using external ice (based upon the complex dielectric measurements) sensor whether it is ice or rain water or deicing fluid or snow. If the measured results form a semicircular shaped locus of dielectric properties in complex dielectric space during the frequency scan and those measurements are also determined to be in agreement with on board stored laboratory ice data, ice is confirmed to be present. The presence of ice is also confirmed if a particular vector can be constructed from the measured data taken at a single preselected excitation frequency and found to have a vector angle in agreement with the vector angle from stored laboratory results taken at the same measurement conditions. In addition, complex dielectric measurement algorithms identify whether cracks, flaws or voids or increased electrical conductivity exist in the ice covering and sensor from their effects on the shape and size of the measured complex dielectric locus or from the length of the vector at the pre selected frequency. The presence of flaws, cracks or voids or enhanced electrical conductivity are determined from the values for the low frequency and high frequency intercepts and the value for diameter of the complex dielectric locus if these values are found to differ from those calculated for ice based on stored ice data. These differences, if found to exist, are used to correct the initially chosen ice thickness value based on the assumption of normal ice: ice with no flaws, cracks or voids or higher electrical conductivity. For more details on the mathematical principle of this type of sensing technique see Mugal et. al. [11].

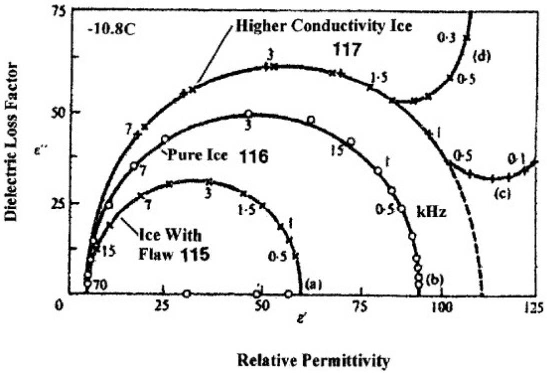


Figure 6: Cole Cole plot for glaze ice experimental results

This sensor is capable to identify the presence of ice, its thickness, thickness rate, and can identify glaze ice, rime ice, rain water, deicing fluid, snow or air. In addition it is redundant in confirming the icing event.

IV. Conclusion and Future Work

The mere existence of a permanent dipole moment in water provides structural information about the molecule. It is found that the dielectric variations in different types of ice can be very effective in finding the parameters such as ice type, ice thickness and icing rate. The patent of Jaravinen [8] can be considered as a benchmark as it is able to sense all the above paramters, hence the direct approach mentioned by Homola et. al. [7] is able to deliver maximum information. Also due to the variation in response of ice and snow by varying the electrical field; the application of Cole-Cole Diagram for complex dielectric constant of snow and ice is adequately proved. It is stressed that a simulation study and analytical study on the capacitive variations of atmospheric ice be carried out to compare the numerical and theoretical results with the experimental variations. These results can further be utilized for the determination of atmospheric ice type and measurement of its rate and thickness. A hybrid measurement technique may also be considered in future for robust results.

Acknowledgements

The work reported in this paper was partially funded by the Research Council of Norway, project no. 195153/160 and partially by the consortium of the ColdTech project - Sustainable Cold Climate Technology.

References

[1] Auty R. P. and Cole R. H., “Dielectric properties of ice and solid D_2O”, Journal of Chemical Physics, Vol. 20, Issue 8, pp. 1309-1314, 1952.

[2] Craig D. F. and Craig D. B., “An investigation of icing events on haeckel hill”, Proceedings of Boreas III Conference, Finland, 1995.

[3] Eisenberg D. and Kauzmann W. “The Structure and Properties of Water”, Oxford University Press, 1969.

[4] Evanes S., “Dielectric properties of ice and snow - a review”, Journal of glaciology, Vol. 5, Issue 42, pp. 773-792, 1965.

[5] Fikke S., et. al., “Cost 727 - Atmospheric icing on structures; measurement and data collection on icing”, ISSN 1422-1381, MeteoSwiss, 2007.

[6] Foder H. F., “ISO 12494 - Atmospheric icing on structures and how to use it”, Proc. of the 11th International Offshore and Polar Engineering Conference, ISBN 1-880653-51-6, June 2001.

[7] Homola M. C., Nicklasson P. J., and Sundsbo P.A., “Ice sensors for wind turbines”, Cold Regions Science and Technology, 46: pp. 125-131, 2006.

[8] Jarvinen P. O., “Total impedience and complex dielectric property ice detection system”, US Patent 7439877, 2008.

[9] Kao K. C., “Dielectric phenomena in solids”, Elsevier Academic Press, ISBN 0-12-396561-6, 2004.

[10] Mughal U. N., Virk M. S., and Mustafa M. Y., “Review Of Atmospheric Ice Detection Techniques”, unpublished.

[11] Mughal U. N., Virk M. S., and Mustafa M. Y., “Dielectric Based Sensing Of Atmospheric Ice”, 38th International Conference on Application of Mathematics in Engineering and Economics, AIP Conference Proceedings, in press.

[12] Owusu K. P., “Capacitive probe for ice detection and accretion rate measurement: proof of concept”, Masters thesis report submitted at University of Mannitoba, 2010.

[13] Pallas-Areny R. and Webster J. G., “Sensors and signal conditioning”, 2nd Edition, John Wiley and & Sons, 2001.

[14] Seifert H., “Technical Requirements for Rotor Blades Operating in Cold Climates”, Proceedings of the BOREAS II conference, Pyhatunturi, Finland, 2003.

[15] Tiuri M., Sihvola A., Nyfors E., and Hallikaiken M., “The complex dielectric constant of snow at microwave frequencies”, IEEE Journal of Oceanic Engineering, Vol. 9, Issue 5, pp. 377 - 382, 1984.

[16] Weinstein L. M., “Ice sensor”, US Patent 4766369, 1988.

Radiofrequency Transceiver for Probing SAW Sensors and Communicating through a Wireless Sensor Network

Christophe Droit, Jean-Michel.Friedt
SENSeOR SAS
Besançon FRANCE
{christophe.droit,jmfriedt}@femto-st.fr

Gwenhael Goavec-Mérou, Gilles Martin, Sylvain Ballandras,
Karla Breschi, Julien Bernard, Hervé Guyennet
Time & frequency (TF)
Computer science & complex systems (DISC)
FEMTO-ST institute UMR 6174, Besançon FRANCE
gwen@trabucayre.com, {gilles.martin,ballandr}@femto-st.fr,
karla.jimenez_ramirez@edu.univ-fcomte.fr,
{julien.bernard,herve.guyennet}@femto-st

***Abstract*—A radiofrequency transceiver is used for the dual purpose of probing surface acoustic wave resonator sensors and communicating the resulting measurements through a digital wireless link. Thus, the radiofrequency hardware exhibits a complementary use of short range probing of piezoelectric sensors subject to harsh environmental conditions, and long range digital communication through a wireless sensor network. The demonstration is performed in the 434 MHz European Industrial, Scientific and Medical band, yielding half-duplex communication ranges well beyond 100 m.**

***Keywords*—transceiver; radio communication; 434 MHz ISM band; SAW resonator; temperature sensor.**

I. Introduction

Piezoelectric surface acoustic wave (SAW) transducers [1] have been widely used as passive (no local energy source) wireless sensors [2], [3], [4], [5], [6]. Despite apparent similarities with silicon based RadioFrequency IDentification tags (RFID), SAW devices underlying physical principles differ vastly, requiring only linear processes in the electromagnetic to mechanical sensing wave conversion and thus improved interrogation ranges [7]. SAW sensors are probed using active interrogation units operating on principles similar to RADAR. Amongst the classes of SAW devices, two main families include the resonators (narrowband device, characterized by a resonance frequency dependent on a physical quantity under investigation) [8] and the delay lines (wideband devices, characterized by a propagation delay dependent on a physical quantity under investigation) [9], [10], [11]. While the latter strategy often requires fast electronics and large data storage memories (typical time constants are in the hundreds of MHz bandwidth, with typical 40-ns long pulses), probing resonator can be as simple as an embedded frequency sweep network analyzer probing the reflection coefficient of the transducer. Because of the wireless link, a pulsed mode RADAR provides improved isolation (and hence interrogation range): the typical solution is a pulsed-mode frequency-sweep RADAR. Multiple references in the literature discuss the implementation of dedicated hardware for probing such devices [12], [13], [14], [15], either as Fourier-transform based [16] by emitting a wideband pulse and identifying the frequency of the signal returned by the sensor, or sweeping a narrowband pulse [17] for identifying the frequency at which the sensor returns most power (meeting the resonance frequency condition, hence allowing for the resonator to store energy which is then released during the listening step).

On the other hand, the widespread availability of wireless communication interfaces provides embedded chips with most functionalities needed for probing a SAW sensor: tunable radiofrequency source, power amplifier, low noise amplifier on the reception stage, I/Q demodulator and low pass filters. Our aim is to use such a transceiver not only for its original purpose of transmitting digital data through a wireless link, but also for probing the frequency-dependent response of SAW resonators. Hence, we select transceivers which provide the I and Q demodulated analog outputs, and analyze the needed signal processing steps for extracting the relevant information. All operations will be restricted to the European 434 MHz Industrial, Scientific and Medical (ISM) band: the same hardware is first configured to probe locally (range 0.1-2 m) a SAW sensor, and then reconfigured for sharing the acquired data through a digital wireless link with a sink in charge of storing the data and sharing them through the Internet.

II. Hardware selection

Because we aim at processing the analog signal returned by the sensor, access to the raw I and Q outputs of the receiver stage of the transceiver is mandatory. Due to increased requirement of compacity and low pin count, most radiofrequency transceivers only provide digital interfaces to the user. We have identified 3 suppliers of radiofrequency transceivers potentially compatible with our need: Semtech XE1203F, Maxim MAX7203 and Melexis TH7122. Because the latter is already used in a commercial product [18] in a dual chip (separate emitter and receiver) configuration which is hardly satisfactory,

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-207-3

we have focused on the former reference which provides a tunable frequency source with 500 Hz frequency step, I/Q analog outputs, and most significantly a fast (pin triggered) switching from emitter to receiver mode capability. This last point is significant: since the time constant of a resonator operating at frequency f_0 is $Q/(\pi f_0)$ with Q the quality factor of the device, the power loss is 8.7 dB/time constant during the exponential decaying response of the sensor. Typical values at $f_0 = 434$ MHz are $Q \in [8000 - 1000]$ so that typical time constants for sampling the returned signal are in the 6-7 μs.

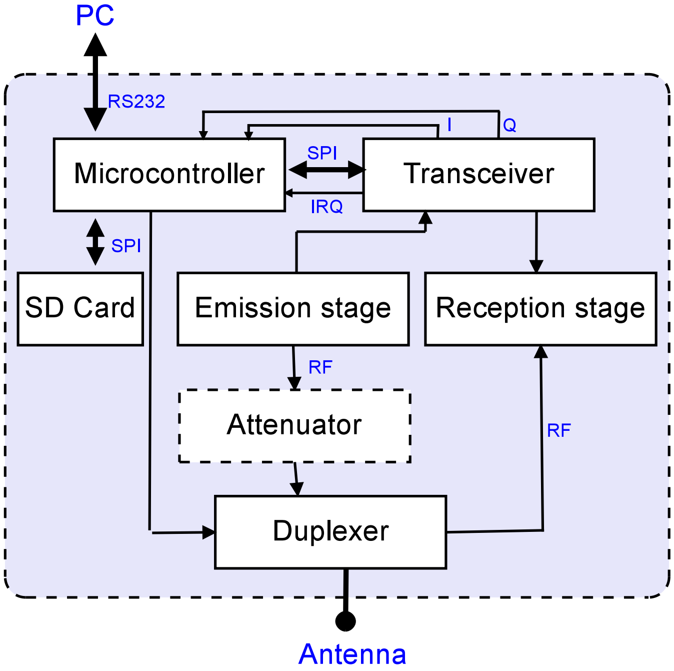


Fig. 1. Synoptic description of the card. The microcontroller configures components depending on the actions chosen. The two modes available are communication mode (fixed 434 MHz carrier, 4800 bit/s) and a SAW probing mode. In the latter operating mode, the radiomodem scans the ISM frequency band step by step so as to detect the resonant frequencies of resonators. The resonance frequency difference is returned as the quantity representative of the temperature measurement. This measurement result is also stored on a SD card. The external radiofrequency duplexer is mandatory for improved isolation between the emission and reception stage and thus an interrogation range of the SAW sensor reaching 1 m.

Using an integrated transceiver for probing SAW devices, although attractive in terms of integration, footprint and power consumption, is challenging since the signal processing performed at the output of the mixer of the reception stage are hardly documented. Nevertheless, the frequency-dependent response of SAW sensors is characterized using such a hardware, and the frequency at which the returned power is maximum is identified by sampling simultaneously the I and Q output using a dual analog-to-digital converter embedded in a ST Microelectronics STM32 microcontroller (Fig. 1). Using this approach, the total number of integrated circuits needed to design a SAW resonator reader is restricted to 4 : a XE1203 transceiver, a microcontroller providing fast (>1 Msamples/s A/D conversion), a fast radiofrequency duplexer for switching from emission to reception stages in a monostatic antenna configuration, a digitally programmable radiofrequency attenuator for improved measurement range dynamics and possibly a RS232 to USB (FTDI FT232RL) converter (Fig.1). Reducing the number of active circuits aims at reducing the total power consumption: the STM32 based on an ARM Cortex M3 architecture exhibits reduced deep-sleep mode power consumption of less than 4 μA. All dedicated radiofrequency chips are powered by General Purpose Input Output (GPIO) pins of the microcontroller while the radio frequency transceiver provides deep sleep mode capabilities. Measured power consumptions are summarized in table I.

Operation mode	Consumption (mA)
RF digital communication	140
Probing SAW resonators	80
Standby mode microcontroller and transceiver in reception mode	22.3
Standby all components	1.2

TABLE I
POWER CONSUMPTION OF THE SAW INTERROGATION UNIT BASED ON A XE1203F RADIOMODEM AND STM32 MICROCONTROLLER, DEPENDING ON THE OPERATING MODES. THE SUPPLY VOLTAGE IS 3.3 V.

Although the time constant of the resonator discharge is 6 to 7 μs, Fig. 2 exhibits a significant returned signal for more than 25 μs after switching the duplexer from emission to reception positions. This signal is interpreted as the impulse response of the low pass filters located on the I and Q channels output, after the internal radiomodem mixers (here set to a cutoff frequency of 200 kHz). The oscillations observed on the I and Q outputs are the result of mixing a fixed frequency returned by the sensor (resonance frequency) with the emitted tunable frequency source. Following this time dependent characterization of the output of the I and Q channels, the practical use of these information only requires a single measurement by the analog to digital (A/D) converters of the STM32, selected at a time 20 μs after switching the duplexer position.

Once the relevant data from the SAW sensors are recorded, the radiofrequency transceiver is reconfigured to operate in its original purposes, namely wireless digital data communication. While the current demonstration focuses on a point to point communication towards a sink configured as a sink (constantly listening for incoming messages), dynamic signal routing is under investigation using the MAC layer provided by the TinyOS executive environment under the Collection Tree Protocol (CTP) routing protocol [19]. Because failure of the sink (either due to operating system crash or overload, malicious attacks when connected to the Internet, or power failure) is considered as the weakest link in the dissemination of the measurement data, the embedded SAW reader has been fitted with a Secure Digital (SD) mass storage medium for keeping a local record of all emitted sentences (Fig. 4). Because most users request the ability to recover the stored data on widely available Personal Computers, a FAT-based data storage was selected as a tradeoff between a filesystem compatible with low power microcontroller, while still widely available on most commonly used operating systems. Thus,

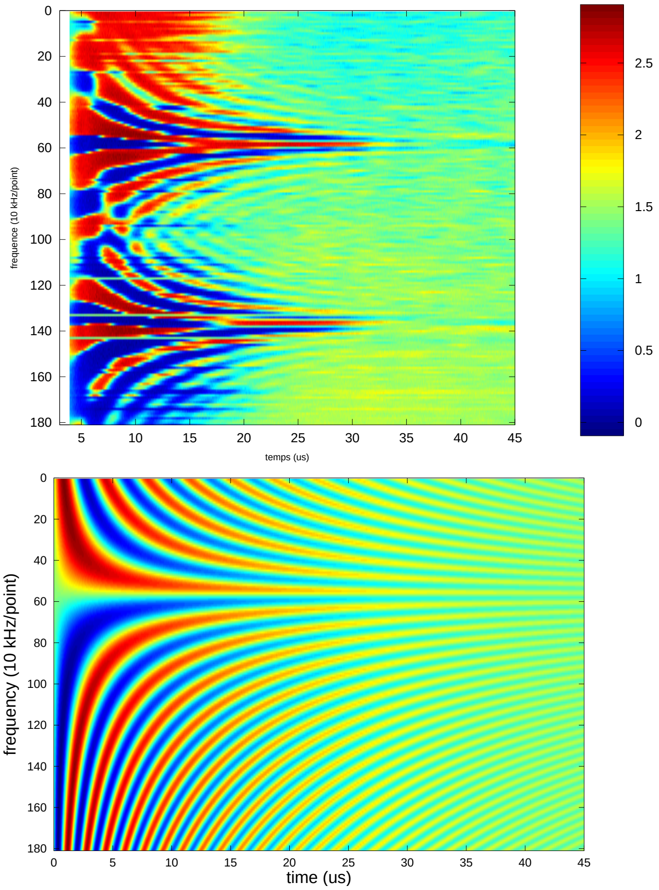


Fig. 2. Top: experimental measurement of the time-dependent output of the I output of the XE1203F transceiver when a dual resonator SAW sensor is connected to the antenna output. Bottom: simulation of the returned signal as $\sin(2\pi(f-f_0)t)$ with f_0 the fixed resonance frequency of the sensor and f the frequency emitted by the transceiver (ordinate), hence demonstrating the the signal returned by the sensor is at the natural frequency of the resonator. The optimum signal to noise ratio was identified with a unique sampling by the microcontroller A/D converter 20 μs after the duplexer was switched from emission to reception positions.

the EFSL library was ported to the STM32 platform for this purpose.

All measurements are differential: the sensor is made of two resonators in parallel, one exhibiting a strong frequency dependence with temperature and the other one a turnover temperature within the operating range. Using this approach, each measurement requires a sampling duration of 33ms, including programming the transceiver, recording the two values (I and Q) from the analog to digital converter for each sampled frequency (128 samples in the 1.7 MHz wide ISM band), applying a cross-correlation algorithm through a fast Fourier transform to measure the frequency difference between both resonances, and transmitting the data through the wireless link as well as storing a copy on SD card. Digital data communication is performed at the bandwidth of 4800 bits/s for improved immunity to noise and extended communication range. SAW resonators hardly provide enough information to allow for both measuring a physical quantity and identification: since a resonator is characterized by only two parameters (resonance frequency and quality factor), the only available means for identifying multiple sensors located within interrogation range of a transceiver is by using frequency multiplexing. Despite not being compliant with ISM radiofrequency emission regulations, the transceiver can synthesize frequencies in the 434 ±8 MHz range, far beyond the ISM band, compatible with reading up to 32 resonances.

Although the (SAW) sensor does not require local power, the interrogation unit is battery powered. In order to extend the operating duration of the sensor network, deep sleep mode is active most of the time with only intermittent wakeup sequences to probe the sensor and transmit data to the sink. Each SAW reader is identified by a unique, user defined, 32-bit address used as pattern during the data transmission (hardware feature of the XE1203F chip).

III. Operating modes

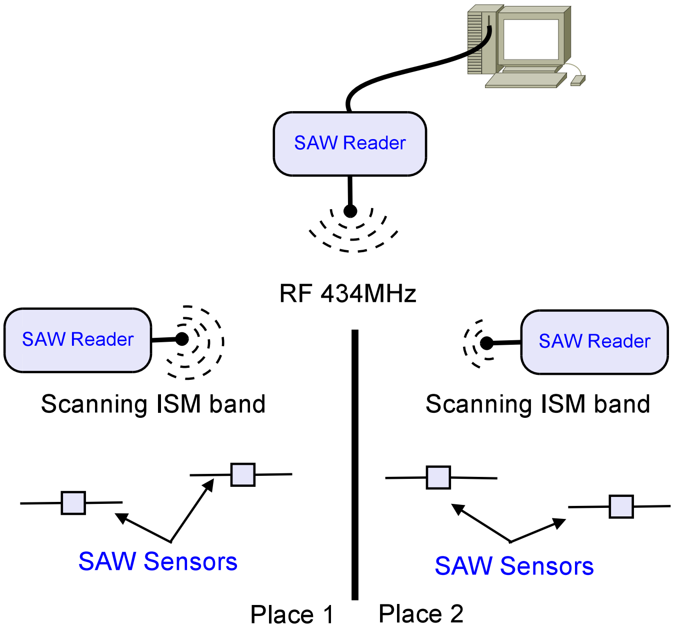


Fig. 3. Three SAW readers are used in this model of the wireless sensor network considered here: one reader acts as the sink and triggers the data transfer (master) from the other readers used for probing SAW sensor responses (slaves). Each slave reader is associated with nearby SAW sensors: after receiving the command for performing a measurement, the SAW reader probes the nearby resonator resonance frequencies and transmits the result through a wireless digital link.

Two operating modes have been implemented:

- one reader is configured as the sink, constantly listening for incoming data, and connected to a Personal Computer for data storage and transfer. In this case, the radiomodem transceiver is used in its default operating mode, namely digital data transmission. The other readers, spread on the field within a digital communication range of about 100 m, are configured in standby mode and periodically wake up to probe the nearby SAW sensor (interrogation range of $\simeq$1 m). Once the SAW sensor properties (resonance frequency and signal power) are recorded on the remote system, data are transmitted to the sink (Fig.3). One challenge in scalability of this simple approach is that readers are assumed *not* to wake up simultaneously in order to avoid interferences on the radiofrequency link. Furthermore, the digital link is half duplex, leading to potentially significant limitations in the extension of this approach to a fully distributed, multi-hop wireless network protocol. This issue is under investigation by porting the current low level (C-language based) implementation of the communication algorithms to the TinyOS executive environment, targeted at providing the MAC layer and associated communication protocols. Although the periodic wake up of each node hardly qualifies this implementation as a deployed Wireless Sensor Network (WSN), power consumption is optimized since the readers deployed in the field spend most of their life in deep sleep mode, yielding significant power consumption reduction.

- The second approach is based on a master/slave communication protocol, in which the sink (master) requests measurements from the readers deployed on the field (slaves). Again the multi hop protocol is not implemented, but here all nodes are constantly in receive mode, potentially acting as routers of the incoming messages to nodes located further away from the sink. The drawback is that although the microcontroller is in sleep mode (wake up by a hardware interrupt generated by the radiomodem), the radiofrequency transceiver exhibits significant power consumption even in receive mode, reducing the life expectancy in a battery powered application.

IV. Experimental results

One example of harsh environment [20] where battery powered sensors exhibit a limitation is buried sensors: once the sensor is installed in concrete or buried in soil, access for maintenance or battery replacement is no longer an option. Thus, our experimental demonstration is performed on a SAW sensor buried 30 cm deep in soil (fig.4). Such a device has been installed for more than 4 years with no dedicated packaging other than a standard micro-electronics 5 mm$\times$ mm ceramic packaging, with neither drift nor signal loss despite direct contact of the soldered dipole antenna (enameled wires) and the sensor with soil [21]. Alternative technologies include distribued measurements along a buried optical fiber exhibiting Brillouin backscatter [22], or if infinite life expectancy is not mandatory, battery powered systems with extended life expectancies have been used [23], [24].

The measurement standard deviation is 2 kHz: considering that the dual resonator sensor difference frequency dependence with temperature is 2500 Hz/^{o}C, the observed temperature variations between day and night is about 4 ^{o}C. The temperature measurement resolution is 1 ^{o}C.

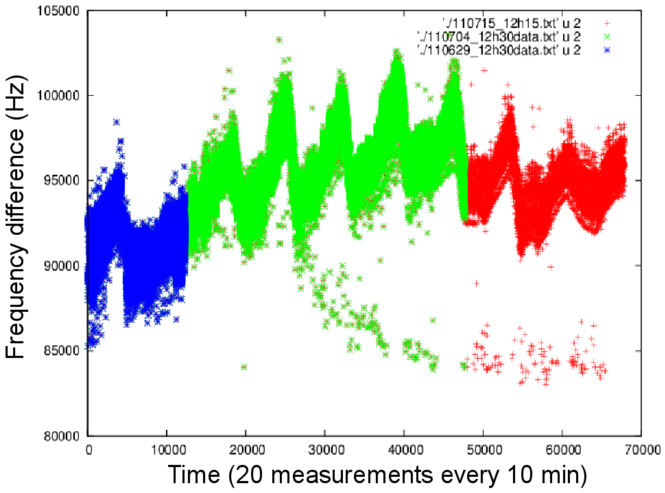


Fig. 4. 10-day temperature record of a SAW sensor buried 30 cm deep in soil: 20 measurements are stored on a FAT formatted SD card every 10 minutes and transmitted through a digital radiofrequency link. The various colors indicate successive manual recoveries of the data stored on the SD card.

V. Conclusion

While the use of SAW as passive sensors interrogated through a wireless link has demonstrated unique operating conditions in harsh environments, the widespread deployment as part of a wireless sensor network is here investigated by reconfiguring the same digital data transmission transceiver for probing SAW resonator properties and hence the associated physical quantity under investigation. A practical demonstration of temperature measurement with local data storage on a non-volatile medium and real time data transmission to a remote sink connected to a personal computer is performed. Thus, the complementarity of the approach is emphasized: a short (0.1 to 1 m interrogation range) interrogation distance of the passive sensor located in a harsh environment, coupled with long range wireless digital data transmission.

References

[1] R. White and F. Voltmer, “Direct piezoelectric coupling to surface acoustic waves,” *Applied Physics Letters*, vol. 7, no. 12, pp. 314–316, december 1965.

[2] W. Buff, F. Plath, . Schmeckebier, M. Rusko, T. Vandahl, H. Luck, F. Möller, and D. Malocha, “Remote sensor system using passive SAW sensors,” in *IEEE Ultrasonics Symposium*, 1994, pp. 585–588.

[3] A.Pohl, F.Seifert, L.Reindl, G.Scholl, T. Ostertag, and W.Pietsch, “Radio signals for SAW ID tags and sensors in strong electromagnetic interference,” in *IEEE Ultrasonics Symposium*, Cannes, France, 1994, pp. 195–198.

[4] H. Scherr, G. Scholl, F. Seifert, and R. Weigel, “Quartz pressure sensor based on SAW reflective delay line,” in *IEEE Ultrasonics Symposium*, San Antonio, TX, USA, 1996, pp. 347–350.

[5] L. Reindl, G. Scholl, T. Ostertag, C. Ruppel, W.-E. Bulst, and F. Seifert, “SAW devices as wireless passive sensors,” in *IEEE Ultrasonics Symposium*, 1996, pp. 363–367.

[6] P. Hartmann, “A passive SAW based RFID system for use on ordnance,” in *IEEE International Conference on RFID*, 2009, pp. 291–297.

[7] C. Hartmann and L. Claiborne, “Fundamental limitations on reading range of passive IC-based RFID and SAW-based RFID,” in *IEEE International Conference on RFID*, Gaylord, Texan Resort, Grapevine, TX, USA, March 2007.

[8] J. Beckley, V. Kalinin, M. Lee, and K. Voliansky, “Non-contact torque senseor based on SAW resonators,” in *IEEE Int. Freq. Control Symposium and PDA Exhibition*, New Orleans, LA, USA, 2002, pp. 202–213.

[9] A. Pohl, R. Steindl, and L. Reindl, “The “intelligent tire” utilizing passive SAW sensors – mesurement of tire friction,” *IEEE Transactions on Instrumentation and Measurement*, vol. 48, no. 6, pp. 1041–1046, december 1999.

[10] W. Bulst, G. Fischerauer, and L. Reindl, “State of the art in wireless sensing with surface acoustic waves,” *IEEE Transactions on Industrial Electronics*, vol. 48, no. 2, pp. 265–271, April 2001.

[11] J. H. Kuypers, M. Esashi, D. A. Eisele, and L. M. Reindl, “2.45 GHz passive wireless temperature monitoring system featuring parallel sensor interrogation and resolution evaluation,” in *IEEE Ultrasonics Symposium*, 2006, pp. 1453–1458.

[12] X. Q. Bao, W. B. an V.V. Varadan, and V. Varadan, “SAW temperature sensor and remote reading system,” in *IEEE Ultrasonics Symposium*, Denver, CO, USA, 1987, pp. 583–585.

[13] A. Pohl, G. Ostermayer, and F. Seifert, “Wireless sensing using oscillator circuits locked to remote High-Q SAW resonators,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 45, no. 5, pp. 1061–1168, september 1998.

[14] A. Stelzer, S. Schuster, and S. Scheiblhofer, “Readout unit for wireless SAW sensors and ID-tags,” in *International Workshop on SiP/SoC Integration of MEMS and Passive Components with RF-ICs*, 2004.

[15] G. Bruckner, A. Stelzer, L. Maurer, J. Biniasch, L. Reindl, R. Teichmann, and R. Hauser, “A high-temperature stable SAW identification tag for a pressure sensor and a low-cost interrogation unit,” in *11th International Sensor Congress (SENSOR)*, Nuremberg, Germany, 2003, pp. 467–472.

[16] M. Hamsch, R. Hoffmann, W. Buff, M. Binhack, and S. Klett, “An interrogation unit for passive wireless SAW sensors based on Fourier transform,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 51, no. 11, pp. 1449–1456, november 2004.

[17] J.-M. Friedt, C. Droit, G. Martin, and S. Ballandras, “A wireless interrogation system exploiting narrowband acoustic resonator for remote physical quantity measurement,” *Rev. Sci. Instrum.*, vol. 81, p. 014701, 2010.

[18] D. Beier and W. Meierfrankenfeld, “Method for temperature measurement in a household appliance,” *U.S. Patent 7 358 464*, April 15 2008.

[19] R. Fonseca, O. Gnawali, K. Jamieson, S. Kim, P. Levis, and A. Woo, “The collection tree protocol (CTP),” Online at www.tinyos.net/tinyos-2.x/doc/html/tep123.html, last accessed 08/08/2012, 2007.

[20] R. Fachberger, G. Bruckner, R. Hauser, and L. Reindl, “Wireless SAW based high-temperature measurement systems,” in *IEEE International Frequency Control Symposium and Exposition*, 2006, pp. 358–367.

[21] J.-M. Friedt, T. Rétornaz, S. Alzuaga, T. Baron, G. Martin, T. Laroche, S. Ballandras, M. Griselin, and J.-P. Simonnet, “Surface acoustic wave devices as passive buried sensors,” *Journal of Applied Physics*, vol. 109, no. 3, p. 034905, 2011.

[22] T. Parker, “A fully distributed simultaneous strain and temperature sensor using spontaneous brillouin backscatter,” *IEEE Photonics Technology Letters*, vol. 9, no. 7, pp. 979–981, 1997.

[23] J. Hart, K. Rose, K. Martinez, and R. Ong, “Subglacial clast behaviour and its implication for till fabric development: new results derived from wireless subglacial probe experiments,” *Quaternary Science Reviews*, vol. 28, pp. 597–607, 2009.

[24] J. Hart and K. Martinez, “Environmental sensor networks: a revolution in the earth system science ?” *Earth Science Reviews*, vol. 78, pp. 177–191, 2006.

Distributed Multi-Head Clustering for People-Centric Sensor Networks

Kevin Wiesner, Markus Duchon, Michael Dürr
Mobile and Distributed Systems
Ludwig-Maximilians-Universität München (LMU Munich)
Munich, Germany
Email: {kevin.wiesner, markus.duchon, michael.duerr}@ifi.lmu.de

***Abstract*—The emergence of powerful sensor-equipped smartphones led to a new form of people-centric sensing networks (PCSN), in which users collect sensor information via their mobile phone. This kind of mobile sensing allows for large-scale data collection on low costs but introduces new challenges: PCSNs need to cope with continuously contributed data and keep transmission and energy costs for users at a minimum in order to reach large-scale participation. In this paper, we propose a Distributed Multi-Head Clustering (DMHC) algorithm that aims at resolving these problems by forming sensing clusters with different roles for participating nodes. We conducted simulations to evaluate DMHC and our analysis shows that it significantly reduces mobile network traffic and user costs without introducing too much overhead.**

***Keywords*-*People-centric sensing*; *Mobile phone sensing*; *Clustering*.**

I. Introduction

Mobile phone technology has recently undergone a rapid change: Improvements in computation, storage, and wireless communication lead to a spread of powerful mobile devices such as smartphones and tablets. A recent trend is the integration of sensing capabilities into the latest generation of mobile devices. Currently available phones come with built-in accelerometers, gyros, location, audio, and image sensors. We expect that in the future even more sensing hardware will be integrated into mobile phones. For instance, Nokia proposed a future mobile phone[1] that is equipped with sensors for monitoring the environment, the user's health, and the current weather. With this development mobile phones evolve from standard phones, intended for personal communication only, to ubiquitous sensing devices that are globally distributed.

These devices could be applied to form a new kind of sensor network, so-called *people-centric sensing networks* [2] (also referred to as *mobile phone sensing* [3] or *mobile crowdsensing* [4]), where people serve as carriers for mobile phone-based sensors. People-centric sensing networks (PCSN) allow for large-scale global data collection and real-time information display. They could be used, for instance, to monitor environmental pollution, temperature, or noise intensity of urban areas. Even though PCSNs are related to wireless sensor networks (WSN), there are significant differences between those two types of sensor networks (cf. [5]). The main advantage of PCSNs is that data can be collected on a large-scale with automatically deployed, consumer-paid sensor nodes. This new kind of real-time data collection opens up new opportunities for services and applications. However, it also entails several new problems: One major challenge is to process enormous amounts of data contributed by users. Especially for network operators, millions of continuously transmitting mobile phones would lead to a serious challenge. People-centric sensing will not be accepted if data measurements congest the mobile network. At the same time, it is desirable to upload as much data as possible to improve the quality of information. Thus, methods are needed that reduce mobile network traffic without reducing data quality.

In this paper, we propose an algorithm that aims at resolving this problem. Our Distributed Multi-Head Clustering (DMHC) algorithm relieves the mobile network by forming sensing clusters. Within a cluster, collected sensor data is exchanged via ad hoc communication and uploaded in an aggregated form by the clusterhead. To compensate for the overhead introduced by forming the cluster and to minimize energy costs for users, DMHC selects only a subset of nodes as sensingheads, which are required to collect and transfer measurement data to the clusterhead. The election of clusterhead and sensingheads is based on remaining energy levels, communication costs, and capabilities of the nodes. Our analysis shows that DMHC significantly reduces mobile network traffic and user costs while generating only negligible overhead.

The remainder of this paper is organized as follows. In Section II, the problem is formulated and requirements are specified. Section III discusses related work. In Section IV, we present our DMHC algorithm, followed by our simulation results in Section V. Finally, we present our conclusions and future work in Section VI.

II. Problem description and requirements

In this paper, we explore the problem of clustering PCSNs. Clustering, in this context, means partitioning its nodes into a set of *clusters* of geographically co-located nodes. Within each cluster, nodes assume one of the following roles (cf. Figure 1): *Clusterheads* are responsible for collecting measurement data within the cluster and transmitting these in aggregated form to a server responsible for storing the

ISBN: 978-1-61208-207-3

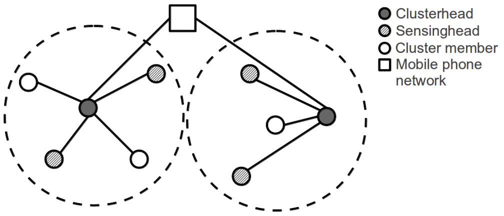


Figure 1. People-centric sensing network with multi-head clusters

collected data. *Sensingheads* are responsible for recording the measurement readings and transmitting them to the clusterhead. *Cluster members* have no specific function to fulfill, but might be requested to become a sensinghead if more measurements are needed. Each cluster consists of one clusterhead, one or more sensingheads, and possibly some cluster members. If there are too few nodes within a cluster, the clusterhead may additionally assume the sensinghead role; thereby minimal clusters consist of one node being clusterhead and sensinghead at the same time.

We want to limit the geographic extent of clusters in order to obtain measurements of nearby sensors at comparable locations. Thus, we aim for 1-hop clusters and so each node should have a link to the clusterhead. This implies that measurements of nodes that are within communication range are "comparable". As the geographical variance of measurements strongly depends on the sensor type (e.g., temperature or CO_2), we assume that the communication range is automatically adapted (by varying the transmission power) in order to meet the sensor type spatial specifics. In case of parallel measurements with different sensor types, the range needs to be adapted according to the sensor type with the highest spatial variance.

Furthermore, the clustering algorithm should fulfill the following requirements:

- *Mobility-Adaptive*: The clustering algorithm needs to cope with continuous node mobility.
- *Distributed*: To allow for scalable sensing networks, the clustering algorithm should be distributed.
- *Energy-Efficient*: Due to limited energy resources, clustering should use as little power as possible.

III. Related Work

There is a lot of research work related to people-centric sensing. Most work focuses on approaches and techniques that enable data collection with mobiles phones ([6]–[8]), but the potential communicational overload caused by these continuous sensing approaches is often neglected.

However, in the field of wireless sensor networks (WSN) and mobile ad hoc networks (MANET), efficient communication is an important research question. To achieve scalability, nodes are often grouped into clusters. For this process, various clustering algorithms have been proposed ([9], [10]). Our DMHC is based on Distributed Mobility-Adaptive Clustering (DMAC) proposed by Basagni [11], which enables clustering for scenarios with mobile nodes. Several surveys provide a detailed overview of clustering algorithms ([12], [13]). In contrast to our work, most existing clustering algorithms only distinguish between clusterheads and cluster members.

An orthogonal approach to the cluster-based concept is to reduce communication costs by eliminating redundant sensing data transmissions with the help of a prediction-based algorithm [14]. Data is only transmitted when it deviates from the predictions and if it changes the statistic with a high probability. The problem of this approach is that applied prediction models have to be adapted and optimized for each type of sensor data, if significant reduction is to be achieved. This is a complementary concept to ours and could be applied in addition to our clustering algorithm.

Another aspect of our approach is the reduction of energy costs, which we achieve with different roles per cluster. Several other approaches for energy efficient mobile sensing have been proposed. Wang et al. [15] presented a framework that only powers a minimum set of sensors and uses appropriate sensor duty cycles to achieve energy efficiency. Priyantha et al. [16] proposed a sensing architecture where sampling and processing of sensor data is offloaded to a dedicated low-power processor. As these approaches typically focus on energy-efficiency per node, they could also be integrated into our proposed clustering algorithm.

To the best of our knowledge, our approach is the first that allows for multi-head cluster partitioning for people-centric sensing networks.

IV. Distributed Multi-Head Clustering

In this section, we describe a distributed multi-head algorithm that sets up sensing clusters in PCSNs. The algorithm consists of two phases: (A) the set-up and maintenance of clusters (including the election of clusterheads) and (B) the election and initiation of sensingheads.

A. Cluster set-up and maintenance

The set-up and maintenance is mainly based on DMAC. DMAC partitions nodes of a mobile network into clusters in a distributed manner by applying a weight-based criterion. Our DMHC is an extension and adaption of DMAC for people-centric sensing networks. Messages, node weights, and procedures used in our proposed DMHC are explained in the following.

1) Clustering messages: In order to form a cluster, nodes need to be aware of neighboring nodes. This is achieved by using periodic broadcasts, so called *PeriodicClusteringMessages* (PCM). A PCM contains a node's ID, its remaining energy (i.e., its battery charge level), and its current communication costs. The latter are a combination of its *costs per transmission* (*cpt*) and its accumulated *total* *costs* for previous transmissions (*tc*). Due to the reception (or the absence) of PCMs, the status of links to neighboring nodes can be detected.

Besides PCMs, nodes use two types of messages for the cluster set-up: *CH* messages indicate that the sender has assumed the clusterhead role and are broadcasted in order to reach all neighboring nodes. Nodes in the vicinity overhearing the *CH* can join this cluster. If a node receives multiple *CH* messages, it joins the cluster with the bigger weight (cf. Section IV-A2). This can be indicated by sending a *Join* message to the clusterhead. *Join* messages are broadcasted as well, but have to be directed, i.e., they have to contain the clusterhead's ID in order to indicate the cluster they want to join. As *Join* messages may indicate a change from of cluster, they have an impact on the previous cluster as well and thus are also processed by the former clusterhead. In addition, *Join* messages include a set of sensing capabilities S (e.g., $S = \{Temp, CO_2\}$) that indicates, which sensors are provided by the joining node.

2) Node weights: In our approach, we adapted DMAC's concept of applying a weight-based criterion to allow for distributed clustering. However, DMAC does not specify the determination of the nodes' weight. In DMHC, the weight is calculated based on factors that are highly relevant for the deployment in people-centric sensing networks, namely communication costs and remaining energy levels.

In order to avoid excessive costs for users caused by continuous data transmissions, DMHC selects the node with the lowest communication costs (i.e., *min(cpt+tc)*) as clusterhead. In case communication costs are equal for two nodes, the one with the higher remaining energy (*re*) level receives the bigger weight. If the previous factors do not yield a distinct clusterhead, the node with the lower ID is chosen.

3) Clustering procedures: In order to respond to the previously specified messages, DMAC specifies several procedures. Those procedures are run by each node locally.

- *Init()*: The init() procedure is called whenever the node has no associated clusterhead. This may happen in two situations: (1) If the node has just joined the network (e.g., when it has just been switched on) and thus obviously cannot be member of a cluster yet. (2) If a node has lost its clusterhead. *Init()* determines whether there is a neighboring node with a bigger weight than itself. If this is the case, it joins that cluster, otherwise, it will become a clusterhead.
- *LinkFailure(u)*: If a node's connection to another node *u* gets lost (recognized through the lack of PCMs), it checks whether itself or *u* have had the clusterhead role. If itself is the clusterhead, it removes node *u* from the cluster members. If node *u* has been the clusterhead, the node restarts the *Init()* procedure to find a new clusterhead. Otherwise, the link failure has no direct impact on the node and is ignored.
- *NewLink(u)*: If a node receives a *PCM* of a new neighbor *u* and *u* is a clusterhead with bigger weight, the node affiliates with *u*.
- *OnReceivingCH(u)*: If a node receives a *CH* message, which indicates that *u* is a clusterhead, the node affiliates with this cluster if *u* has a bigger weight.
- *OnReceivingJoin(u,z)*: If a nodes receives a *Join* message, which indicates that *u* wants to affiliate with the cluster of *z*, the node has to check if *u* has just left its own cluster or wants to join it (i.e., $node = z$). If z is no clusterhead, it ignores the incoming *Join*. Node *u* learns about the failed association with the next PCM received from *z*.

B. Sensinghead election

If possible, there should be multiple sensingheads in a cluster in order to improve the robustness and compensate for faulty measurements of individual nodes. In order to keep the energy consumption as low as possible for the users, too many redundant measurements should be avoided. Thus, a trade-off between redundancy and energy saving is needed.

1) Election process: DMHC solves this by selecting only a specified fraction (*sensinghead ratio* ρ) of participating nodes as sensingheads. To ensure robust measurement results for small clusters, a minimum number of sensingheads υ can be specified. If a new node joins a cluster with n nodes, the sensinghead election is triggered if

$$(|Sensingheads| < \upsilon) \vee (\frac{|Sensingheads|}{n+1} < \rho). \quad (1)$$

Next, a priority class P_j is calculated that takes account of the capabilities of the new node j:

$$P_j = \max_{s \in (S_j \bigcap S_{Req})} (n - \sum_{k=1}^{n} |\{s\} \bigcap S_k|) \quad (2)$$

where S_j is the set of sensing capabilities of the joining node j, and S_{Req} denotes the set of capabilities required for the sensing tasks in the network. If $S_j \bigcap S_{Req} = \emptyset$, P_j is set to 0; nodes in this priority class are ignored in the following steps. The other priority classes (P_1,...,P_{n-1}) sort the nodes in such a way that nodes with scarce capabilities within the cluster (i.e., a sensor type that only a small subset of cluster members possesses) get into a lower priority classes. The idea behind this is to chose those nodes only for sensing tasks that require these scarce capabilities and that only those nodes can fulfill.

For the sensinghead election, we used two possible approaches in DMHC: The first one is called *sequential sensinghead election* (SSE) and is based on the sequence of arriving and leaving nodes. The first nodes joining a cluster are selected as sensingheads until the sensinghead ratio ρ is reached. From then on, the clusterhead checks for each new node whether ρ is still met. If the sensinghead ratio drops below ρ due to the newly arrived node, this node becomes a sensinghead. If a sensinghead leaves the cluster, for instance due to an association with another cluster, the cluster member that joined first is selected as a new sensinghead. The second approach, called *highest remaining energy* (HRE), considers the remaining energy level as the decisive factor. Every time a new node joins a cluster, the clusterhead compares the energy level of the newly joined node with that of the current sensingheads. If the new node has a higher energy level, it becomes a sensinghead and the former sensinghead with the lowest remaining energy level becomes an ordinary cluster member.

2) Sensinghead messages: Since nodes need to know whether they are supposed to conduct measurements or not, the clusterhead has to inform them about their role within the cluster. For this reason we introduce two new types of messages: *RevokeMeasurement (RM)* and *MeasurementRequest (MR)* messages. When joining a cluster, a new node assumes to be a sensinghead by default. *RevokeMeasurement* messages are sent by the clusterhead to inform the receiver that he is not a sensinghead anymore. *MeasurementRequest* messages are used to indicate that nodes should start measuring. MR messages are only sent after link failures or *HRE* sensinghead elections.

V. Evaluation

We conducted simulations to evaluate the performance of DMHC. In this section, we will first describe the simulation setup, followed by the presentation of the simulation results.

A. Simulation setup

For our simulations we used the JiST/SWANS simulation environment [17], which allows the simulation of large-scale wireless networks. The size of the simulation area was set to 5×5km. In this area the mobile phone nodes, varying from 1-100 nodes, were moving around. Nodes were randomly distributed, and mobility was modeled by using the Random Waypoint Model with speeds between 1 and 6 m/s and a pause time of 10 seconds. Each run simulated a period of 6 hours and was repeated 50 times. For the wireless communication, we facilitated the built-in wireless LAN (WLAN) simulation components and employed the free-space model using a standard configuration for the WLAN communication (transmission strength: 15 dBm, antenna gain: 1dB). For the sake of simplicity, we assumed $(S_n)_{n \in N} = S_{Req}$ for sensing capabilities of all nodes N.

B. Mobile network transmissions

We first evaluated the amount of mobile network transmissions by comparing the naive approach without clustering, in which all nodes conduct measurements and transmit the collected data themselves, to our cluster-based approach. It is obvious that our algorithm reduces the network traffic, as only clusterheads communicate via the mobile network, instead of all mobile nodes without clustering. However, we analyzed, which node densities result in clusters that are large enough to significantly reduce mobile network traffic.

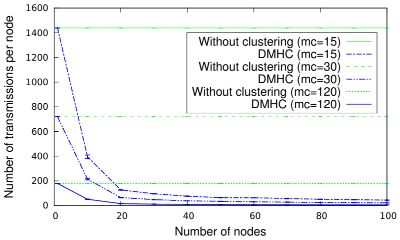


Figure 2. Normalized average number of mobile network transmissions

We ran simulations with different measurement cycles (*mc*={15s, 30s, 120s}). Figure 2 illustrates the normalized average number of transmissions, i.e., the total number of average transmissions in relation to the number of nodes. DMHC significantly reduces the network transmissions compared to the naive approach, even for low node densities. With 10 nodes only, we achieve a network traffic reduction of approximately 70%, and from 40 nodes on even more than 95% reduction. These results lead to the conclusion that DMHC makes sense even in less densely populated areas and can help to significantly reduce the network traffic imposed by people-centric sensing.

C. Communication costs

DMHC considers the communication costs during the clusterhead election. Ideally, in case of a high amount of data flatrate users, each cluster consists of at least one node with zero communication costs.

In our analysis, we evaluated the impact of the penetration rate of flatrate users on the average communication costs per node. Therefore, we ran simulation trials where 5%, 15%, and 25% of nodes had a data flatrate and were not charged for transmitting data. For the remaining nodes, each transmission via the mobile phone network was counted as a "charged transmission". The communication costs can then derived by including the actual costs per data transmissions (*charged_transmissions* $*$ *cpt*). For these trials, sensingheads conducted measurements with *mc*=15s, and PCM messages were broadcasted with a PCM cycle (*pc*) of 3-4 seconds.

The results are illustrated in Figure 3. In trials without clustering, the average communication costs drop according to the percentage of flatrate users, as only those nodes do not contribute to the overall communication costs. In trials with DMHC, the flatrate penetration has a relatively low impact compared to the cost reduction introduced by simply clustering participating nodes. The average communication costs for high-density settings become very low, as the probability of having a user with a data flatrate within each cluster obviously increases with larger cluster sizes. The

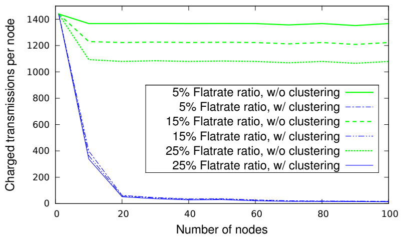


Figure 3. Number of charged transmissions per node

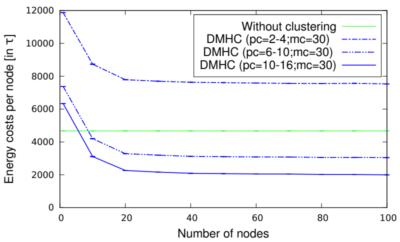


Figure 4. Energy costs for Scenario 1

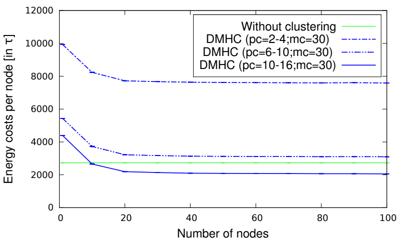


Figure 5. Energy costs for Scenario 2

results show that DMHC significantly lowers the costs and hereby provides a basis for a large-scale user participation.

D. *Energy costs*

We consider energy costs as the sum of energy used for mobile network transmissions, ad hoc transmissions, and sensor measurements. For those energy consuming tasks, the amount of task occurrences were analyzed in three different simulation trials for varying ratios of PCM and measurement cycles. We specified a high, medium, and low PCM-to-measurements ratio. In the high PCM-to-measurements ratio, PCMs were sent every 2-4 seconds (i.e., with a *pc* of 2-4s) and measurements were collected every 30 seconds. In this setting, a lot of ad hoc messages were exchanged in comparison to the amount of conducted measurements. In the medium and low PCM-to-measurements settings, PCMs were broadcasted every 6-10 and 10-16 seconds respectively, while measurements were conducted with the same rate (*mc*=30s). We made assumptions for relative energy costs for the above mentioned energy consuming tasks and considered two scenarios. In *Scenario 1*, mobile network transmissions consume six times more energy than ad hoc transmissions, based on the findings in [18]. Further, we assumed that the energy consumption of sensor measurements is low in comparison to energy used for transmissions (e.g., for temperature sensors). In *Scenario 2*, we assumed lower relative energy costs for mobile network transmissions (factor 3 compared to those of ad hoc transmissions), but also assumed slightly higher energy costs for sensor measurements. In Table I, the energy costs for the mentioned scenarios are listed, specified in relation to an energy cost unit τ.

Table I
ENERGY COST RATIOS FOR SCENARIOS

	Scenario 1	Scenario 2
Mobile network transmissions	6τ	3τ
Ad hoc transmissions	1τ	1τ
Sensor measurements	0.5τ	0.8τ

The results of Scenario 1 show that for trials with medium and low PCM frequency, energy costs are lower than without DMHC from 10 nodes onwards (see Figure 4). Only the setting with a very high frequency of periodic broadcasts exceeds the energy costs of the naive approach. Figure 5 shows the results for Scenario 2. The energy costs of DMHC exceed those of the naive approach in the setting with a medium PCM frequency. However, the additional energy overhead is relatively small and might be acceptable, if communications costs are lowered significantly instead.

E. *Number of measurements*

We evaluated the impact of the sensinghead ratio ρ and the minimum sensinghead number υ on the actual amount of measurements. In a first step (*S1*), we simulated three settings with different sensinghead ratios $\rho = \{10\%, 25\%, 50\%\}$ and a constant sensinghead minimum of $\upsilon = 2$. In a second step (*S2)*, we varied the sensinghead minimum $\upsilon = \{2, 3, 5\}$ for a constant $\rho = 10\%$. For these trials, *mc* was set to 15 seconds, and *pc* to 3-4 seconds.

The results from *S1* show that the number of measurements rapidly converges (see Figure 6). From about 30 nodes on, the amount of measurements per node remains stable, which shows that the effect from introducing sensingheads can also be useful in low-density settings. As the sensinghead ratio ρ specifies the minimal ratio, the number of measurements per node converges to an amount slightly higher than indicated, i.e., 58%, 17%, and 8% above ρ. The results from *S2* lead to similar conclusions (see Figure 7). Although υ has a significant impact for low-density settings, the number of measurements drops very fast in all trials. A stable average is reached from 30-40 nodes onwards.

F. *Ad hoc overhead for sensinghead election*

In the last analysis, we compared the ad hoc overhead of *SSE* and *HRE*. The main part of the overhead arises from PCMs sent out by each node. The number of those messages is the same for both approaches. The difference lies in the amount of non-periodical messages, thus we focused solely on ad hoc messages required for the pure cluster formation and maintenance (i.e., *CH*, *Join*, *RM*, and *MR*). The results (Figure 8) show that SSE performs slightly better. This is due to the fact that HRE re-determines all sensingheads every time a node joins or leaves the cluster. However, the difference of both approaches is relatively small compared to the overall ad hoc overhead.

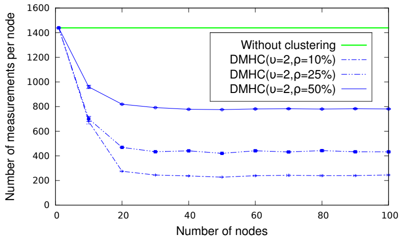


Figure 6. Varying sensinghead ratios (*S1*)

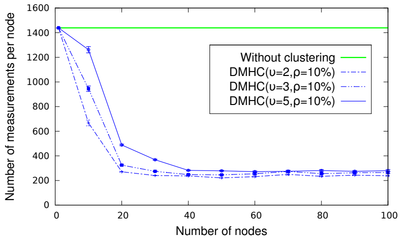


Figure 7. Varying min. sensingheads (*S2*)

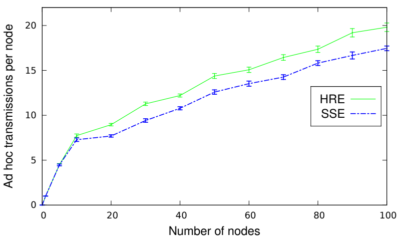


Figure 8. Ad hoc overhead

VI. Conclusion and Future Work

We presented our DMHC algorithm, which forms sensing clusters in order to reduce network traffic and user costs. Each cluster consists of a clusterhead, responsible for the communication of the data, and multiple sensingheads, responsible for the data collection. Clusterheads are selected based on the communication costs in order to keep user costs as low as possible. For the sensinghead election, we proposed two approaches: *SSE* and *HRE*. We analyzed our DMHC algorithm based on simulations using the JiST/SWANS framework. The results show that already for low node densities, DMHC significantly reduces network transmissions, transmission costs, and number of measurements. Energy costs are also within reasonable boundaries, and the ad hoc overhead comparison shows that the performance of both sensinghead election algorithms is adequate.

In our future work, we will elaborate our concept on two main aspects: First, we will implement an adaptive sensinghead election, which automatically adapts ρ to optimize the coverage. Second, we plan to integrate prediction-based approaches to further minimize traffic. In addition, a more comprehensive evaluation is planned, in which the proposed approach is compared to other clustering schemes and more realistic urban mobility and energy models are applied. Further, we will investigate in privacy and incentive schemes that can be utilized to complement our concept.

References

[1] Nokia, "Eco Sensor Concept," http://ncomprod.nokia.com/A41283046, Accessed: 15.06.2012.

[2] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G.-S. Ahn, "The Rise of People-Centric Sensing," *IEEE Internet Computing*, vol. 12, pp. 12–21, 2008.

[3] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, "A survey of mobile phone sensing," *IEEE Communications*, vol. 48/9, pp. 140–150, 2010.

[4] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications*, vol. 49/11, pp. 32 –39, 2011.

[5] K. Wiesner, M. Dürr, and M. Duchon, "Private Pooling: A Privacy-Preserving Approach for Mobile Collaborative Sensing," in *Proc. of MobiSec '11*, 2011.

[6] S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "Metrosense project: People-centric sensing at scale," in *Proc. of WSW at Sensys'06*, 2006.

[7] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich, "Mobiscopes for human spaces," *IEEE Pervasive Computing*, vol. 6, pp. 20–29, 2007.

[8] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "Soundsense: scalable sound sensing for people-centric applications on mobile phones," in *Proc. of MobiSys '09*, 2009, pp. 165–178.

[9] A. Ephremides, J. Wieselthier, and D. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proc. of the IEEE*, vol. 75/1, pp. 56–73, 1987.

[10] M. Gerla and J. T.-C. Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, pp. 255–265, 1995.

[11] S. Basagni, "Distributed Clustering for Ad Hoc Networks," in *Proc. of I-SPAN '99*, 1999, pp. 310–315.

[12] J. Yu and P. Chong, "A survey of clustering schemes for mobile ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 7/1, pp. 32 –48, 2005.

[13] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826 – 2841, 2007.

[14] S. Motegi, Y. Nishimura, and K. Yoshihara, "Communication Algorithm for Statistic Monitoring in People-Centric Sensing Networks," in *Proc. of ICN '10*, 2010, pp. 133–138.

[15] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proc. of MobiSys '09*, 2009, pp. 179–192.

[16] B. Priyantha, D. Lymberopoulos, and J. Liu, "LittleRock: Enabling Energy-Efficient Continuous Sensing on Mobile Phones," *IEEE Pervasive Computing*, vol. 10/3, pp. 12–15, 2011.

[17] "JiST / SWANS - v1.0.6," http://jist.ece.cornell.edu/, Accessed: 15.06.2012.

[18] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in *Proc. of IMC '09*, 2009, pp. 280–293.

Loop-Free Routing in Low-Power and Lossy Networks

Jianlin Guo, Chuan Han, Philip Orlik, Jinyun Zhang
Mitsubishi Electric Research Laboratories
Cambridge, USA
{guo, chan, porlik, jzhang}@merl.com

Koichi Ishibashi
Communication Systems Technology Department
Mitsubishi Electric Corporation IT R&D Center
Ofuna, Japan
Ishibashi.Koichi@ce.MitsubishiElectric.co.jp

***Abstract*—IPv6 based Low-Power and Lossy Networks (LLNs) are emerging. Internet Engineering Task Force (IETF) has developed an IPv6 Routing Protocol for LLNs (RPL), which is widely considered as a feasible routing protocol for LLNs. However, routing loops and lack of a loop-free local route repair mechanism are two major open issues to be addressed in RPL. Based on the framework of RPL, this paper proposes a Loop-Free Routing Protocol for LLNs (LRPL). We provide an innovative rank computation method and a loop-free local route repair mechanism to eliminate routing loops in RPL. Simulation results show that the proposed LRPL performs much better than conventional routing protocols in terms of packet delivery rate, end-to-end packet delay, and routing overhead.**

Keywords-loop-free routing; loop-free local route repair; low power and lossy network; routing protocol for low power and lossy network; directed acyclic graph; destination oriented directed acyclic graph; bidirectional routes

I. INTRODUCTION

Low-Power and Lossy Networks (LLNs) are a class of networks in which routers and their communication links are constrained. LLN routers typically operate with constrains on processing power, memory, power consumption, and lifetime. Their communication links are characterized by high loss rate, low data rate, low transmission power, and short transmission range. There can be from a few dozen up to thousands of nodes within a LLN. The characteristics of LLN require that routing overhead must be much less than application data. Therefore, routing in LLN is different from routing in mobile ad-hoc networks. Conventional routing protocols, such as Ad-hoc On-demand Distance Vector (AODV) [1] and Dynamic Source Routing (DSR) [2], designed for mobile ad-hoc networks are not suitable for routing in LLNs because of high routing overhead. IETF has developed an IPv6 Routing Protocol for LLNs (RPL) [3].

Based on routing metrics and constraints, RPL builds Directed Acyclic Graph (DAG) topology to establish bidirectional routes for LLNs. RPL routes are optimized for traffic to or from one or more roots that act as sinks. A DAG is partitioned into one or more Destination Oriented DAGs (DODAGs), one DODAG per sink. DODAG is basic logic structure in RPL. The sink in a DODAG is called the DODAG root. RPL supports multipoint-to-point traffic (from nodes inside the LLN to the DODAG root) and point-to-multipoint traffic (from the DODAG root to nodes inside the LLN). Support for point-to-point traffic is also available. The traffic of LLN flows along the edges of DODAG, either upwards to the DODAG root or downwards from the DODAG root.

Upward routes, having the DODAG root as destination, are provided by the DODAG construction mechanism using the DODAG Information Object (DIO) messages. The DODAG root configures the DODAG parameters such as RPLInstanceID, DODAG Version Number, DODAGID, Rank, DTSN, etc. and advertises these parameters in DIO messages. To join a DODAG, a node selects a set of parents on the routes towards the DODAG root and configures its own rank. It also selects a preferred parent as next hop for upward traffic. Upon joining a DODAG, a node transmits the DIO messages to advertise the DODAG parameters.

Downward routes, from the DODAG root to other nodes, are provided by these nodes transmitting the Destination Advertisement Object (DAO) messages. A node selects a subset of its parents as its DAO parents. Three modes of operation for downward routes are specified in RPL:

1) No downward routes maintained by RPL.
2) Storing mode of operation in which each router maintains downward routing tables to all nodes in its sub-DODAG, i.e. nodes that are deeper down in the DODAG. The DAO messages propagate from the nodes towards the root, where each intermediate node adds its reverse routing stack to the DAO message.
3) Non-Storing mode of operation in which only the DODAG root stores routes to all nodes in the network. Each node unicasts DAO messages to the root, which then calculates routes to all destinations by piecing together the information collected from DAO messages. In non-storing mode, downward traffic is sent by way of source routing.

RPL has been implemented and evaluated by researchers. It has been shown that IPv6 with the RPL routing has a battery lifetime of years [4]. RPL based routing for advanced metering infrastructure in smart grid has been proposed [5], in which an expected transmission time based rank computation method has been provided and evaluated. Some considerations in RPL implementation are presented in [6].

RPL is widely considered as a feasible routing protocol for LLNs. However, there are several important issues left unresolved. RPL is not a loop-free routing protocol. Experiment shows that loops occur frequently and in 74.14% of the 4114 snapshots, at least one loop was observed [7]. Even though RPL provides mechanism to resolve loops, researchers have shown that the mechanism may cause even worse turmoil than the routing loops themselves [8]. There is no local route repair mechanism provided in RPL.

In this paper, we present an innovative rank computation method for loop-free routing in LLNs. We also provide a method for local route repair without causing any routing loop. The proposed local route repair method applies to both Storing mode and Non-Storing mode of operation in RPL. Based on the proposed rank computation method, a node can discover multiple bidirectional routes towards the DODAG root. Simulation results show the proposed Loop-Free Routing Protocol for LLNs (LRPL) achieves almost 100% of packet delivery rate with low end-to-end delay and frequent packet transmission in large scale LLNs. It performs much better than the conventional routing protocols.

II. RANK DEFINITION AND RANK SPLIT OPERATION

Rank plays very important role in the DODAG construction and maintenance. The rank of a node defines a position of the node relative to other nodes with respect to the DODAG root. Each node maintains its own rank. Nodes maintain their ranks based on parent-child relationship such that a child must have a rank strictly greater than ranks of all its parents. The DODAG root has no parent and therefore has the lowest rank. The acyclic structure of a DODAG is maintained as long as the rank of any node is strictly greater than ranks of all its parents. It is safe for a node to decrease its rank, as long as its new rank remains greater than ranks of its parents. However, rank increase can cause routing loops within a DODAG. RPL allows rank increase which is the source of routing loops in RPL.

Figure 1 shows an example of RPL routing loop in which the DODAG consists of 10 routers N_1 to N_{10} and the root. The integers are the respective ranks. The DODAG structure is shown by directed edges. If the route from N_1 to the root is broken, N_1 can poison the broken route by advertising a rank of infinity. If this infinity rank advertisement is lost, N_2 still has N_1 as its parent. N_3 then advertises its rank equal to 3, N_1 receives the advertisement from N_3 and selects N_3 as its parent. Loop N_1-N_3-N_2-N_1 is created. The cause of this loop is that N_1 increased its rank to infinity.

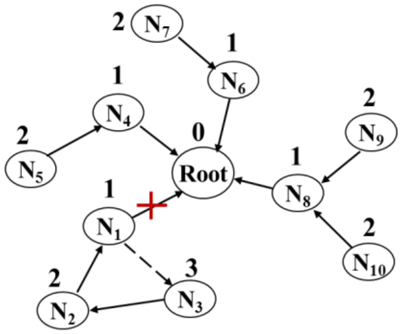


Figure 1. Routing Loop Example in the RPL

The routing loops can be avoided if nodes do not increase their ranks. In order to meet this requirement, we define the rank R as a proper fraction such that:

$$R = \frac{m}{n} \qquad (1)$$

where m and n are integers such that $0 \leq m < n$.

Even though the rank is defined as proper fraction, it is maintained as two integers, numerator m and denominator n. The fractional value of rank is only used in rank operations such as rank comparison.

The principle of this innovative rank definition is that there are an infinite number of proper fractions between any two proper fractions. This principle guarantees that given any two ranks, there always exists a rank in between them. However, integer rank does not possess such property because there is no integer existing between any two consecutive integers.

For any two ranks $R_1 = m/n$ and $R_2 = p/q$, the rank split operation is defined as:

$$sp(R_1, R_2) = \frac{m+p}{n+q} \qquad (2)$$

It can be shown that if $R_1 < R_2$, then $R_1 < sp(R_1,R_2) < R_2$.

In this paper, we define the root rank as 0/1 and the infinite rank as 1/1. The infinite rank can not be advertised in the DIO messages.

III. DODAG CONSTRUCTION

In RPL, a node may act as a router or a leaf node. To construct a new DODAG, the DODAG root transmits a DIO message containing new (RPLInstanceID, DODAGID) tuple. To construct a new DODAG Version, the DODAG root transmits a DIO message with an increased DODAG Version Number. The DODAG Version Number is monotonically incremented by the DODAG root. The DIO message is transmitted via link-local multicasting to all-RPL-nodes. Nodes obtain the DODAG parameters configured by the DODAG root in received DIO messages. A node must keep the DODAG parameters unchanged except Rank and DTSN.

In this paper, we use symbols such as N_i, N_j, N_k, etc. to denote nodes and use $R(N_i)$ to denote the rank of node N_i. For simplicity, we assume RPLInstanceID and DODAGID are fixed. To construct and maintain a DODAG, a node N_i maintains following state parameters:

TABLE 1. Node State Parameters

$R(N_i)$	Rank of node N_i as proper fraction m/n
$P(N_i)$	Parent set of node N_i
$p(N_i)$	Preferred parent of node N_i
$c(N_i)$	The minimum cost from node N_i to the DODAG root
$c(N_i,N_j)$	Cost from node N_i to node N_j
$VN(N_i)$	DODAG Version Number maintained by node N_i
DR-SN(N_i)	DODAG repair sequence number of node N_i
T_p	Parent threshold

The cost can be hop count, expected transmission time, and other options. For a node, if the number of parents is less than T_p, the node can add more parents into its parent set if such parents are available. A node N_i maintains its parent set $P(N_i)$ such that for each parent $N_p \in P(N_i)$, $R(N_i) > R(N_p)$.

Initially, all nodes do not belong to any DODAG and do not transmit the DIO messages because a node can transmit the DIO messages only if the node joins a DODAG. The DODAG root initiates a new DODAG construction process by configuring the DODAG parameters and transmitting the DIO messages to advertise the DODAG parameters.

In response to receiving a DIO message, a node can update its state parameters only if one of the following conditions holds:

(1) The node wants to join a DODAG
(2) The DODAG Version Number in the DIO message is greater than the DODAG Version Number maintained by receiving node
(3) The DODAG Version Number in the DIO message equals the DODAG Version Number of receiving node, and the rank in the DIO message is lower than the rank of receiving node.

Upon receiving a DIO message transmitted by the DODAG root containing new (RPLInstanceID, DODAGID) tuple or new DODAGVersionNumber, the first hop nodes of the DODAG root may join new DODAG or new DODAG Version. To do so, the first hop nodes add the DODAG root into their parent set and store the DODAG parameters. The first hop nodes keep all DODAG parameters unchanged except the rank. The first hop nodes set their ranks such that their ranks > 0/1 and their ranks <= sp(0/1, 1/1) = 1/2. Upon joining a new DODAG or a new DODAG Version, the first hop routers generate and transmit the DIO messages to advertise the DODAG parameters.

Upon receiving the DIO messages transmitted by the first hop routers, the second hop nodes of the DODAG root that want to join new DODAG or new DODAG Version perform similar procedure as the first hop nodes do. However, in this case, the second hop nodes may receive multiple DIO messages from the first hop routers. The second hop nodes use received DIO messages to calculate their ranks and select a subset of the DIO message senders as their parents. To calculate its rank, a second hop node find the maximum rank, Rank_Max, among all ranks of its parents and sets its rank such that its rank > Rank_Max and its rank <= sp(Rank_Max, 1/1). The second hop routers then generate and transmit the DIO messages same as the first hop router do.

A first hop node of the DODAG root may also receive the DIO messages transmitted by other first hop routers. The first hop node may perform same procedure as the second hop nodes do to select more parents.

This DIO message propagation process continues until all nodes in network receive the DIO messages, store the DODAG parameters, select parents and determine ranks.

Figure 2 shows the process of DODAG construction, where router N_j transmitted the DIO message containing $VN(N_j)$, $R(N_j)$, $c(N_j)$, etc. and node N_i receives the DIO message. $VN(N_i)$, $R(N_i)$, $P(N_i)$, and $p(N_i)$ are state parameters maintained by node N_i.

Upon receiving the DIO message, node N_i first checks if the received DIO message is malformed or was received already. If yes, N_i discards the DIO message. If no, N_i checks if N_j equals N_i. If yes, N_i discards the DIO message, because N_i just received its own DIO message. Otherwise, N_i processes the DIO message further.

N_i checks if a new DODAG is advertised in the DIO message. If yes, N_i joins new DODAG. It initializes its state parameters as $VN(N_i) = VN(N_j)$, $P(N_i) = \{N_j\}$, $p(N_i) = N_j$, $R(N_i) = sp(R(N_j),1/1)$, $c(N_i) = c(N_j) + c(N_i, N_j)$, and stores other DODAG parameters. N_i then resets its trickle timer to transmit the DIO message and schedules a DAO message transmission if N_j is also selected as its DAO parent. Otherwise, the DIO processing goes to next step.

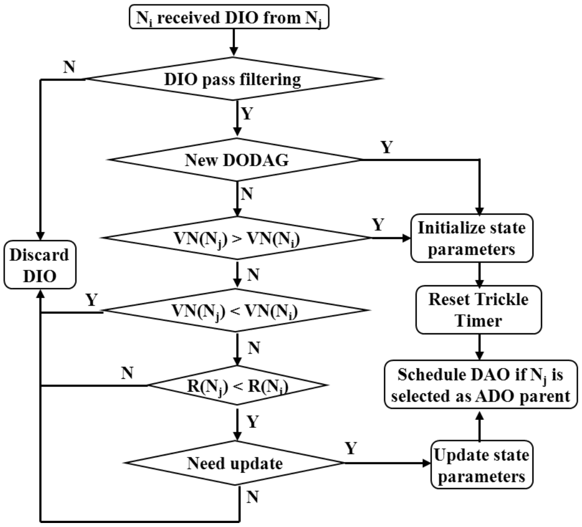

N_i received DIO from N_j
DIO pass filtering
N
Y
New DODAG
Y
N
VN(N_j) > VN(N_i)
Y
Initialize state parameters
Discard DIO
N
Reset Trickle Timer
Y
VN(N_j) < VN(N_i)
N
N
R(N_j) < R(N_i)
Schedule DAO if N_j is selected as ADO parent
Y
Need update
Y
Update state parameters
N

Figure 2. The DODAG Construction Process

N_i checks if the $VN(N_j) > VN(N_i)$. If yes, N_i joins new DODAG Version. It initializes its state parameters same as joining new DODAG. N_i also clears downward routing tables if the mode of operation is Storing. N_i then resets its trickle timer to transmit the DIO message and schedules a DAO message transmission if N_j is also selected as its DAO parent. Otherwise, the DIO processing goes to next step.

N_i checks if $VN(N_j) < VN(N_i)$. If yes, it discards the DIO message. If $VN(N_j) = VN(N_i)$ and $R(N_j) \geq R(N_i)$, N_i discards the DIO message. If $VN(N_j) = VN(N_i)$ and $R(N_j) < R(N_i)$, N_i checks if it is necessary to update its state parameters by using received the DIO message. If no, N_i discards the DIO message. If yes, N_i updates state parameters. If N_j is not in its parent set $P(N_i)$ and $|P(N_i)| < T_p$, N_i adds N_j into its parent set such that $P(N_i) = P(N_i) \bigcup \{N_j\}$ and updates its preferred parent as

$$p(N_i) = \arg \min_{N_k \in P(N_i)} \{c(N_i, N_k) + c(N_k)\} \quad (3)$$

and the minimum cost as

$$c(N_i) = c(N_i, p(N_i)) + c(p(N_i)) \quad (4)$$

If there are multiple parents that have the same minimum cost, N_i can randomly pick one preferred parent. N_i then schedules a DAO message transmission if N_j is also added into its DAO parent set. If N_j is already in DAO parent set, N_i makes necessary updates without scheduling the DAO message transmission.

A node can receive multiple DIO messages from neighbors within the same DODAG. These DIO messages can be used to select a subset of the DIO message transmitters as its parents and determine its rank. Among all its parents, the node selects

ISBN: 978-1-61208-207-3

one parent with the minimum cost as its preferred parent to be used as the next hop along upward routes to the root.

Figure 3 shows an example of the DODAG construction. Initially, nodes N_1 – N_6 are not members of any DODAG version, and their parent sets are empty. The DODAG root sets its rank to 0/1, the DODAG version number to 1, and its parent set to empty.

The root transmits the DIO message carrying its DODAG version number 1, and rank 0/1. Nodes N_1, N_2 and N_3 receive the DIO message. Because nodes N_1, N_2 and N_3 are not members of the newly advertised DODAG, N_1, N_2 and N_3 joins the DODAG and set their DODAG version numbers to 1, ranks to sp(1/1, 0/1) = 1/2, and select the root as their preferred parent.

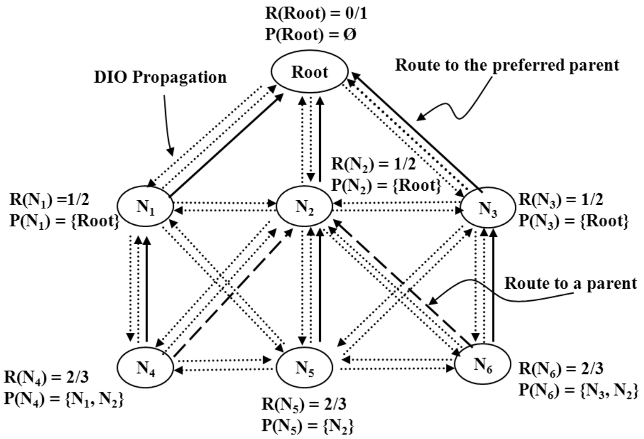


Figure 3. The DODAG Construction Example

Upon joining the DODAG, nodes N_1, N_2, and N_3 transmit the DIO messages with DODAG version number 1 and rank 1/2. The DIO messages from routers N_1, N_2, and N_3 are discarded by the root because the DODAG version number in the DIO messages equals the DODAG version number of the root, and the rank in the DIO messages is greater than the rank of the root.

N_1 discards the DIO message from N_2 because the DODAG version number in the DIO message equals the DODAG version number of N_1, and the rank in the DIO message equals N_1's rank. Similarly, N_2 discards the DIO messages from N_1 and N_3, and N_3 discards the DIO message from N_2.

N_4 receives DIO messages from N_1 and N_2. Because N_4 is not a member of the advertised DODAG, N_4 joins the DODAG and sets its DODAG version number to 1, its rank to sp(1/1, 1/2) = 2/3, and select N_1 as the preferred parent and N_2 as parent. Similarly, N_6 receives the DIO messages from N_2 and N_3, joins the DODAG, sets its DODAG version number to 1, rank to sp(1/1, 1/2) = 2/3, adds N_2 and N_3 into its parent set, and selects N_3 as the preferred parent. N_5 receives the DIO messages from N_1, N_2, and N_3. Because N_5 is not a member of the advertised DODAG, N_5 joins the DODAG and sets its DODAG version number to 1, its rank to sp(1/1, 1/2) = 2/3. However, N_5 only selects N_2 as its parent and preferred parent even though N_2 may select N_1, N_2, and N_3 as parents.

Upon joining the DODAG, nodes N_4, N_5 and N_6 also transmit their DIO messages. These DIO messages are discarded by their neighbors because the DODAG version number in the DIO messages equals the DODAG version number of the neighbors, and the rank of N_4, N_5 and N_6 are not lower than ranks of the neighbors.

IV. DODAG Local Repair

The DODAG local repair is performed by using two new RPL control messages, the DODAG Repair Request (DR-REQ) message and the DODAG Repair Reply (DR-REP) message.

The DR-REQ message consists of N_q, R(N_q), VN(N_q), DR-SN(N_q), NL-REQ, and other fields. The N_q is the identifier of node generating DR-REQ message, R(N_q) is the rank of N_q, VN(N_q) is the DODAG Version Number of N_q, DR-SN(N_q) is the DODAG repair sequence number of N_q, NL-REQ is the node list traveled through by DR-REQ message and present only in Non-Storing mode. In addition, the DR-REQ message may also have a hop count field and a maximum hop count field. Once hop count reaches the maximum hop count, the DR-REQ message is discarded.

The DR-REP message consists of N_q, R(N_q), DR-SN(N_q), D, R(N_p), c, VN(N_p), NL-REP, and other fields. N_q, R(N_q) and DR-SN(N_q) are same as in the DR-REQ message. N_q is destination of DR-REP message. D indicates the travel direction of DR-REP message, R(N_p) is the rank of router generating the DR-REP message if D = UP and is the rank of router transmitting the DR-REP message if D = DOWN, c is the minimum cost of link(s) from the router transmitting the DR-REP message to the DODAG root, VN(N_p) is the DODAG Version Number of DR-REP message generator, and NL-REP is combination of NL-REQ in the DR-REQ message and node list travelled by upward DR-REP message. D and NL-REP are present only in Non-Storing mode.

When a node detects a broken route by using mechanisms provided in RPL, it may need to discover new parents. The DODAG is locally repaired by node transmitting a DR-REQ message. The DR-REQ message is transmitted by the DR-REQ message generator via link-local multicasting to all-RPL-nodes.

Upon receiving a DR-REQ message, a link-local neighbor discards the DR-REQ message if it does not have a route to the DODAG root. If the link-local neighbor is the DODAG root or a router that has a route to the DODAG root and a rank lower than the rank carried in the DR-REQ message, this neighbor generates a DR-REP message. If the link-local neighbor has route to the DODAG root and its rank is greater than or equal to the rank carried in the DR-REQ message, this neighbor forwards the DR-REQ message to its preferred parent.

In Storing mode, the DR-REP message generator transmits the DR-REP message to node N_q by using downward routing tables. Route entry is added into downward tables while the DR-REQ message is processed. In Non-Storing mode, the DR-REP message is forwarded up to the DODAG root, which then transmits the DR-REP message to node N_q by using source routing.

A. DODAG Local Repair in Storing Mode

In Storing mode, if the route from node N_q to its parent N_{qp} is broken, N_q removes N_{qp} from its parent set such that $P(N_q) = \{N_k \mid N_k \in P(N_q) / \{N_{qp}\}\}$. If the updated parent set $P(N_q)$ is empty, N_q transmits a DR-REQ message to discover new parents. If the updated parent set $P(N_q)$ is not empty, N_q checks if N_{qp} is its preferred parent $p(N_q)$. If yes, N_q selects a new preferred parent $p(N_q)$ as shown in equation (3) and updates $c(N_q)$ as shown in equation (4). If N_{qp} is also in N_q's DAO parent set, N_q schedules a No-Path DAO message transmission.

Whether or not N_{qp} is N_q's preferred parent, N_q can transmit a DR-REQ message to discover additional parents if $|P(N_q)| < T_p$. To construct a DR-REQ message in Storing mode, N_q increases DR-SN(N_q) by 1 and uses N_q, $R(N_q)$, $VN(N_q)$, and DR-SN(N_q) to fill the fields in the DR-REQ message.

A.1 DR-REQ Message Processing

Figure 4 shows the procedure of processing the DR-REQ message when router N_i receives a DR-REQ message from N_j in which $VN(N_q)$, N_q, $R(N_q)$ and DR-SN(N_q) are the parameters carried in the DR-REQ message, and $VN(N_i)$, R (N_i), and $P(N_i)$ are state parameters of N_i.

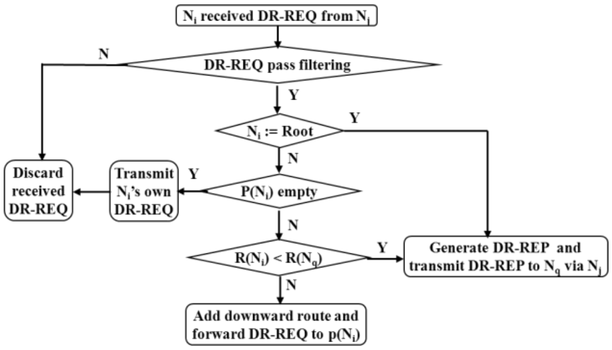

N_i received DR-REQ from N_j
N
DR-REQ pass filtering
Y
N_i := Root
Y
N
Discard received DR-REQ
Transmit N_i's own DR-REQ
Y
$P(N_i)$ empty
N
$R(N_i) < R(N_q)$
Y
Generate DR-REP and transmit DR-REP to N_q via N_j
N
Add downward route and forward DR-REQ to $p(N_i)$

Figure 4. The DR-REQ Processing in Storing Mode

Router N_i first performs the filtering process. The DR-REQ message is discarded if this DR-REQ message is received already by checking N_q and DR-SN(N_q) or if the $VN(N_q)$ is not equal to $VN(N_i)$ or if the DR-REQ message is transmitted by N_i's parent or if the DR-REQ message is generated by N_i's parent or by N_i itself.

If N_i is the DODAG root, N_i accepts the DR-REQ message, generates a DR-REP message by copying N_q, $R(N_q)$, DR-SN(N_q) from the DR-REQ message, and setting $R(N_p)$ = R(Root), c = 0, $VN(N_p)$ = VN(Root), and transmits the DR-REP message to node N_q via next hop node N_j.

If N_i is not the DODAG root, the processing of DR-REQ message is as follows. If N_i's parent set $P(N_i)$ is empty, N_i discards the DR-REQ message and transmits a its own DR-REQ message. If N_i's parent set $P(N_i)$ is not empty and $R(N_i) < R(N_q)$, N_i accepts the DR-REQ message and generates a DR-REP message by copying N_q, $R(N_q)$, DR-SN(N_q) from the DR-REQ message, and setting $R(N_p) = R(N_i)$, $c = c(N_i)$, $VN(N_p) = VN(N_i)$. N_i transmits the DR-REP message to node N_q via next hop node N_j. If N_i's parent set $P(N_i)$ is not empty and $R(N_i) \geq R(N_q)$, N_i adds a downward routing entry to node N_q into its downward routing table, and forwards the DR-REQ message to its preferred parent $p(N_i)$.

A.2 DR-REP Message Processing

Figure 5 shows the procedure of processing the DR-REP message when node N_i receives a DR-REP message from router N_j in which $VN(N_p)$, N_q, $R(N_p)$, DR-SN(N_q) and $R(N_q)$ are the parameters carried in the DR-REP message, and $VN(N_i)$, $R(N_i)$, $P(N_i)$, $p(N_i)$, $c(N_i)$, and T_p are state parameters of node N_i.

If $VN(N_p)$ is not equal to $VN(N_i)$ or this DR-REP message is received already, node N_i discards the DR-REP message. Otherwise, N_i processes the DR-REP message further.

If N_i is the DR-REQ message generator and N_j is not in N_i's parent set $P(N_i)$, N_i adds N_j into $P(N_i)$ if $|P(N_i)| < T_p$ and updates $p(N_i)$ according to equation (3) and $c(N_i)$ according to equation (4). N_i then schedules a DAO message transmission if N_j is also added into its DAO set.

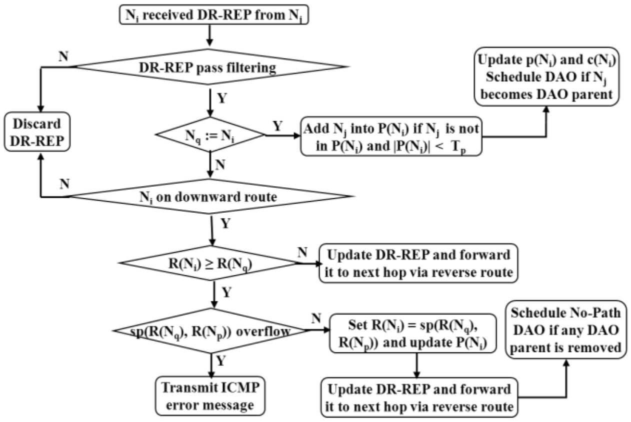

N_i received DR-REP from N_j
N
DR-REP pass filtering
Update $p(N_i)$ and $c(N_i)$ Schedule DAO if N_j becomes DAO parent
Y
Discard DR-REP
$N_q := N_i$
Y
Add N_j into $P(N_i)$ if N_j is not in $P(N_i)$ and $|P(N_i)| < T_p$
N
N
N_i on downward route
Y
$R(N_i) \geq R(N_q)$
N
Update DR-REP and forward it to next hop via reverse route
Y
sp($R(N_q)$, $R(N_p)$) overflow
N
Set $R(N_i)$ = sp($R(N_q)$, $R(N_p)$) and update $P(N_i)$
Schedule No-Path DAO if any DAO parent is removed
Y
Transmit ICMP error message
Update DR-REP and forward it to next hop via reverse route

Figure 5. The DR-REP Processing in Storing Mode

If N_i is not the DR-REQ message generator, the processing of the DR-REP message is as follows. If N_i is not on the downward route, N_i discards the DR-REP message. Otherwise, if $R(N_i) \geq R(N_q)$, N_i decreases its rank $R(N_i)$ as

$$R(N_i) = sp(R(N_q) + R(N_p)) \qquad (5)$$

and updates its parent set $P(N_i)$ as

$$P(N_i) = \{N_k \mid R(N_k) < R(N_i), N_k \in P(N_i)\} \qquad (6)$$

If the preferred parent $p(N_i)$ is removed due to its rank decrease, N_i selects a new $p(N_i)$ according to equation (3) and updates $c(N_i)$ according to equation (4). N_i then updates the DR-REP message by setting $R(N_p) = R(N_i)$ and $c = c(N_i)$, forwards the DR-REP message to next hop node obtained from downward routing table. N_i schedules a No-Path DAO message transmission if any DAO parent is removed.

If $R(N_i) < R(N_q)$, N_i updates the DR-REP message by setting $R(N_p) = R(N_i)$ and $c = c(N_i)$, forwards it to next hop node obtained from downward routing table. In Storing mode, $R(N_i) < R(N_q)$ occurs if N_i is on multiple DODAG repair routes. When N_i receives a DR-REP message, it may decrease its rank. Therefore, subsequent DR-REP messages may carry a rank $R(N_q)$ greater than or equal to $R(N_i)$. If N_i is only on a single DODAG repair route, $R(N_i) \geq R(N_q)$ must be true based on the DR-REQ message processing procedure.

By the definition of rank split operation, it is easy to show that rank $R(N_p)$ in the DR-REP message is the maximum rank of routers on the route from the DR-REP message generator to the DR-REP message transmitter. $R(N_p)$ is always less than $R(N_q)$. Therefore, when the DR-REP message reaches the DR-REQ message generator N_q, rank $R(N_p)$ in the DR-REP message must be less than $R(N_q)$. Therefore, the rank monotonically increases along a route from the DE-REP message generator to the DR-REQ message generator. This guarantees that rank increases monotonically along the route from the DODAG root to any node.

B. DODAG Local Repair in Non-Storing Mode

The processing of upward route failure from node N_q to its parent N_{qp} in Non-Storing mode is mostly similar to that in Storing mode. The first difference is that after removing a DAO parent, the node schedules a transmission of DAO message instead of No-Path DAO message. The second difference is that NL-REQ field is present in the DR-REQ message; D and NL-REP fields are present in the DR-REP message. The third difference is that the DR-REP message is first forwarded upwards to the DODAG root, which then sends the DR-REP message downwards to node N_q.

B.1 DR-REQ Message Processing

Figure 6 shows the procedure of processing the DR-REQ message when N_i receives a DR-REQ message from N_j in which $VN(N_q)$, N_q, $R(N_q)$, $DR\text{-}SN(N_q)$, and NL-REQ are the parameters in the DR-REQ message and $VN(N_i)$, $R(N_i)$, and $P(N_i)$ are state parameters of N_i.

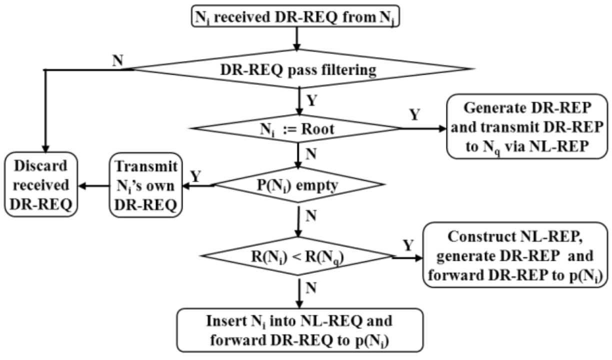

N_i received DR-REQ from N_j
DR-REQ pass filtering
N
Y
N_i := Root
Y
Generate DR-REP and transmit DR-REP to N_q via NL-REP
N
Discard received DR-REQ
Transmit N_i's own DR-REQ
Y
P(N_i) empty
N
R(N_i) < R(N_q)
Y
Construct NL-REP, generate DR-REP and forward DR-REP to p(N_i)
N
Insert N_i into NL-REQ and forward DR-REQ to p(N_i)

Figure 6. The DR-REQ Processing in Non-Storing Mode

The DR-REQ message is discarded if this DR-REQ message is received already or if $VN(N_q)$ is not equal to $VN(N_i)$ or if the DR-REQ message is transmitted by N_i's parent or if the DR-REQ message is generated by N_i's parent or by N_i itself.

If N_i is the DODAG root, N_i accepts the DR-REQ message, and generates a DR-REP message similarly as in Storing mode. However, in this case, the DODAG root sets D to DOWN, NL-REP field in DR-REP message to NL-REQ field in DR-REQ message, and transmits the DR-REP message to node N_q via the route provided by NL-REP field.

If N_i is not the DODAG root, the processing of the DR-REQ message is as follows.

If N_i's parent set $P(N_i)$ is empty, N_i discards the received DR-REQ message and transmits its own DR-REQ message. If N_i's parent set $P(N_i)$ is not empty and $R(N_i) < R(N_q)$, N_i accepts the DR-REQ message, and generates a DR-REP message similar as the DODAG root does. However, N_i sets D = UP, NL-REP = NL-REQ $\bigcup \{N_i\}$, and forwards the DR-REP message to its preferred parent $p(N_i)$. If N_i's parent set $P(N_i)$ is not empty and $R(N_i) \geq R(N_q)$, N_i updates the DR-REQ message by inserting N_i in NL-REQ such that NL-REQ = NL-REQ $\bigcup$ $\{N_i\}$, and forwards the DR-REQ message to its preferred parent $p(N_i)$.

B.2 DR-REP Message Processing

Figure 7 shows that N_i receives a DR-REP message from N_j in which $VN(N_p)$, N_q, $R(N_p)$, $DR\text{-}SN(N_q)$, D, $R(N_q)$ and NL-REP are the parameters in the DR-REP message, $VN(N_i)$, $R(N_i)$, $P(N_i)$, $p(N_i)$, $c(N_i)$, and T_p are state parameters of N_i.

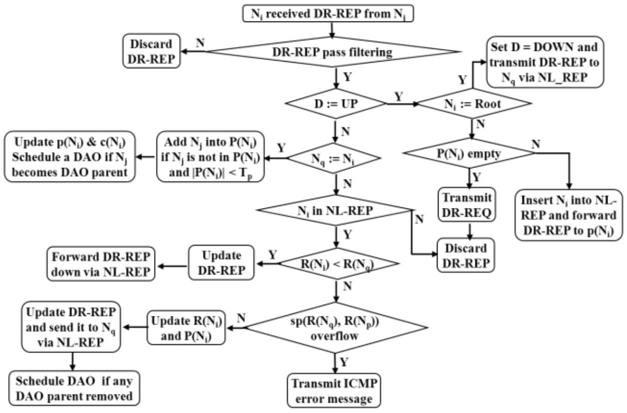

N_i received DR-REP from N_j
Discard DR-REP
N
DR-REP pass filtering
Set D = DOWN and transmit DR-REP to N_q via NL_REP
Y
Y
D := UP
Y
N_i := Root
N
N
Update p(N_i) & c(N_i) Schedule a DAO if N_j becomes DAO parent
Add N_j into P(N_i) if N_j is not in P(N_i) and |P(N_i)| < T_p
Y
N_q := N_i
P(N_i) empty
N
Y
N
Transmit DR-REQ
Insert N_i into NL-REP and forward DR-REP to p(N_i)
N_i in NL-REP
N
Y
Discard DR-REP
Forward DR-REP down via NL-REP
Update DR-REP
Y
R(N_i) < R(N_q)
N
Update DR-REP and send it to N_q via NL-REP
Update R(N_i) and P(N_i)
N
sp(R(N_q), R(N_p)) overflow
Y
Schedule DAO if any DAO parent removed
Transmit ICMP error message

Figure 7. The DE-REP Processing in Non-Storing Mode

If $VN(N_p)$ is not equal to $VN(N_i)$ or this DR-REP message is received already, N_i discards the DR-REP message.

If D = UP, the DR-REP message is transmitted upwards. If N_i is the DODAG root, N_i updates the DR-REP message by changing D = DOWN, $R(N_p) = R(Root)$, c = 0, and transmits DR-REP message down to node N_q via the route provided by NL-REP field. If N_i is not the DODAG root and its parent set $P(N_i)$ is not empty, N_i updates DR-REP message such that NL-REP = NL-REP $\bigcup\{N_i\}$, and forwards the DR-REP message to its preferred parent $p(N_i)$. If N_i is not the DODAG root and its parent set $P(N_i)$ is empty, N_i discards the received DR-REP message.

If D = DOWN, the DR-REP message is transmitted downwards. If N_i is DR-REQ message generator, N_j is not in its parent set $P(N_i)$ and $|P(N_i)| < T_p$, N_i adds N_j into $P(N_i)$ and updates $p(N_i)$ according to equation (3) and $c(N_i)$ according to equation (4). N_i then schedules a DAO message transmission if N_j is also added into its DAO parent set. If N_i is not the DR-REQ message generator and is not on the downward route, N_i discards the DR-REP message. Otherwise, if $R(N_i) < R(N_q)$, N_i updates the DR-REP message by setting $R(N_p) = R(N_i)$, c = $c(N_i)$, and forwards the DR-REP message to node N_q via the route provided by NL-REP field. If $R(N_i) \geq R(N_q)$, N_i decreases its rank $R(N_i)$ to $sp(R(N_q), R(N_p))$ and updates its parent set $P(N_i)$, the preferred parent $p(N_i)$ and cost $c(N_i)$ according to equations (5), (6), (3) and (4) respectively. N_i then updates the DR-REP message by setting $R(N_p) = R(N_i)$, c = $c(N_i)$, and forwards the DR-REP message to node N_q via the route provided by NL-REP field. Furthermore, N_i schedules a DAO message transmission if any DAO parent is removed due to its rank decrease.

By definition of the rank split operation, it can also been shown that rank $R(N_p)$ in the downward DR-REP message is the maximum rank of routers on the route from the root to DR-REP transmitter. $R(N_p)$ is always less than $R(N_q)$. Therefore, when the DR-REP message reaches the DR-REQ message generator, the rank $R(N_p)$ in the DR-REP message must be less than $R(N_q)$, which is the rank of the DR-REQ message generator. Hence, the rank monotonically increases from the root to the DR-REQ message generator. This guarantees that rank increases monotonically along a route from the root to any node.

Figure 8 illustrates how the broken route in Figure 1 is handled by the proposed DODAG local repair method. The fractions are the ranks of nodes and the root, respectively. After the route to the root is broken, N_1 removes the root from its parent set $P(N_1)$ and transmits a DR-REQ message with N_q = N_1 and $R(N_q) = R(N_1) = 1/2$. N_2 discards the DR-REQ message because this DR-REQ message is transmitted by its parent N_1. N_3 forwards the DR-REQ message to N_2 because $R(N_q)$ in the DR-REQ message is smaller than its rank $R(N_3)$ = 3/4. However, the DR-REQ message forwarded by N_3 is discarded by N_2 because the DR-REQ message is generated by N_2's parent N_1. N_5 forwards the DR-REQ message to N_4 because $R(N_q)$ is smaller than $R(N_5)$ = 2/3. N_4 forwards the DR-REQ message to the root because the rank $R(N_q)$ equals its rank $R(N_4)$ = 1/2. The root generates a DR-REP message with $R(N_p)$ = R(Root) = 0/1 and transmits the DE-REP message back to N_1.

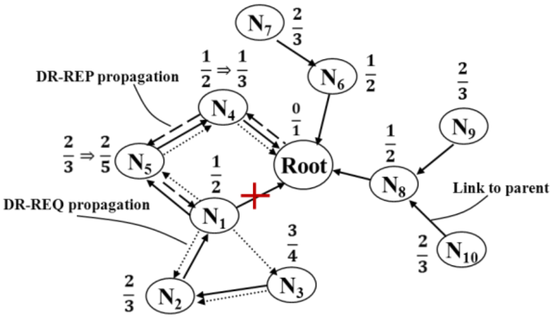


Figure 8. Example of the DODAG Local Repair

Upon receiving this DR-REP message, N_4 decreases its rank $R(N_4)$ to 1/3 because its old $R(N_4)$ = 1/2, which equals to $R(N_q)$. N_4 then sets $R(N_p)$ to its new rank $R(N_4)$ = 1/3 and forwards the DR-REP message to N_5. Upon receiving the DR-REP message, N_5 decreases its rank to 2/5 because its old rank $R(R_5)$ = 2/3, which is greater than $R(N_q)$ = 1/2. N_5 then sets $R(N_p)$ to its new rank $R(N_5)$ = 2/5 and forwards DR-REP message to N_1. Upon receiving the DR-REP message from N_5, N_1 selects N_5 as its parent and transmits DIO message without changing its rank. The DODAG local repair process initiated by N_1 is completed.

V. Simulations

The performance of AODV and DSR has been evaluated considerably. The NS2 simulator is used to simulate AODV and DSR in [10 - 17. Unfortunately, most of simulation results are obtained with a small number of nodes, less or equal to 50 nodes [11-17]. Another common fact is that all simulations are performed using IEEE 802.11 wireless network instead of IEEE 802.15.4 wireless network, which is designed for LLNs. RPL has been implemented and simulated in [5]. However, the simulation was also done over IEEE 802.11 wireless networks.

We used NS2 simulator with IEEE 802.15.4 to simulate the performance of proposed routing protocol in large scale LLNs. Nodes are randomly displaced in a rectangle with the DODAG root in the middle of rectangle. In the simulation, transmission range is 30 meters and data rate is 100kbps. The CBR traffic is employed with 50 bytes of payload. TwoRayGround channel model and Shadowing channel model [8] are used. Performance metrics are data packet delivery rate (PDR), data average end-to-end delay (AED) and routing overhead (ROH) per data packet.

TABLE 2. TwoRayGround Channel Model with 1000 Nodes

Metrics	CBR Interval = 5 Minutes		CBR Interval = 2 Minutes	
	AODV	LRPL	AODV	LRPL
PDR	56.78%	100%	13.8%	100%
AED	920ms	140ms	2310ms	150ms
ROH	5.96	0.22	4.42	0.09

Tables 2 shows simulation results using TwoRayGround channel model, 1000 nodes and 24 hours simulation time. 1000 nodes are randomly deployed in a 320m by 320m rectangle. LRPL achieves 100% of packet delivery rate. AODV only achieves 56.78% of packet delivery rate for 5-minute CBR Interval and drops 82.6% of data packet for 2-minute CBR interval. For 5-minute CBR interval, LRPL is 6.6 times faster than AODV. For 2-minute CBR interval, LRPL is 15.4 times faster than AODV. For 5-minute CBR interval, LRPL's routing overhead is 27 times lower than AODV outing overhead. For 2-minute CBR interval, LRPL's routing overhead is 49 times lower than AODV routing overhead.

TABLE 3. Shadowing Channel Model with 500 Nodes

Metrics	PLE = 2.0		PLE = 2.5		PLE = 3.0	
	AODV	LRPL	AODV	LRPL	AODV	LRPL
PDR	36.7%	99.98%	34.1%	99.83%	32.5%	99.99%
AED	1530ms	177ms	1680ms	184ms	1840ms	166ms
ROH	168.75	0.44	188.75	0.43	550.5	0.43

Table 3 shows the performance comparison with Shadowing channel model and 500 nodes, which are randomly deployed in a 250m by 200m rectangle. The shadowing deviation is 4dB, CBR interval is 30 minutes and simulation time is 24 hours. Table 3 illustrates performance variation of routing protocols as path loss exponent (PLE) changes. LRPL almost achieves 100% of packet delivery rate. However, AODV drops more than 63% of packets. LRPL is about 10 times faster than AODV. The routing overhead of LRPL is at least 380 times lower than that of AODV.

TABLE 4. Shadowing Channel Model with 500 Nodes

Metrics	PLE = 2.0	PLE = 2.5	PLE = 3.0	PLE = 3.5	PLE = 4.0
PDR	99.98%	99.83%	99.99%	99.94%	99.99%
AED	177ms	184ms	166ms	205ms	194ms
ROH	0.44	0.43	0.43	0.58	0.55

Table 4 illustrates a more complete performance of LRPL with Shadowing channel model and 500 nodes. It can be seen that the overall performance of LRPL is excellent. LRPL maintains its performance as path loss exponent increases from 2.0 to 4.0, especially the packer delivery rate, which is almost 100%. The end-to-end packet delay and the routing overhead tend to increase; the change however is very small.

TABLE 5. Shadowing Channel Model with 1000 Nodes

Metrics	PLE = 2.0	PLE = 2.5	PLE = 3.0	PLE = 3.5	PLE = 4.0
PDR	99.60%	99.79%	99.29%	99.28%	99.54%
AED	271ms	249ms	369ms	354ms	303ms
ROH	0.90	1.02	1.16	1.61	1.24

Tables 5 shows simulation results of LRPL using Shadowing channel model and 1000 nodes. It can also be seen that the overall performance of LRPL is also excellent. LRPL achieves also 100% of packet delivery rate for all cases. As path loss exponent increases from 2.0 to 4.0, the end-to-end packet delay and the routing overhead tend to increase.

Tables 4 and 5 show that packet delivery rate of LRPL is almost same for 500 nodes and 1000 nodes. However, the end-to-end delay increases for about 55% and the routing overhead however increases about 150%. The routing overhead increase is mostly contributed by the DODAG local repair packets. It indicates that as the number of nodes increases, communication interference also increases. Therefore, the communication link breaks more often.

To compare the proposed LRPL with RPL, we refer to the results in [5], which simulated RPL using 802.11 wireless network. The performance of RPL was evaluated with smaller shadowing deviation of 1dB and 2dB. For shadowing deviation of 2dB, RPL only achieves a 97.9% of packet delivery rate. On the other hand, LRPL achieves more than 99% of packet delivery rate with shadowing deviation of 4dB. It can be seen that even with lower data rate of 802.15.4 and larger shadowing fading effect, LRPL performs better than RPL.

VI. Conclusion

In this paper, we present a loop-free routing protocol in LLNs based on IETF RPL framework. The proposed routing protocol defines rank as proper fraction to guarantee no routing loops can be created. A DODAG local repair method is also proposed for fast route repair. The proposed routing protocol is simulated by using NS2 simulator with a large number of nodes over IEEE 802.15.4 low power and lossy wireless networks. Simulation results show that the proposed routing protocol performs much better than conventional routing protocols. It achieves almost 100% of packet delivery rate with much shorter end-to-end delay and lower routing overhead. Therefore, it is a desired routing protocol for LLNs, especially when network scale is large and message generation rate is high. We are planning to implement RPL in 802.15.4 wireless network. The results will be reported in the future.

References

[1] C.E. Perkins, E.M. Royer, and S.R. Das, “Ad-hoc On-demand Distance Vector (AODV) routing”, RFC 3561, July 2003

[2] D. Johnson, Y. Hu, and D. Maltz, “The Dynamic Source Routing protocol (DSR) for mobile Ad Hoc networks for IPv4”, RFC 4728, February 2007

[3] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Stuick, JP. Vasseur, and R. Alexander “RPL: IPv6 Routing Protocol for Low power and Lossy Networks”, RFC 6550, March 2012

[4] N. Tsiftes, J. Eriksson, and A. Dunkels, “Poster Abstract: Low-Power Wireless IPv6 Routing with ContikiRPL”, The 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, Stockholm, Sweden, April 2010

[5] D. Wang, Z. Tao, J. Zhang, and A. Abouzeid, “RPL Based Routing for Advanced Metering Infrastructure in Smart Grid”, IEEE International Workshop on Smart Grid Communications, May 2010

[6] T. Clausen and U. Herberg, “Some Considerations on Routing in Particular and Lossy Environment”, Proceedings of the 1st IAB Interconnecting Smart Objects with the Internet Workshop, Prague, Czech Republic, March 2011

[7] T. Clausen, J. Yi, and U. Herberg, “Experieces with RPL: IPv6 Routing Protocol for Low power and Lossy Networks”, the 83rd IETF Plenary Meeting, Paris, France, March 2012

[8] W. Xie, M. Goyal, H. Mosseini, J. Martocci, Y. Bashir, E. Baccelli, and A. Durresi, “Routing Loops in DAG-based Low Power and Lossy Networks”, Proc. IEEE AINA 2010, 2010

[9] K. Fall and K. Varadhan, “The Network Simulator Manual”, http://www.isi.edu/nsnam/ns/ns-documentation.html, May 2010

[10] A. Goel and A. Sharma, “Performance Analysis of Mobile Ad-hoc Network Using AODV Protocl”, International Journal of Computer Science and Security (IJCSS), Volume (3): Issue (5), 2009

[11] D. Singh, P. Trivedi, and J.D. Lal, “Performance Evaluation of DSR and AODV Networking Protocol With Varying Pause Time”, Proceedings of SPIT-IEEE Colloquium and International Conference, Mumbai, India, Vol. 3, 190, 2007

[12] A.H.A. Rahman and Z.A. Zukarnain, “Performance Comparison of AODV, DSDV and I-DSDV Routing Protocols in Mobile Ad Hoc Networks”, European Journal of Scientific Research ISSN 1450-216X Vol.31 No.4 (2009), pp.566-576

[13] N.P. Bobade and N.N. Mhala, “Performance Evaluation of Ad Hoc On Demand Distance Vector in MANETs with varying Network Size using NS-2 Simulation”, International Journal on Computer Science and Engineering, Vol. 02, No. 08, 2010, 2731-2735

[14] A.K. Gupta, H. Sadawarti, and A.K. Verma, “Performance analysis of AODV, DSR & TORA Routing Protocols”, International Journal of Engineering and Technology, Vol.2, No.2, April 2010, ISSN: 1793-8236

[15] S. Shah, A. Khandre, M. Shirole, and G. Bhole, “Performance Evaluation of Ad Hoc Routing Protocols Using NS2 Simulation”, Mobile and Pervasive Computing (CoMPC–2008)

[16] C.E. Perkins and P. Bhagwat, “Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers”, In Proceedings of the SIGCOMM 94, August 1994

[17] S.S. Tyagi and R.K. Chauhan, “Performance Analysis of Proactive and Reactive Routing Protocols for Ad hoc Networks”, International Journal of Computer Applications, Vol. 1, No. 4, 2010

IPv6 Wireless Sensor Network Gateway Design and End-to-End Performance Analysis

Gopinath Rao Sinniah, Zeldi Suryady,
Reza Khoshdelniat, Usman Sarwar, Mazlan Abbas
Wireless Communication Cluster
MIMOS Berhad
Kuala Lumpur, Malaysia
{*gopinath.rao, zeldi.suryady, reza.khoshdelniat, usman.sarwar, mazlan.abbas*}*@mimos.my*

Sureswaran Ramadass
National Advanced Centre of IPv6 (NAv6)
Universiti Sains Malaysia (USM)
Pulau Pinang, Malaysia
sures@nav6.usm.my

***Abstract*—The need for low power personal area network (LoWPAN) devices to be connected to the Internet is increasing due to the demand and proliferation of new applications. Previously, these devices didn't have the need to be connected to Internet. With the introduction of IPv6 over Low power personal area network (6LoWPAN) and the push for Internet of Things (IoT), these devices are now reachable using the common TCP/IP stack. Gateway is an important component to ensure that the packets from LoWPAN network are properly routed to the Internet. This paper provides a new gateway architecture to support 6LoWPAN stack and the performance analysis for end-to-end communication in an office environment. The architecture can be used for implementation in various 6LoWPAN related applications. Performance is measured on the latency and transmission success rate. The experiment results shows that the communication between WSN and client using the 6LoWPAN gateway is successful. Besides that, the success rate is 100% for 1 hop and slightly lower in 2 hops. The latency rate between 100 and 135 ms is acceptable and comparable with existing prior art which is 125 ms on average.**

Keywords-6LoWPAN; Wireless Sensor Network; Gateway; IPv6; 802.15.4.

I. Introduction

One of the growing sectors in wireless technology is IEEE802.15.4 Low Power Personal Area Network. IEEE802.15.4 is the basis for the ZigBee [13], WirelessHART [14], ISA100.11a [15] and MiWi [16]. These existing standards were created to provide connectivity in Personal Area Networks (PAN) area without having connectivity to the Internet. This is because small devices with low resources are thought to be incapable to have TCP/IP stack and also because the needs to be connected to the Internet were not matured.

Knowing the fact that existing TCP/IP is too heavy to be used in IEEE802.15.4 devices, 6LoWPAN [1] working grouping was created to provide a solution. The Working Group (WG) stated that the solution would be "pay as you use" header compression method that removes redundant or unnecessary network level information in the header. Some of the information can be derived from link-level IEEE802.15.4 header. Hence the 40 bytes IPv6 header was reduced to 2 bytes. This is achieved by reusing the link layer header information. The reduction of the header size is necessary as the total header size of IEEE802.15.4 is only 127 bytes which is too small to accommodate the entire 40 bytes IPv6 header.

The solution by the 6LoWPAN Working Group does not provide an end-to-end communication between the nodes and external devices in the Internet. It is because the header format of 6LoWPAN is different than the standard IP header. Therefore, a gateway or an intermediary device is required to provide a conversion between the 6LoWPAN and IP Header. The adaptation layer that is positioned between the link and network layer, provides the header compression for 6LoWPAN nodes. A gateway architecture was proposed [2] to provide solution for this. It provides an interface for communication between the IEEE802.15.4 nodes that uses 6LoWPAN stack to the external network that has interfaces such as WiMAX, Ethernet and WiFi. The gateway is configured so that it can process both the data that is sent periodically by the nodes and also request from the client. This paper adds contributions to [2], by providing the detail gateway architecture and performance analysis of the gateway. Figure 1 shows the overall communication scenario and the communication stacks between 6LoWPAN nodes, gateway and end user.

This paper is organized as follows. Section II reviews the existing solutions related to WSN gateways. Section III provides the gateway architecture and the communication between various components. Section IV discusses the implementation for the gateway, while Section V gives the experiments that were conducted and the results. Finally, conclusion and future research work are presented in Section VI.

ISBN: 978-1-61208-207-3

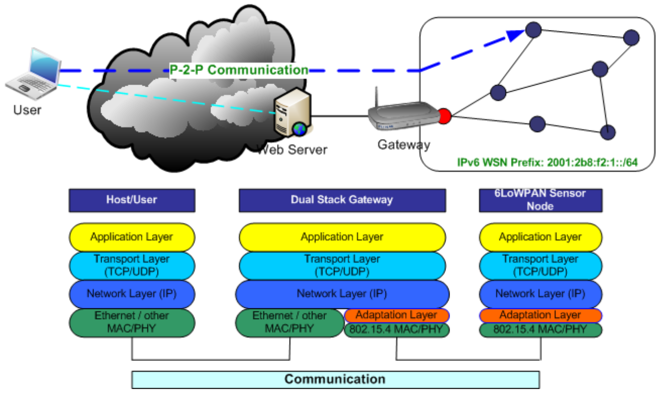


Figure 1. Interconnection between WSN nodes and end user and the communication stacks

II. Existing Solutions

There are several gateway architectures that were proposed for various implementation scenarios. One of the applications is for AC energy usage monitoring using 6LoWPAN. Jiang et al. [4] discussed the application that uses TinyOS and blip. Edge routers were used to route data to a database and uses a web server for visualization of the data. Wenbin et al. [5] developed WSN gateway specifically to monitor forest environment. Information from the sensors is sent to a monitoring centre using GPRS module. Jara et al. [6] introduced a WSN architecture that uses mobile nodes to collect healthcare information. 6LoWPAN gateway was used to connect the nodes to a database. All these solutions have little information on the gateway design and didn't provide the operation of the gateway.

Dun-Fan et al. [3] proposed gateway architecture for environmental monitoring which connects WSN with external network and shares the data collected using web services. The paper didn't explicitly mention the addressing of the nodes and communication between the nodes and external network. It is stated that ZigBee proprietary protocol was used for WSN. This is a drawback as the end users cannot directly communicate with the nodes.

There is an implementation [7] that uses 6LoWPAN short ID which is the MAC address. The internal node retrieves the destination MAC address by querying the gateway using the destination IPv6 address. Gateway retrieves link layer MAC address from the destination address provided by the internal node. This is unnecessary process as the nodes can directly send the data using the destination address and it is not practical as not all global address generated using link layer MAC address. Zimmermann et al. [8] introduces a one-to-one translation between link local address and global address at the gateway. They use a DNS-ALG like server to intercept the DNS query to assign link local address to internal node. If the DNS query could not be intercepted, communication would be disrupted.

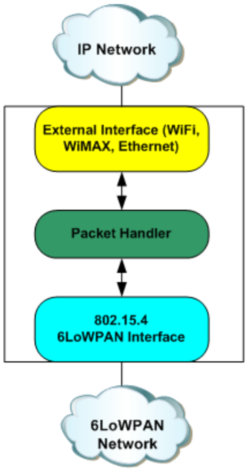

IP Network
External Interface (WiFi, WiMAX, Ethernet)
Packet Handler
802.15.4 6LoWPAN Interface
6LoWPAN Network

Figure 2. 6LoWPAN Gateway Modules

Jin et al. [12] proposed an interoperable architecture between NEMO and 6LoWPAN focusing on routing scheme. The nodes are configured with global IPv6 address and as such translation of header is not required. In our solution, we propose a solution for nodes that uses MAC address for communication. Besides that, we focus on the performance analysis on real testbed compared to the simulation results by [12].

III. Gateway Architecture

A complete end-to-end architecture could consist of sensor nodes, a gateway, database server or web server and end users. The gateway is designed to support two kinds of standard communications:

- Pull communication method - IPv6 clients request data from sensor node in 6LoWPAN network.
- Push communication method - Sensor nodes periodically send data to an external IPv6 device. The external IPv6 device in this system could be any remote station or server.

Based on dual stack protocol, the gateway is designed to have 3 modules as shown in Figure 2. The PHY/MAC for multiple interfaces that connect to external IP network is defined as external interface module, 6LoWPAN PHY and MAC layer is defined as 6LoWPAN Interface Module and all the services that might be implemented on top of adaptation layer which are network layer, transport layer and application layer reside in Packet Handler Module of the architecture.

The function of the three modules are explained briefly below.

- **6LoWPAN Interface (WSN) Module** - This module consists of IEEE802.15.4 compliance hardware which has the 6LoWPAN stack on it. The module is responsi-

ISBN: 978-1-61208-207-3

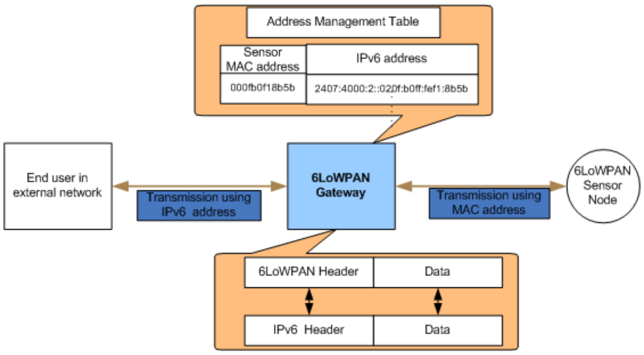


Figure 3. Address management table in the Gateway

ble for handling connectivity and data transmission of 6LoWPAN network using IEEE802.15.4 standard.

- **Interface Module** - This module defines the Physical and MAC layer of any interface that provides connectivity to external IP network. Therefore, the role of this module is to offer functionalities required to ensure connectivity to external public IP network. Some of the interfaces might provide connectivity to LAN/Wireless LAN (e.g., Wi-Fi), while others can provide connectivity to backhaul internet (e.g., Ethernet or WiMAX).
- **Packet Handler Module** - This module provide services to handle both 6LoWPAN and IPv6 packets. This is a significant module that bridges all the interfaces that connects to different networks. Since most of the main processes occur in this module, the service module has major responsibility integrating the 6LoWPAN network with the IP network through other external interfaces. The main purpose of this module is to provide functionalities for handling standard IPv6 packet from external network as well as 6LoWPAN packet. Both IPv6 and 6LoWPAN packets are analysed and processed accordingly. 6LoWPAN packets are transformed to IP packet and vice versa to enable smooth communication between 6LoWPAN nodes and external network.

Packets arrive at the gateway both from external network and 6LoWPAN network, would be first identified based on the address. If the packets are from external network, the gateway would find the destination address in the mapping table. The address would be translated into matching MAC address and the data is copied to 6LoWPAN header and sent to the node. If there is no matching address in the table, the request from the external network would be discarded. The same process is applied when a packet arrives from 6LoWPAN network. The MAC source address is replaced with the IPv6 address of the node and packet is sent using IPv6 header. The translation and mapping table is given in Figure 3.

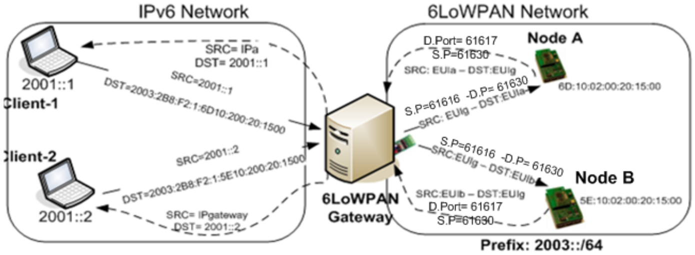


Figure 4. Gateway handling one-to-one communication

A. Gateway Communication

There are two scenarios for pulling sensor data; one client to one sensor node communication and many clients to one sensor node communication. For both the scenarios, a table is created to handle the packets that arrive at the gateway. Gateway maintains the entry in an *Address Information Table*, which will be used to route the sensor nodes' response packet back to the corresponding users. *Address Information Table* consists of ID number of packet, source address (Client's IPv6 address) and destination address (Sensor Node MAC address), port number allocated and the status. The status could be that the packets are already sent to sensor node, gateway already replied to client's request or and packet pending for transmission awaiting reply from sensor node for earlier requested information. Packets that are destined to the same node would be queued and not immediately transmitted to avoid collision. With the use of this table, retransmission of packets would be reduced and this will save energy in the nodes. An example of one-to-one communication is given in Figure 4. Different port numbers are used to differentiate the sensor's traffic from both the schemes. RFC 4944 [11] defines a well-known port range (61616-61631) for UDP packet in 6LoWPAN. In this implementation, the ports used are

- Port 61631 is used at the gateway to receive data from sensor nodes in push based method.
- Port 61616 is used by the gateway to send data to the sensor nodes in pull based mechanism.
- Port 61617 is used by the gateway to receive data from sensor nodes in pull based mechanism.
- Port 61630 is used by the nodes to receive the request from the external node through the gateway.

In the push method, sensor nodes send data to a fix destination port, 61631. All the data arrive at the gateway at that port would be automatically forwarded to a pre-configured destination address of a collector or database server. This is shown in Figure 5.

IV. Gateway Implementation

A testbed was created to validate the gateway architecture and to measure the end-to-end performance as shown in

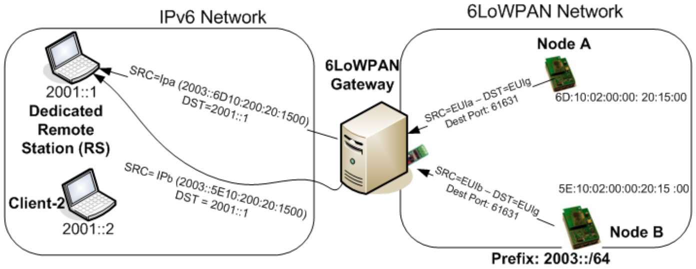


Figure 5. Pushing data from sensor to external node

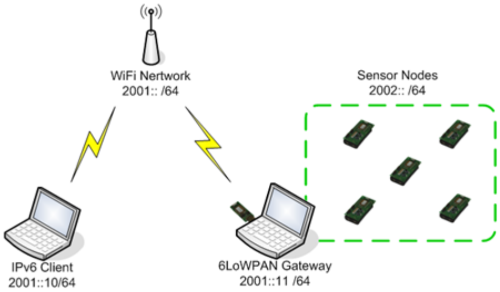


Figure 6. Testbed for validating and performance measurements

Figure 6. The tests were conducted in an indoor lab environment with over 20 active WiFi Access Points operational which is detected using a Network Stumbler software [11]. The sensor nodes that were deployed provide readings for temperature and light intensity measurements.

The setup consists of nano router and sensor nodes developed by Sensinode Inc. [9] as our hardware platform. Gateway is a laptop computer with Linux OS and has three interfaces; a nano router for the wireless sensor network and WiFi and Ethernet interface that connects to the IPv6 network. Nano router is a USB device that is attached to one of the available USB port in the gateway. Packet Handler module explained earlier is configured and executed on the gateway. The sensor nodes are installed with the free real-time operating system (FreeRTOS) with the NanoStack software module which consists of 6LoWPAN stack with added features. Each of the sensor node has 2 AA batteries. The modules were developed using c programming language. The communications for both push based and pull based schemes are maintained through the use of a gateway.

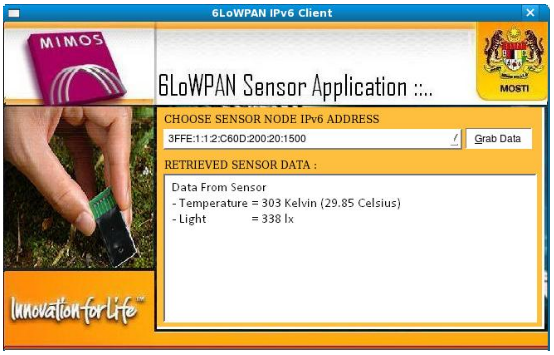
A client laptop was also used to retrieve sensor data to verify the bidirectional communication. To validate the performance, tests with different settings were conducted with different data sizes. Furthermore, to test the bidirectional communication, a ping message was sent from the gateway and using the reply, the latency was calculated. Table 1 provides the properties for the tests.

V. System Performance and Evaluation

As described earlier, the request from a client will be forwarded by the gateway using a simple client as shown in Figure 7 [2]. All the sensor nodes' IPv6 addresses are listed in the client and when a particular IPv6 address is selected, a request is forwarded to the gateway which will then do the necessary actions. The temperature and light reading from the sensor will then displayed on the client. This shows the success of bidirectional communication (Pull based mechanism). In the push based mechanism, the data is periodically sent to a web server and the data is displayed using a web browser as shown in Figure 8.

Table I
Performance Measurement Properties

Properties	Details
Network Size	4-8 nodes for 1 hop away. 2x2, 2x4 and 2x6 for 2 hops
Distance	3 meters for each hop
Data Sampling intervals	20 seconds
Duration	120 samples (1 hour)
Message size	4, 8, 16, 37 bytes
Measurements	Transmission Success Rate and Latency
Method	Start with 1 node and gradually increase the nodes while sending data simultaneously



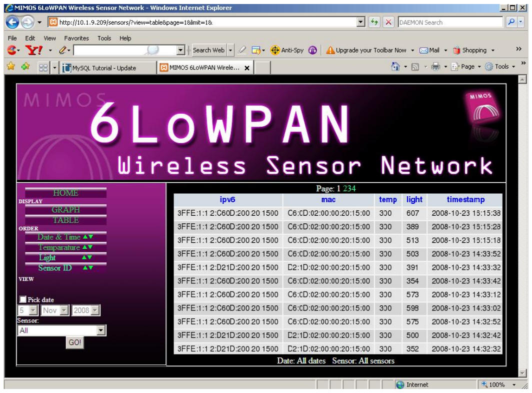
CHOOSE SENSOR NODE IPv6 ADDRESS

3FFE:1:1:2:C60D:200:20:1500

RETRIEVED SENSOR DATA :

Data From Sensor
- Temperature = 303 Kelvin (29.85 Celsius)
- Light = 338 lx

Figure 7. IPv6 client application to read data directly from sensors. ©2009 MIMOS Bhd. All Rights Reserved



http://10.1.9.209/sensors/?view=table&page=1&limit=1&

DAEMON Search

□ Pick date

5 Nov 2008

Sensor:

All

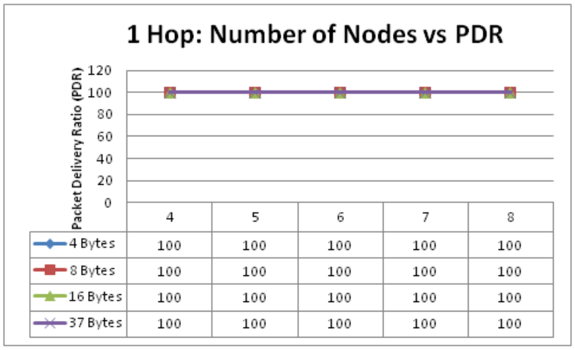
Page: 1 234

ipv6	mac	temp	light	timestamp
3FFE:1:1 2:C60D:200 20 1500	C6:CD:02:00:00:20:15:00	300	607	2008-10-23 15:15:38
3FFE:1:1 2:C60D:200 20 1500	C6:CD:02:00:00:20:15:00	300	389	2008-10-23 15:15:28
3FFE:1:1 2:C60D:200 20 1500	C6:CD:02:00:00:20:15:00	300	513	2008-10-23 15:15:18
3FFE:1:1 2:C60D:200 20 1500	C6:CD:02:00:00:20:15:00	300	503	2008-10-23 14:33:52
3FFE:1:1 2:D21D:200 20 1500	D2:1D:02:00:00:20:15:00	300	391	2008-10-23 14:33:32
3FFE:1:1 2:C60D:200 20 1500	C6:CD:02:00:00:20:15:00	300	354	2008-10-23 14:33:42
3FFE:1:1 2:C60D:200 20 1500	C6:CD:02:00:00:20:15:00	300	573	2008-10-23 14:33:12
3FFE:1:1 2:C60D:200 20 1500	C6:CD:02:00:00:20:15:00	300	598	2008-10-23 14:33:02
3FFE:1:1 2:C60D:200 20 1500	C6:CD:02:00:00:20:15:00	300	575	2008-10-23 14:32:52
3FFE:1:1 2:D21D:200 20 1500	D2:1D:02:00:00:20:15:00	300	500	2008-10-23 14:32:42
3FFE:1:1 2:D21D:200 20 1500	D2:1D:02:00:00:20:15:00	300	352	2008-10-23 14:32:32

Date: All dates Sensor: All sensors

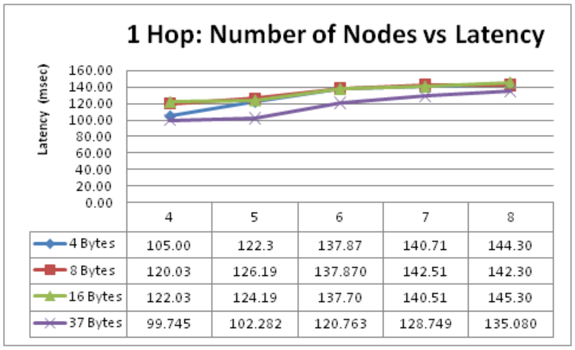
Figure 8. Display sensor information using web browser. ©2009 MIMOS Bhd. All Rights Reserved

ISBN: 978-1-61208-207-3



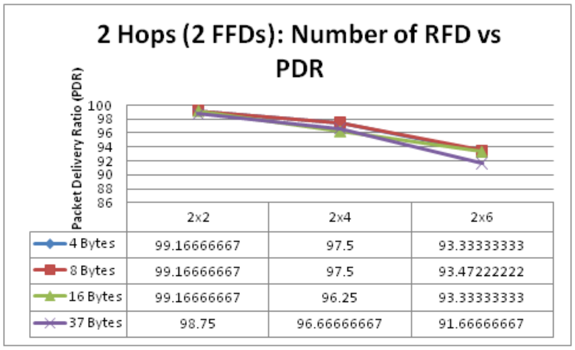
	4	5	6	7	8
4 Bytes	100	100	100	100	100
8 Bytes	100	100	100	100	100
16 Bytes	100	100	100	100	100
37 Bytes	100	100	100	100	100

Figure 9. Performance of packet delivery rate against number of nodes and different packet sizes for 1 hop



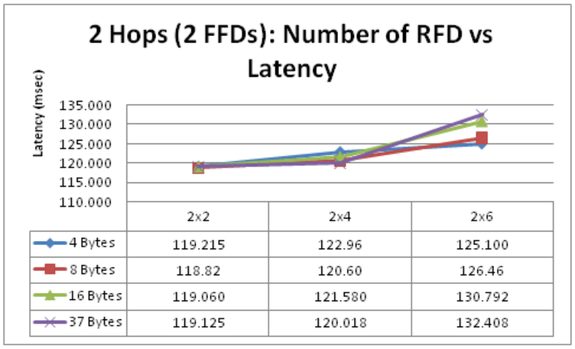
	4	5	6	7	8
4 Bytes	105.00	122.3	137.87	140.71	144.30
8 Bytes	120.03	126.19	137.870	142.51	142.30
16 Bytes	122.03	124.19	137.70	140.51	145.30
37 Bytes	99.745	102.282	120.763	128.749	135.080

Figure 11. Average latency for 1 hop with various numbers of nodes with different packet sizes



	2x2	2x4	2x6
4 Bytes	99.16666667	97.5	93.33333333
8 Bytes	99.16666667	97.5	93.47222222
16 Bytes	99.16666667	96.25	93.33333333
37 Bytes	98.75	96.66666667	91.66666667

Figure 10. Performance of packet delivery rate against number of nodes for 2 hops



	2x2	2x4	2x6
4 Bytes	119.215	122.96	125.100
8 Bytes	118.82	120.60	126.46
16 Bytes	119.060	121.580	130.792
37 Bytes	119.125	120.018	132.408

Figure 12. Average latency for 2 hops with various numbers of nodes with different packet sizes

Figure 9 shows the results of packet delivery rate against the number of nodes and data sizes. There are no changes with the increase of nodes for various packet sizes in 1 hop scenario. All the data transmitted was successfully received. However, with the increase in the number of hops, the packet delivery rate dropped as shown in Figure 10. It could be observed that the efficiency also dropped with the increase of packet size. This could be due to the relay node (Full Function Device(FFD))that was used could not handle the packets properly. It could also be because of the high number of operational access points which shares the same frequency as IEEE802.15.4 in the lab. To verify this, experiments were repeated outside the building, in an open space that does not have any access points coverage. The results obtained in outdoor environment were similar to the indoor environment. This proofs that the cause of packet drop is due to the relay node.

In order to calculate the latency, the client sends a ping message to a specific sensor node and round trip time (RTT) retrieved. From the RTT value, average per-hop latency is calculated. The average latency for 1 hop and 2 hops are given in Figure 11 and 12. The base latency which is latency measured with only one node active is about 65 milliseconds for 1 hop and 90 milliseconds for 2 hops. It can be observed that with the increase in the number of active nodes, the latency increased and ranges from 100 to 135 milliseconds. This average latency is comparable with average latency claimed in the white paper by IPSO-Alliance [10] which is about 125 milliseconds. Total latency is calculated based on the processing latency of packet at the node, processing latency at the network gateway or router and latency due to network condition. Major contribution of the latency is the wireless network condition such obstacles, interference of signal from other devices and others. We noticed that there is a slight decrease in latency when the experiments conducted in an open space without any other interference. Besides the two measurements, the average power consumption to transmit data every 20 seconds interval is about 0.015 volts. This was obtained by sending the available power in the battery as data. The difference between of power reading between 2 intervals were calculated and averaged. Based on the tests conducted, the average total consumption of battery before it malfunction is about 1.5 volts. This can be used to calculate the battery life of a sensor node in various types of implementations.

VI. Conclusion and Future Work

This paper proposed a gateway architecture to interconnect wireless sensor network with external network using

ISBN: 978-1-61208-207-3

6LoWPAN protocol. The gateway provides the mechanism for the end clients to directly communicate with the sensor node which was assigned with IPv6 address. Besides that the gateway forwards the periodical data to a web server.

The architecture is validated with the successful transmission of sensor data which was displayed using a client and web server. Further tests were conducted to validate the latency and the transmission success rate. The latency for 1 hop with various number of nodes ranges between 100 to 135 milliseconds while the transmission success rate is 100 % for 1 hop. The success rate drop with the increase of number of hop which could be because of the relay node (FFD) not forwarding the packets appropriately. Nevertheless the results are in accordance with the other prior art.

As future work, the proposed solution can be further tested in other environments by setting different transmission intervals, less interferences, etc. The performance can also be evaluated with the implementation of other components such as security, routing and mobility with multi-hop scenarios.

REFERENCES

[1] IPv6 over Low Power Personal Area Network (6LoWPAN) IETF Working Group. Retrieved: July, 2012. http://datatracker.ietf.org/wg/6lowpan/

[2] G. R. Sinniah, Z. Suryady, U. Sarwar, M. Abbas, “A Gateway Solution for IPv6 Wireless Sensor Network”, Ultra Modern Telecommunication & Workshops, St. Petersburg, Russia. pp 1-6, October 2009.

[3] Y. Dun-Fan, M. Liang-liang, and W. Wei, “Design and Implementation of Wireless Sensor Network Gateway Based on Environmental Monitoring”, 2009 International Conference on Environmental Science and Information Application Technology, pp. 289-292, Jul. 2009.

[4] X. Jiang, S. Dawson-haggerty, P. Dutta, and D. Culler, “Design and Implementation of a High-Fidelity AC Metering Network”, IPSN ’09: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks. Washington, DC, USA: IEEE Computer Society pp. 253-264, 2009.

[5] L. Wenbin, C. Dongxu, and Z. Junguo, “Design and Implementation of Wireless Sensor Network Gateway Faced to Forest Information Monitor”, 2010 International Conference on Intelligent System Design and Engineering Application, pp. 524-526, Oct. 2010.

[6] A. J. Jara, M. a. Zamora, and A. F. G. Skarmeta, “HWSN6: Hospital Wireless Sensor Networks Based on 6LoWPAN Technology: Mobility and Fault Tolerance Management”, 2009 International Conference on Computational Science and Engineering, pp. 879-884, 2009.

[7] D. Choi, J.-tak Park, and S.-yoen Kim, “IPv6 global connectivity for 6LoWPAN using short ID”, 2011 International Conference on Information Networking (ICOIN), 2011, pp. 384-387, 2011.

[8] A. Zimmermann and J. S. Silva, “6glad: IPv6 global to link-layer address translation for 6lowpan overhead reducing”, Next Generation Internet Networks, NGI 2008, pp. 209-214, 2008.

[9] “Sensinode hardware and NanoStack Operating System”, 2008. Retrieved: July, 2012. Available: http://www.sensinode.com/

[10] J. Abeill, M. Durvy, J. Hui, S. Dawson-Haggerty. “Lightweight IPv6 Stacks for Smart Objects: the Experience of Three Independent and Interoperable Implementations”, November 2008. Available at: http://www.ipso-alliance.org/white-papers

[11] “Network Stumbler”. Retrieved: July, 2012. Available: http://www.netstumbler.com/

[12] H. K. Jin, S. H. Choong and O. Koji, “A Routing Scheme for Supporting Network Mobility of Sensor Network Based on 6LoWPAN”, APNOMS ’07, LNCS 4773, pp. 155-164, 2007.

[13] ZigBee Alliance. “ZigBee Specifications”. Retrieved: July, 2012. http://www.zigbee.org/Specifications.aspx

[14] Hart Communication Foundation. “WirelessHART Overview”. Retrieved: July, 2012. http://www.hartcomm.org/protocol/about/aboutprotocol_specs.html

[15] The International Society for Automation. “ISA100, Wireless Systems for Automation”. Retrieved: July, 2012. http://www.isa.org/isa100

[16] Microchip Technology Inc. “MiWi Development Environment”. Retrieved: July, 2012. http://www.microchip.com/miwi

COPlanner: A Wireless Sensor Network Deployment Planning Architecture Using Unmanned Vehicles As Deployment Tools

Ramin Soleymani-Fard, Chia-Yen Shih, Marvin Baudewig and Pedro José Marrón
Networked Embedded Systems Group
Department of Computer and Cognitive Sciences
Universität Duisburg-Essen
Duisburg, Germany
{*ramin.soleymani-fard, chia-yen.shih, baudewig, pjmarron*}*@uni-due.de*

***Abstract*—Wireless sensor networks (WSNs) have been deployed for a variety of applications. As the scale of WSN deployments has largely increased and the application scenarios have become more complex, WSN deployment planning can save unnecessary expense on redundant hardware, software and human resources and thus makes the deployment more efficient. This paper presents an ongoing research on developing a deployment planning architecture, called *COPlanner*. Our goal is to provide deployment planning strategies for the static and mobile WSN applications to meet their requirements with respect to sensing coverage, network connectivity and data collection. Most importantly, for the inaccessible area, we consider using *Unmanned Aerial or Ground Vehicles* (UAVs or UGVs) as the deployment tools. Therefore, the planning architecture also covers the *Waypoint Planning* problem and offers scheduled routes for the autonomous vehicles to deploy sensor nodes and collect the data.**

Keywords-deployment planning; sensing coverage; connectivity; unmanned vehicles; waypoint planning; obstacle avoidance.

I. Introduction

Wireless sensor networks (WSNs) have been deployed in physical environments (both indoor and outdoor areas) for a wide range of applications including habitant monitoring, disaster management, inventory control, etc. A WSN typically consists of a set of static or mobile sensor nodes, or so-called *Cooperating Objects* (COs) [1], which provide physical measurements and collaboratively carry out system operations to achieve the application objectives.

As the scale of current CO deployments becomes considerably large, e.g., thousands of sensor nodes and the application scenarios become more complex, effective and thorough pre-deployment planning is necessary and is the key to reduce the deployment cost. To provide a useful deployment plan, the developer must consider several aspects when designing an architecture for a WSN deployment tool. First, the planning architecture needs to allow the user to specify the application objectives, deployment requirement and the target area characteristics as well as the constraints for the deployment. Second, for most WSN applications with static nodes the main deployment requirements are centered around the issues of sensing coverage and network connectivity [2] [3]. Therefore, the architecture must offer a set of deployment planning strategies [4] [5] [6] and parameterization for optimizing the CO deployments in order to meet the application requirement. As for the application with mobile nodes [7], the architecture must provide the movement schedules as well as the trajectories for the mobile nodes to perform system operations in order to achieve required application performance. Third, the most important things is that the planning architecture has to include an evaluation unit for assessing the application performance based on the resulting CO deployments and to output the optimal deployment plan for the application.

We have been developing a WSN deployment planning architecture called *COPlanner*, which aims to cover the above aspects regarding static and mobile sensor node deployments. Moreover, COPlanner provides realistic deployment strategies by taking into account the physical obstacles. Furthermore, COPlanner considers using UAVs/UGVs as deployment tools for the inaccessible target areas. Therefore, the deployment plan generated by COPlanner also tackles the *Waypoint Planning* problem, which involves defining the optimal route for the deployment vehicles to visit a given set of waypoints as the deployment locations. COPlanner is developed in the scope of an European project, *PLANET* [8], in which the deployment planning tool is required to support the CO deployment using UAVs/UGVs for two applications: the *Wildlife Monitoring* in *Doñana Biological Reserve* (DBR) [9] and the *Automated Airfield* applications.

The remainder of the paper is structured as follows. We list the related work in Section II. Section III presents our architecture design of the COPlanner and it compositional components; Section IV describes the planning strategies we have developed for the static WSNs regarding sensor coverage and network connectivity, and for the mobile networks regarding waypoint planning. Finally, we describe our future work and conclude our work in Section V.

II. Related work

There have been lots of deployment strategies proposed to achieve sensing coverage and network connectivity. Many approaches are based on virtual forces, with which the nodes are either attracted or repulsed in order to achieve required coverage or connectivity. In [10], the distribution of sensing events was used to generate virtual forces, which shape the network topology to adapt the distribution of expected sensing events. Other grid-based approaches [5] [11] divided the target area into subareas and arranged the node locations in order to fulfill the deployment requirement. While these approaches aim to solve the deployment issues, they do not consider the impact of obstacles. All of our deployment strategies deal with the obstacle issue. Tan et al. [12] and Wang et al. [5] proposed algorithms for the sensor deployment to cover the target area, which can also include obstacles. Both approaches tried to minimize the number of nodes to be deployed. However, these approaches do not maintain k-connectivity.

While many deployment strategies are available, little work has been found on the WSN deployment planning environment. Li et al. described a planning platform [6] that guides the developer through three steps for deployment planning: *Pre-Placement*, *Simulation/Evaluation* and *Optimization*. The aim of the platform is to provide an integrated, general deployment planning environment. However, the deployment strategies for various applications were not addressed. The goal of COPlanner is to provide different deployment strategies to meet various application requirements. Therefore, more development efforts spent on the optimization deployment approaches for different network deployment configurations.

III. Architecture

The main functionality of COPlanner is to, given the user application input, create efficient deployment plans for various static and mobile WSNs applications. Figure 1 depicts an overview of the COPlanner architecture. In our design, the user can specify application-specific input and obtain the final deployment plan through a web-based application, so a web service client component is devised to handle the interaction with the web server. The user application input is maintained by the *User Application* (UA) component. Such input includes the target area description with the deployment constraints, application task description, planning configuration and sensing device configuration. Each piece of information is maintained by a corresponding subcomponent associated with the UA component.

The core component of COPlanner is the *Planning Manager*, which decides the type of the deployment plan to be generated based on the application input. Once the plan type is determined, the manager interacts with the *Planning Tool Manager* (PTM), which is implemented with a powerful scheme that allows flexibly extending the COPlanner's capability with newly developed deployment planning algorithms/strategies. PTM manages a set of *Planning Tools*, each of which implements an optimization deployment algorithm to meet the specific user requirements on the application performance, e.g., data delivery rate, latency and system lifetime. Section IV will detail these strategies.

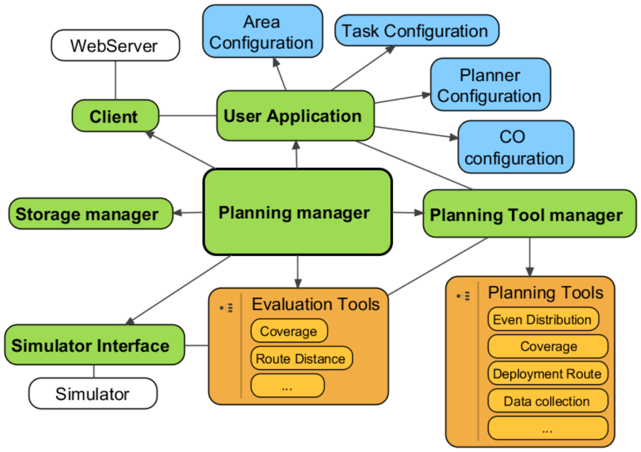


Figure 1. The Architecture of COPlanner

The planning tool outputs one or a set of optimized pre-deployment plans created by the optimization algorithm. PTM then interacts with the *Evaluation Tools* component, which includes different performance metrics for evaluating the application performance based on the planned deployments. The evaluation of the pre-deployment plans is performed through the use of a WSN simulator. With the *Simulation Interface* component, the Evaluation Tool can specify the performance metrics and can request the simulator to simulate the application. From the application performance results with different pre-deployment plans, the Evaluation Tool decides on the *best* candidate as the output of the deployment plan. The final deployment plan as well as the performance evaluation is stored in the *Storage Manager* component. Note that the discussion on the simulator is not in the scope of the paper, and we focus our discussion on the architecture of COPlanner and the deployment strategies developed for generating optimized deployment plans.

IV. Planning Strategies

The design goal of COPlanner is to provide deployment strategies, which specify the node deployment locations for the WSN required by various applications. Our deployment strategies are classified into two kinds: (1) one for the static WSN deployments and (2) the other for the deployments involved in mobile vehicles, which can be further categorized into mobile sensor nodes and the deployment tools. These strategies are detailed below.

A. Static WSN Deployment Planning Strategies

Our deployment strategies for the static WSN deployment focus on the issues of sensing coverage and network connectivity with the minimum number of nodes. Moreover,

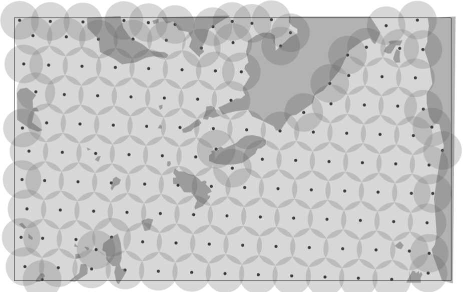


Figure 2. An example of generated *covered* deployment plan

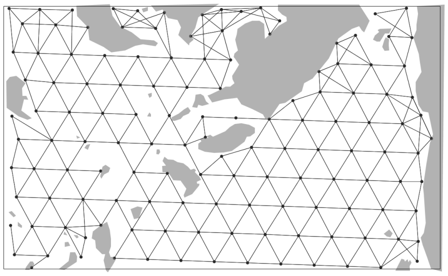


Figure 3. An examples of generated *connected* deployment plan

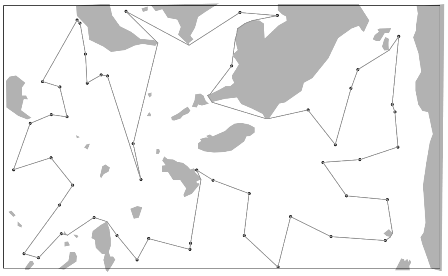


Figure 4. An example of generated route by WPP-O

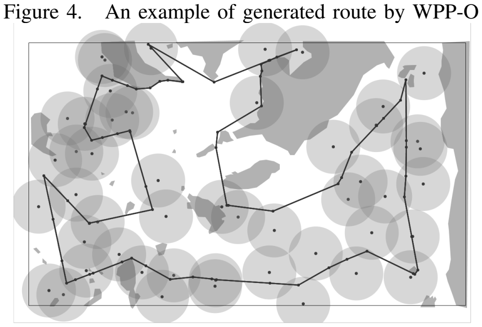


Figure 5. An example of generated route for data collection

different from the typical coverage and connectivity algorithms, our approach considers the existence of obstacles in the target area. In COPlanner, the obstacles are modeled as polygons, in which no deployment locations can be specified. We use the *Unit Disc Model* (UDM) for the sensing and communication coverage. We assume that two nodes A and B are *connected* if their Euclidean distance $|AB| \leq r_c$, where r_c is the communication range. However, A and B are *disconnected* if the segment $\overline{AB}$ intersects with any obstacle polygon. Note that our approaches, except for the CCP approach (described below), are not limited by the use of UDM. To simply the explanation without loss of generality, we use the UDM to show the resulting deployment plans.

We first developed the *Coverage and Connectivity Planning* (CCP) strategy, which generates a deployment plan with node deployment locations such that the every point (except for the ones in the obstacles) in the target area is covered, while the nodes stay connected. To achieve the minimum node number, we used the approach proposed in [4], in which the author analyzed the patterns of node locations in order to maintain *full coverage* and *k-connectivity* ($k \leq 6$) with the minimum number of nodes. However, when considering obstacles, the regularity of the pattern cannot be totally applied. We modified the approach as follows. Initially, our CCP approach uses the *hill climbing* technique to search for the pattern configuration, which identifies the starting node location and the angle for applying the pattern across the target area. In the case where the deployment locations fall into the obstacle ranges, the deployment *holes* will occur. To cope with this problem, CCP uses a heuristic algorithm to fill up the holes with the minimum number of nodes. Furthermore, if k-connectivity is required, additional nodes will be included to match the requirement. Figure 2 and Figure 3 illustrate the examples of resulting deployment plans for coverage and connectivity with the coverage radius $r_s = 50m$ and $r_c = 120m$, respectively.

Another deployment strategy that is currently in development involves evenly distributing the nodes across the target area. The motivation for this approach is the applications that requires sampling at different locations in the target area, e.g., water samples from different locations of a flooded area for pollution detection. Connectivity is not main requirement for such application and the data can be collected, e.g., using mobile elements. Wang describes in [13] a vector-based approach called Minimax Algorithm, which creates a uniformly distributed network topology. The main idea is to use a *Voronoi* diagram and the *minimax* points within the Voronoi cells to organize the originally randomly deployed nodes until the Voronoi cells have approximately the same size. The next step of our work is to adjust the algorithm so that it considers obstacles in the target area.

B. Mobile WSN Deployment Planning Strategies

In addition to the static WSN deployments, COPlanner also covers deployment planning for the mobile WSN applications. Moreover, COPlanner considers using the moving vehicles as the deployment tools. In this case, the deployment plan contains the locations to be visited and the movement schedules for the deployment vehicles and the

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-207-3

mobile nodes. The former can be generalized to the *Waypoint Planning* problem, in which given a set of waypoints, the algorithm outputs the optimal route to visit all the waypoints.

We developed a solution algorithm for the extended Waypoint Planning problem with the obstacles (WPP-O) and also modeled the obstacles as polygons in this case. In addition to the waypoints, our approach first identifies each vertex of the polygons and includes them in the graph G. The algorithm then finds the shortest paths between each pair of nodes in G using our modified Dijkstra algorithm. In the next step, our approach randomly selects a route that covers all waypoints, and tries to find the optimal route using the *Traveling Salesman Problem (TSP) with Simulated Annealing* technique, which assigns the distance as a cost value to each link. The goal is to create a route with the minimum cost value. In each iteration, a pair of links are switched to form a new route. The switch is accepted if the resulting cost is lower or with a certain probability if the cost is higher. When the iterations terminates, our algorithm outputs an optimized route that covers all waypoints while the obstacles are avoided. Figure 4 shows an example of the optimal route generated by our WPP-O algorithm.

Additionally, we are currently developing a deployment strategies for data collection with mobile nodes. The problem of data collection using mobile nodes is defined as: given a set of node locations and the communication range, the algorithm outputs a route for the mobile node to pass through the communication range of every node in order to collects all sensory data. The objective is also to minimize the distance of the route. Figure 5 illustrates an example of resulting route. More improvement on this approach still needs to be made in order to create the optimal route.

V. Conclusion and Future work

In this paper, we presented our ongoing work on a flexible deployment planning architecture (COPlanner) that can be easily extended to include different deployment strategies for various WSN applications using the static or mobile sensor nodes. Particularly, COPlanner provides deployment plans that involve in autonomous vehicles as the deployment tools. We believe that the development of COPlanner can provide a useful WSN deployment planning architecture to ease the WSN deployment task and to reduce the deployment cost.

In the future, we plan to perform thorough experiments to evaluate the developed algorithms, and further to use these techniques on the real deployment in the project. Moreover, we will consider more deployment approaches for more complex applications that required deployments of hybrid WSNs with both static and mobile nodes.

Acknowledgment

This work has been partially supported by EU PLANET, *PLAtform for the deployment and operation of heterogeneous NETworked cooperating objects*, and CONET, *the Cooperating Objects Network of Excellence* (PLANET: FP7-2009-5-257649, CONET: FP7-2007-2-224053).

References

[1] P. J. Marrn, S. Karnouskos, D. Minder, and the CONET consortium, *Research Roadmap on Cooperating Objects*. Office for Official Publications of the European Communities, 2009.

[2] C.-F. Huang and Y.-C. Tseng, “The coverage problem in a wireless sensor network,” *Mob. Netw. Appl.*, vol. 10, no. 4, pp. 519–528, Aug. 2005.

[3] A. Konstantinidis and K. Yang, “Multi-objective k-connected deployment and power assignment in wsns using a problem-specific constrained evolutionary algorithm based on decomposition,” *Comput. Commun.*, vol. 34, pp. 83–98, Jan. 2011.

[4] X. Bai, D. Xuan, Z. Yun, T. H. Lai, and W. Jia, “Complete optimal deployment patterns for full-coverage and k-connectivity (k<=6) wireless sensor networks,” in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, 2008, pp. 401–410.

[5] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, “Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks,” in *Proceedings of the First International Conference on Wireless Internet*, 2005, pp. 114–121.

[6] J. Li, Y. Bai, H. Ji, J. Ma, Y. Tian, and D. Qian, “Power: Planning and deployment platform for wireless sensor networks,” in *Grid and Cooperative Computing Workshops, GCCW ’06. Fifth International Conference on*, 2006, pp. 432–436.

[7] Y. Wang and C.-H. Wu, “Robot-assisted sensor network deployment and data collection,” in *Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on*, june 2007, pp. 467 –472.

[8] The PLANET project, PLAtform for the deployment and operation of heterogeneous NETworked cooperating objects. [Online]. Available: www.planet-ict.eu

[9] Doñana Biological Reserve. [Online]. Available: http://www.ebd.csic.es/website1

[10] C. Koutsougeras, Y. Liu, and R. Zheng, “Event-driven sensor deployment using self-organizing maps,” *Int. J. Sen. Netw.*, vol. 3, pp. 142–151, May 2008, printed.

[11] J.-J. Chang, P.-C. Hsiu, and T.-W. Kuo, “Search-oriented deployment strategies for wireless sensor networks,” in *Object and Component-Oriented Real-Time Distributed Computing. 10th IEEE International Symposium on*, 2007, pp. 164–171.

[12] G. Tan, S. A. Jarvis, and A.-M. Kermarrec, “Connectivity-guaranteed and obstacle-adaptive deployment schemes for mobile sensor networks,” *Distributed Computing Systems, 2008. ICDCS ’08. The 28th International Conference on*, vol. 8, no. 6, pp. 836–848, 2008.

[13] G. Wang, G. Cao, and T. La Porta, “Movement-assisted sensor deployment,” in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4, 2004, pp. 2469–2479.

An Efficient Tag Identification Scheme in RFID Systems

Chiu-Kuo Liang, Chih-Kang Tseng, and Hsin-Mo Lin
Dept. of Computer Science and Information Engineering, Chung Hua University
Hsinchu, Taiwan, R.O.C.
Email: {ckliang, e09902014, e09902004}@chu.edu.tw

***Abstract*—RFID is a generic term for technologies which use RF waves to identify, track, or categorize any object. One of the research areas in RFID systems is a tag anti-collision protocol; how to reduce identification time with a given number of tags in the field of an RFID reader. There are two types of tag anti-collision protocols for RFID systems: tree based algorithms and slotted aloha based algorithms. Many anti-collision algorithms have been proposed in recent years, especially in tree based protocols. However, there still exist challenges on enhancing the system throughput and stability due to the underlying technologies having faced different limitation in system performance when network density is high. Recently, a *Wrap-Around Scan* (*WAS*) technique, which is a tree based approach, was proposed and aims to speedup tag identification in large scale RFID systems. The main idea of the Wrap-Around Scan is to limit the number of collisions by moving to next level as the number of collisions exceeds a predefined threshold. The WAS method indeed improves the efficiency in high density RFID systems. However, the reader using WAS scheme will spend some unnecessary queries for the idle response. In this paper, we proposed an intelligent wrap-around scan technique, which is called the iWAS scheme, to avoid those unnecessary queries. The simulation results show that the proposed techniques provide superior performance in high density environments. It is shown that the iWAS is effective in terms of increasing system throughput and minimizing identification delay.**

Keywords-Tag anti-collision; query tree; wrap-around scan.

I. INTRODUCTION

Radio Frequency IDentification (RFID) is an emerging technology that guarantees to advance modern industrial practices in object identification and tracking, asset management, and inventory control [14]. Recently, several identification systems such as barcodes and smart cards are incorporated for automatic identification and data collection. However, these systems have several limits in read rate, visibility, and contact. RFID systems are a matter of grave concern because they provide fast and reliable communication without requiring physical sight or touching between readers and tags.

One of the areas of research is the speed with which a given number of tags in the field of RFID readers can be identified. For fast tag identification, anti-collision protocols, which reduce collisions and identify tags irrespective of occurring collisions, are required [6], [7]. There are two types of collisions: reader collisions and tag collisions. Reader collisions indicate that when neighboring readers inquire a tag concurrently, so the tag cannot respond its ID to the inquiries of the readers. These collision problems can be easily solved by detecting collisions and communicating with other readers. Tag collisions occur when multi tags try to respond to a reader simultaneously and cause the reader to identify no tag. For low-cost passive RFID tags, there is nothing to do except response to the inquiry of the reader. Thus, tag anti-collision protocols are necessary for improving the cognitive faculty of RFID systems.

In general, the tag anti-collision techniques can be classified into two categories, aloha based and tree based protocols. Aloha-based approaches use time slot to reduce collision probability. Tags randomly select a particular slot in the time frame, load and transmit its identification to the reader. Once the transmission is collided, tags will repeatedly send its id in next interval of time to make sure its id is successfully recognized. Aloha based protocols can reduce the collision probability. However, they have the tag starvation problem that a particular tag may not be identified for a long time. For the consideration of performance, the tag collision rate is increased as the number of RFID tag increased, which may result a low tag recognition rate.

The tree based schemes use a data structure similar to a binary search algorithm. An RFID reader consecutively communicates with tags by sending prefix codes based on the query tree data structure. Only tags in the reader's interrogation zone and of which ID match the prefix respond. The reader can identify a tag if only one tag respond the inquiry. Otherwise the tags responses will be collided if multiple tags respond simultaneously.

Although tree based protocols do not bring the tag starvation problem, but they have relatively long identification delay. Recently, a Wrap-Around Scan (WAS) technique was proposed and aiming to coordinate simultaneous communications in high density RFID environments, to speedup tag identification and to increase the overall read rate and throughput in large-scale RFID systems [15]. The main idea of the Wrap-Around Scan technique is to limit number of collisions during the identification phase. When number of collisions larger than the predefined acceptable ratio, it reveals that the density in RF field is too high. In order to minimize unnecessary inquiry, the prefix matching will be moved to lower level of the query tree, alleviating the collision problems. The method of setting collision bound indeed improves the efficiency of large-scale RFID tag identification. However, the reader using WAS scheme will spend some unnecessary queries for the idle response. In this paper, we proposed an *intelligent Wrap-Around Scan technique* (i*WAS*) to avoid those unnecessary queries. To evaluate the performance of our proposed technique, we have implemented our proposed iWAS scheme along with previous proposed WAS method and the query tree protocol. The experimental results show that the proposed technique presents significant improvement in most circumstance.

The remainder of this paper is organized as follows: Related work is discussed in Section II. In Section III, the tree based tag identification algorithm is introduced as preliminary of this study. In Section IV, our proposed query tree based algorithm, the intelligent Wrap-Around Scan (iWAS) is presented. Performance comparisons and analysis of the proposed technique will be given in Section V. Finally, in Section VI, some concluding remarks are made.

II. Related work

Many research results for collision avoidance have been presented in literature. The existing tag identification approaches can be classified into two main categories, the Aloha-based [3], [4], [8], [11], [14] anti-collision scheme and the tree-based scheme [1], [7], [9], [13]. RFID readers in the former scheme create a frame with a certain number of time slots, and then add the frame length into the inquiry message sent to the tags in its vicinity. Tags response the interrogation based on a random time slot. Because collisions may happen at the time slot when two or more tag response simultaneously, making those tags could not be recognized. Therefore, the readers have to send inquiries contiguously until all tags are identified. As a result, Aloha-based scheme might have long processing latency in identifying large-scale RFID systems [4]. In [14], Vogt et al. investigated how to recognize multiple RFID tags within the reader's interrogation ranges without knowing the number of tags in advance by using framed Aloha. A similar research is also presented in [11] by Zhen et al. In [3], Klair et al. also presented a detailed analytical methodology and an in-depth qualitative energy consumption analysis of pure and slotted Aloha anti-collision protocols. Another anti-collision algorithm called enhanced dynamic framed slotted aloha (EDFSA) is proposed in [5]. EDFSA estimates the number of unread tags first and adjusts the number of responding tags or the frame size to give the optimal system efficiency.

In tree-based scheme, such as ABS [7], IBBT [2] and IQT [10], RFID readers split the set of tags into two subsets and labeled them by binary numbers. The reader repeats such process until each subset has only one tag. Thus the reader is able to identify all tags. The adaptive memoryless tag anti-collision protocol proposed by Myung et al. [6] is an extended technique based on the query tree protocol. Choi et al. also proposed the IBBT (Improved Bit-by-bit Binary-Tree) algorithm [2] in Ubiquitous ID system and evaluate the performance along three other old schemes. The IQT protocol [10] is a similar work approach by exploiting specific prefix patterns in the tags to make the entire identification process. Recently, Zhou et al. [12] consider the problem of slotted scheduled access of RFID tags in a multiple reader environment. They developed centralized algorithms in a slotted time model to read all the tags. With the fact of NP-hard, they also designed approximation algorithms for the single channel and heuristic algorithms for the multiple channel cases.

Although tree based schemes have advantage of implementation simplicity and better response time compare with the Aloha based ones, they still have challenges in decreasing the identification latency. In this paper, we present an enhanced tree based tag identification technique aims to coordinate simultaneous communications in large-scale RFID systems, to speedup minimize tag identification latency and to increase the overall read rate and throughput. Simulation results show that our proposed technique outperforms previous techniques.

III. Tree Based Tag Anti-Collision Schemes

In this section, we present two tree based anti-collision techniques, namely the Query Tree algorithm (QTA) and the Wrap-Around Scan algorithm (WAS), that are most related to our work.

A. Query Tree Algorithm

The query tree algorithm (QT) uses binary splitting strategy to identify tags. A reader transmits the k-length prefix. Then tags send from $(k + 1)th$ bit to the end bit of tag IDs if the first k bits of tag IDs are the same as the prefix. Also, if the received tag IDs collide, the extended prefix attached '0' or '1' to the prefix is retransmitted. Furthermore, if there is no collision, the reader identifies one of the tags. Figure 1 shows an example of the query tree scheme. Table 1 summarizes the detail steps of communication between the reader and the tags with the example shown in Figure 1.

B. Wrap-Around Scan algorithm

In the environment with high tags density, collision may happen very frequently while using the query tree algorithm, and due to that, a lot of query time will be wasted. The main idea of Wrap-Around Scan algorithm is using a threshold jumping strategy in which a threshold of collisions is mainly used as the criterion of moving the identification process to the next level of a query tree. By using the collision threshold, when the collisions happen frequently in a level, the corresponding level of the query tree will be jumping over. As shown in Figure 2, when the number of collision exceeds the predefined threshold, prefix matching at the present level will be skipped in order to save unnecessary prefix inquires for the remaining part at the same level. Through the process, the number of inquiry message could be significantly reduced. The collision threshold in Wrap-Around Scan technique is set to $2^I/M$, where I denotes the level in the query tree and M is a pre-defined constant.

Let us use an example to demonstrate the WAS scheme. Figure 3 shows an example of an RFID network having 6 tags with 4-bits id length using WAS with $M = 3$. The WAS scans the query tree from left to right in level 1, the reverse in level 2, and reverse again in level 3, and so on. Since $M = 3$, the threshold of collisions are 1, 2 and 3 in level 1, level 2 and level 3, respectively. The only one collision happened in

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-207-3

level 1 is node 0, the two collision nodes appeared in level 2 are nodes 10 and 11; while the three collision nodes in level 3 are happened at nodes 011, 101 and 110 in order. Table 2 summarizes the detail steps of communication between the reader and the tags with the example shown in Figure 3.

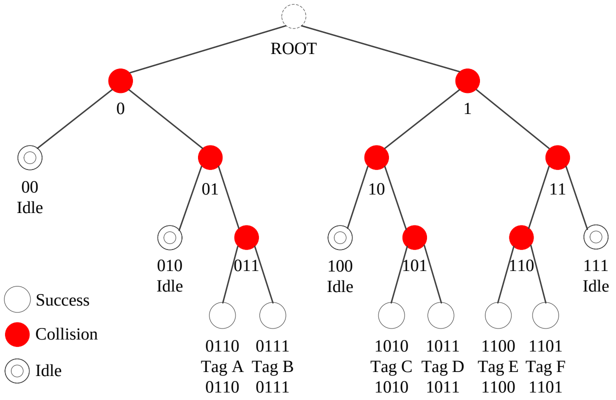


Figure 1. An example of query tree algorithm.

TABLE I. COMMUNICATION STEPS OF FIGURE 1.

Step	Broadcast	Status
1	0	Collision
2	1	Collision
3	00	Idle
4	01	Collision
5	10	Collision
6	11	Collision
7	010	Idle
8	011	Collision
9	100	Idle
10	101	Collision
11	110	Collision
12	111	Idle
13	0100	Identify Tag A
14	0111	Identify Tag B
15	1010	Identify Tag C
16	1011	Identify Tag D
17	1100	Identify Tag E
18	1101	Identify Tag F

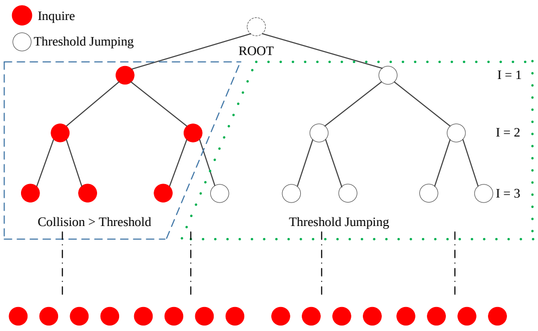


Figure 2. The paradigm of threshold jumping technique.

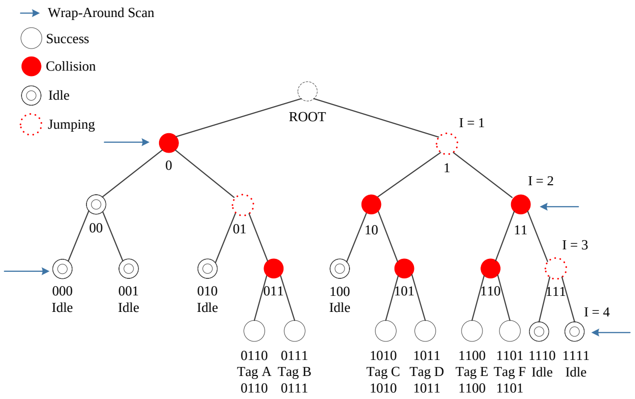


Figure 3. An example of WAS algorithm.

TABLE II. COMMUNICATION STEPS OF FIGURE 3.

Step	Broadcast	Status
1	0	Collision
2	11	Collision
3	10	Collision
4	000	No Response
5	001	No Response
6	010	No Response
7	011	Collision
8	100	No Response
9	101	Collision
10	110	Collision
11	1111	No Response
12	1110	No Response
13	1101	Identify Tag F
14	1100	Identify Tag E
15	1011	Identify Tag D
16	1010	Identify Tag C
17	111	Identify Tag B
18	110	Identify Tag A

IV. THE PROPOSED TECHNIQUE

Recall that, in WAS technique, the identification of tags in the right sub-tree could be advanced by reversing the direction of scan in each level alternatively. Figure 4 demonstrates the idea of wrap-around scan. The level ordered scans start from left to right in all "odd" (or even) levels and starts from right to left in all "even" (or odd) levels. Using wrap-around scan, the amount of prefix inquiries is expected to be decreased as the advance findings of the information of no response / collisions in the query tree could be in a balanced way. In WAS approach, jumping will reduce the number of collisions in a level. However, it will also cause unnecessary inquiry in the next level since some of inquires could be omitted once the status of the corresponding upper level is known. We describe our idea in the following sections.

In query tree algorithm, every time when collision happened, the reader will add one bit for prefix matching at next level. In such way, it may waste too much time in scanning the entire binary tree. The main idea of our proposed technique is that, instead of adding one bit for prefix matching at next level as collision occurred in query tree algorithm, we first analyze the possible ways for causing the collision and take appropriate actions at next level. We study the relationship for a collision parent node and its two child nodes. In our observation, there are four possible ways causing the collision parent node: 1) two collision child nodes, 2) one collision child node and the other identifiable (success) child node, 3) one collision child node and the other idle child node, and 4) two identifiable (success) child nodes. Figure 4 shows the four possible cases for a collision.

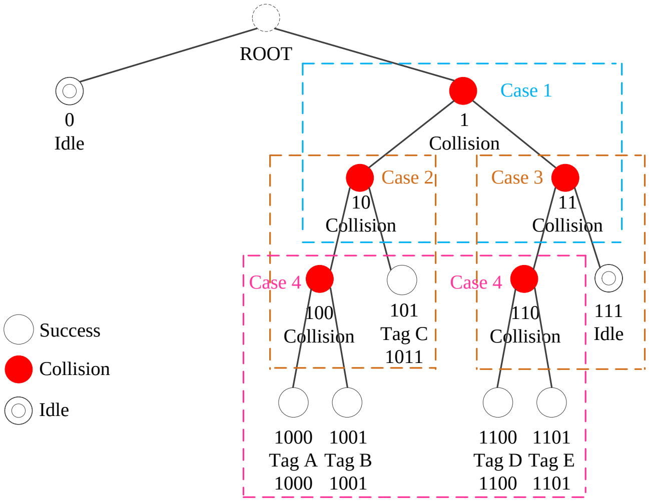


Figure 4. The four possible cases for a collision.

Based on the relationship between a collision parent node and its two child nodes, we observe that cases (3) and (4) can be used as a prior knowledge for deciding the status of a child node when the status of the sibling for the child node is known. For example, in case (3), once the result of the prefix matching is known to be idle, the prefix matching will be unnecessary for the sibling node since the status of their parent node is collided and as a result, the status of sibling node will be collided too. Another observation is that if a collision occurred at level (n-1), then, as we described in case (4), the two child nodes at level n must be identified.

Combining the WAS algorithm with the enhancement technique, we develop an enhanced WAS algorithm with prior knowledge, called the *intelligent Wrap-Around Scan* (iWAS) algorithm to reduce the query time in identification process. Figure 5 shows an example to illustrate the performance of proposed iWAS technique. Table 3 summarizes the detail steps of communication between the reader and the tags with the example of Figure 5. As shown in Table 3, 6 tags can be identified in 9 steps by using iWAS algorithm while using original WAS technique, it will take 18 steps to identify 6 tags. It should be noticed that when the reader broadcasts prefix bit string 010 to tags, there is no response from tags since no tags match 010 in their ids. At this moment, the reader is aware that nodes 000, 001 and 010 are all in idle state and node 0 is in collision state. Thus, the reader can conclude that node 011 should be in collision state. As a result, tags A and B can be identified according to case (4). Similarly, as node 100 is in idle state, the reader is aware that node 10 is in collision state. Thus, node 101 can be recognized as in collision state. As a result, tags C and D can be identified at this moment. Therefore, the proposed iWAS technique can significantly reduce the query time. Simulation results show that our proposed iWAS algorithm outperforms the original WAS algorithm in many situations. We evaluate the performance in the following section.

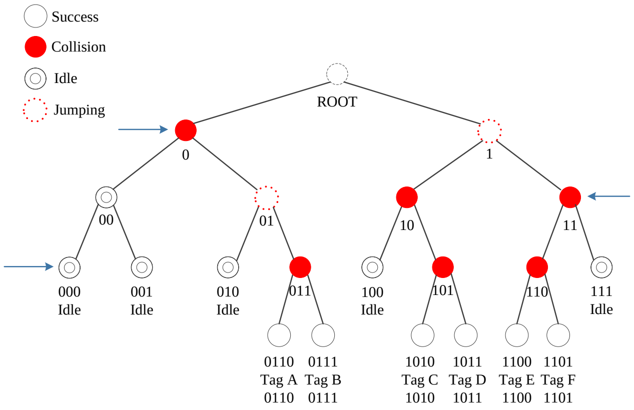


Figure 5. An example of iWAS technique.

TABLE III. COMMUNICATION STEPS OF FIGURE 5.

Step	WAS		iWAS	
	Broadcast	Status	Broadcast	Status
1	0	Collision	0	Collision
2	11	Collision	11	Collision
3	10	Collision	10	Collision
4	000	No Response	000	No Response
5	001	No Response	001	No Response
6	010	No Response	010	No Response Get info. 011 collision Identify Tags A and B
7	011	Collision	100	No Response Get info 101 Collision Identify Tags C and D
8	100	No Response	110	Collision, Identify Tags E and F
9	101	Collision	1111	No Response
10	110	Collision		
11	1111	No Response		
12	1110	No Response		
13	1101	Identify Tag F		
14	1100	Identify Tag E		
15	1011	Identify Tag D		
16	1010	Identify Tag C		
17	0111	Identify Tag B		
18	0110	Identify Tag A		

V. PERFORMANCE EVALUATION

To evaluate the performance of the proposed technique, we have implemented the iWAS scheme along with the query tree protocol (QT) and Wrap-Around Scan scheme (WAS). Figure 6 compares the number of inquires to identify different number of RFID tags. In this experiment, the tag id is set 8 bits length and density = 10% means that there are 2^8 × 10% = 26 tags, and so on. All tags are randomly generated in a uniform distribution manner. As a result, a balance tree is generated as entitled in the figure. In this experiment, the threshold value M is set 3. As shown in Figure 6, the proposed technique iWAS can reduce the amount of inquiry messages. As we expected, the iWAS outperforms both the WAS and QT techniques. When network density increasing, the proposed iWAS method presents significant improvements to the traditional query tree protocol and WAS method.

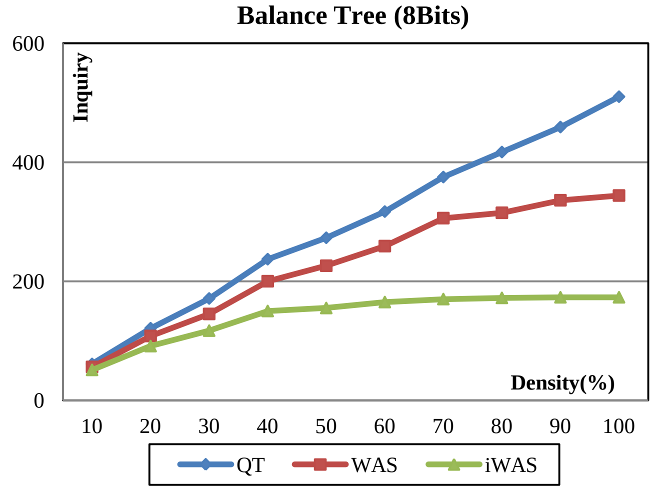


Figure 6. Performance comparison of the iWAS, WAS and QT with 8 bits RFID tags.

Figure 7 uses the same configuration as that of Figure 6 except the length of tag ID is set 16 bits. The number of tags in this experiment is set from 2^{16} × 10% = 6554 to 2^{16} × 100% = 65536. This experiment has similar observations as those of Figure 6. The proposed iWAS has better performance compare to the WAS and QT in terms of the inquiry message. It should be noticed that the amount of inquiry messages of proposed iWAS method is almost the same when the network density exceeds 70%. This is due to that when the network density is high, the balance tree is almost full. As a result, the reader can obtain the collision state in the same level during the identification process of our proposed iWAS method. Therefore, the amount of inquiry message is almost the same when network density is high.

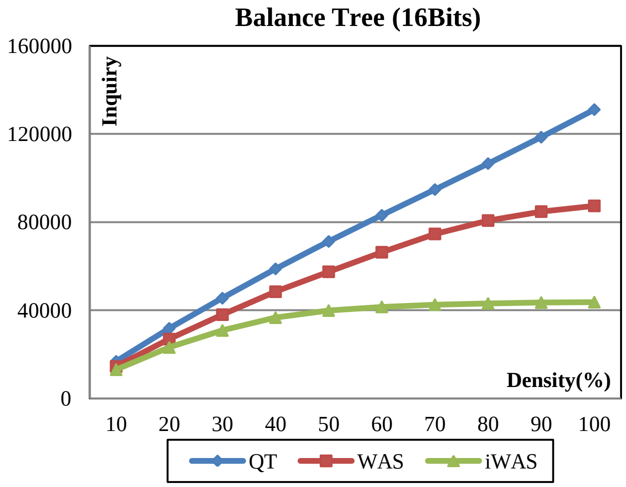


Figure 7. Performance comparison of the iWAS, WAS and QT with 16 bits RFID tags.

The last experiment was conducted with different distribution of ids among tags. The term Tree Balance Factor (TBF) defined in Figure 8 is used to indicate the percentage of tags distribution in left sub-tree and right sub-tree of a query tree. TBF = 10% means that number of tags in left sub-tree and right sub-tree are with the ration 1:9, TBF = 20% represents the number of tags in left sub-tree and right sub-tree are with the ration 2:8, and so on. As a result, an imbalance tree is generated as entitle in the figure. This experiment varies the tree balance factor from 10% which means an imbalance tree, to 50% which reflects a balance tree. The density is set 80% in the experiment. From the experimental results, the iWAS outperforms the WAS and the query tree methods in both imbalance and balance distributions.

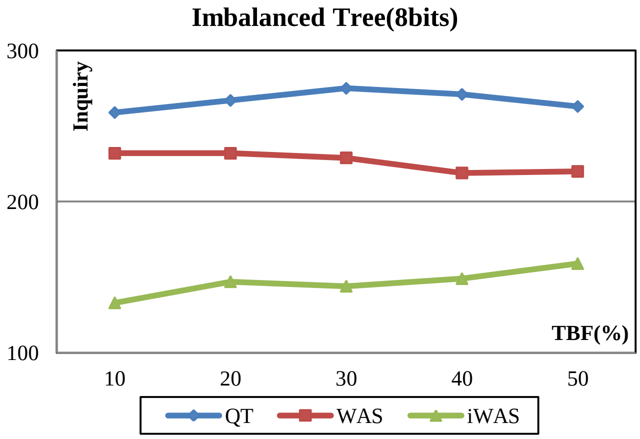


Figure 8. Performance comparison of the iWAS, WAS and QT in imbalance tree.

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-207-3

VI. Conclusion and Future Work

With the emergence of wireless RFID technologies, identifying high density RFID tags is a crucial task in developing large scale RFID systems. In this paper, we have presented an enhanced tree-based tag identification technique for minimizing tag identification cost. By using a prior knowledge, many unnecessary inquires can be reduced. Together with the Wrap-Around Scan technique (WAS), the efficiency of tag identification can be significantly improved. To evaluate the performance of proposed techniques, we have implemented the iWAS technique along with the WAS and the query tree protocol (QT). The experimental results show that the proposed technique provides considerable improvements on the latency of tag identification. It is also shown that the iWAS is effective in terms of increasing system throughput and efficiency. It remains challenging, however, to find an optimal approach that would use least prior knowledge to reduce unnecessary inquires as many as possible.

References

[1] J. I. Capetanakis, “Tree algorithms for packet broadcast channels,” *IEEE Transactions on Information Theory*, vol. 25, pp. 505–515, 1979.

[2] H. S. Choi, J. R. Cha, and J. H. Kim, “Improved Bit-by-bit Binary Tree Algorithm in Ubiquitous ID System,” in proceedings of the IEEE PCM2004, Tokyo, Japan, Nov. 29-Dec. 3, pp. 696–703, 2004.

[3] D. K. Klair, K. W. Chin, and R. Raad, “An investigation into the energy efficiency of pure and slotted aloha based RFID anticollision protocols,” in proceedings of the IEEE WoWMoM’07, June 18–21, Finland, pp. 1-4, 2007.

[4] C. Law, K. Lee, and K. Y. Siu, “Efficient Memoryless Protocol for Tag Identification,” in proceedings of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pp. 75-84, 2000.

[5] S. Lee, S. D. Joo, and C. W. Lee, “An enhanced dynamic framed slotted aloha algorithm for RFID tag identification,” in ACM Mobiquitous, pp. 166-172, 2005.

[6] J. Myung, and W. Lee, “An adaptive memoryless tag anticollision protocol for RFID networks”, IEEE INFOCOM’05, Poster Session, Mar. 2005.

[7] J. Myung, W. Lee, and J. Srivastava, “Adaptive binary splitting for efficient RFID tag anti-collision,” IEEE Communication Letter, vol. 10, no. 3, pp. 144–146, 2006.

[8] L. G. Roberts, “Aloha Packet System with and without Slots and Capture,” ACM SIGCOMM Computer Communication Review, vol. 5, pp. 28–42, 1975.

[9] J. Ryu, H. Lee, Y. Seok, T. Kwon, and Y. Choi, “A Hybrid Query Tree Protocol for Tag Collision Arbitration in RFID systems,” IEEE International Conference on Communications (ICC-07), pp. 24–28, 2007.

[10] A. Sahoo, S. Iyer, and N. Bhandari, “Improving RFID System to Read Tags Efficiently,” IIT Bombay, June: KRSIT Technical Report, 2006.

[11] B. Zhen, M. Kobayashi, and M. Shimizui, “Framed aloha for multiple RFID objects Identification,” IEICE Trans. on Comm, E88-B(3), pp. 991–999, 2005.

[12] Z. Zhou, H. Gupta, S. R. Das, and X. Zhu, “Slotted Scheduled Tag Access in Multi-Reader RFID Systems”, in proceedings of the IEEE International Conference on Networks Protocols (ICNP), pp. 61–70, 2007.

[13] F. Zhou, C. Chen, D. Jin, C. Huang, and H. Min, “Evaluating and Optimizing Power Consumption of Anti-Collision Protocols for Applications in RFID systems”, in proceedings of the International Symposium on Low Power Electronics and Design. California, USA, Newport Beach, pp. 357-362, 2004.

[14] H. Vogt, “Efficient Object Identification with Passive RFID Tags,” *Proc.* Inter. Conf. on Pervasive Computing, LNCS.2414, Springer-Verlag, pp. 98-113, 2002.

[15] C.-H. Hsu, H.-C. Chao, and J. H. Park, “Threshold jumping and wrap-around scan techniques toward efficient tag identification in high density RFID systems,” Information Systems Frontiers, Aug. 2009, doi:10.1007/s10796-009-9209-5. [retrieved: June, 2012]

Multi-channel MAC Protocol for Real-time Monitoring of Weapon Flight test in Wireless Sensor Networks

JoonKi Min
Agency for Defense Development
P.O.Box 1, Taean
Chungnam, Republic of Korea
dosolchun@add.re.kr

Jookyoung Kim, Youngmi Kwon
Dept. of Information Communications
Engineering, Chungnam National University
Yuseong-gu, Deajeon, Republic of Korea
{iliwhoth, ymkwon}@cnu.ac.kr

YongJae Lee
Agency for Defense Development
P.O.Box 1, Taean
Chungnam, Republic of Korea
yjlee@add.re.kr

Abstract— **In this paper, we propose the priority based multi-channel MAC protocol with single radio interface for the real time monitoring of the flight test of weapon systems. Concurrent transmissions with multi-channel of sensor nodes can improve the network throughput compared with single-channel transmission in wireless sensor networks. Our proposed MAC protocol has two operation modes. First is 'Normal' mode and second is 'Priority' mode. In normal mode, nodes are operated on normal CSMA/CA. And nodes have different priority depending on a sensed signal level in priority mode. High priority nodes can use more transmission channel for data send than low priority nodes. This method can guarantee successful transmission of important data generated by high priority nodes. The Class of a node is determined by own sensed data, in 'Priority' mode; nodes have three degrees – Class A, Class B, and Class C. When a sensed data of each node exceeds specific threshold value, each node has specific class respectively. High class node has low backoff exponents and can use more transmission channel. This mechanism allows that high class node has more transmission opportunity. It guarantees transmission of important data generated by high class nodes.**

Keywords- Multi-Channel; Wireless Sensor Network; MAC Protocol; Weapon Flight Test; Test Command and Control.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) [1-3] are used over a wide range such as military application, environmental monitoring, medical care, smart buildings and other industries. Energy efficiency is a main objective in most of the Medium Access Control (MAC) protocols for WSNs [4–7]. Other parameters such as bandwidth utilization, low-latency, and scalability are mostly ignored or dealt with as secondary objectives. However, bandwidth and low-latency are as important as saving energy in some applications such as military, surveillance, fire, and intrusion detection which are required a real time monitoring.

Real-time monitoring is one of the prime necessities of a flight test of weapon systems for the test command and control and the safety of person and property. A flight test of weapon systems are very dangerous and occur within short spans of time. Besides, the test is performed in a wide area and has a difficulty in a wired connection for a transmission of sensed data.

The flight test is an event-triggered application [8] in which sensor nodes do not transmit any data unless a relevant actual event (i.e. a explosion and a crash) occurs. When sensor nodes detect event, they send a sensed data to the sink at the same time. It can generate a traffic burst in the network. Because an event typically triggers many sensor nodes concurrently, the occurrence of traffic bursts produced by different nodes is highly correlated in time. Bursty or heavy wireless communication in one location (or node) may lead to contention for channel access by the nearby sensor nodes. WSNs for the flight test are required efficient and timely collection of large amounts data with high resolution. However with a single channel, WSNs cannot provide these requirements because of radio collisions and limited bandwidth.

Existing MAC protocols are not well-suited for real time monitoring of such event triggered applications with large amount of data. Characteristics of the event in a flight test of weapon systems are different from ordinary environment. We may not know exactly the event area in a wide test zone and the event occurrence time is very short. Therefore, schedule based multi-channel protocols is not appropriated to a real-time monitoring of a flight test of weapon systems. A scheduled multi-channel scheme is needed to negotiate time and procedure before data packet transmission between the sender and receiver node. A data of a event area is the interest information which is more quickly transmitted to sink than other area.

So we design the multi-channel communication protocol based on the modified Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) over a single radio for WSNs in order to improve network throughput and provide reliable and timely communication services for real time monitoring of the flight test.

Proposed MAC protocol adds a specialized priority factor under the weapon flight test environment at normal slotted CSMA/CA mechanism. This MAC protocol gives high transmission priority to the inner nodes of event area than the outer nodes. Furthermore, this MAC protocol can get not only collision reduction effect between nodes in whole sensing area but also guarantee transmission of priority nodes.

Our proposed MAC protocol has two operation modes. First is 'Normal' mode and second is 'Priority' mode. In

ISBN: 978-1-61208-207-3

normal mode, nodes are operated on normal CSMA/CA. Also nodes have different priority depending on a sensed signal level in priority mode. High priority nodes can use more transmission channel for data send than low priority nodes. This method can guarantee successful transmission of important data generated by high priority nodes. Class of a node determined by own sensed data, in 'Priority' mode and class of nodes have 3 degree – Class A, B and C. When sensed data at each node exceed specific threshold value, each node has specific class. High class node has low backoff exponents and can use more transmission channel. This mechanism allows that high class node has more transmission opportunity. It guarantees transmission of important data generated by high class nodes.

The rest of this paper is organized as follows. Section 2 reviews the related works. Section 3 presents the proposed scheme based priority in details. Section 4 presents performance evaluation through simulations. Finally, Section 5 concludes this paper with summary and directions for future work.

II. RELATED WORKS

Researchers have proposed multi-channel MAC protocols [10-15] that exploit multiple channels to increase the network throughput by eliminating the contention and interference on a single-channel in WSNs. WSNs have some limitation such as limited computation, low bandwidth, small MAC layer packet size, battery-operated power, and so on. Therefore, multi-channel MAC protocol for WSNs should consider the minimum control overhead possible in channel negotiation. Channel negotiation packets cannot be ignored as small overhead.

Multi-channel MAC protocols can be classified into three categories [9]: scheduled protocols [10-12], contention protocols [13, 14], and hybrid protocols [15].

A. Scheduled multi-channel protocols.

In this scheme, a time slot in TDMA frame for data transmission is allotted to every node which is unique in its 2-hop neighborhood. This guarantees collision-free medium access, and protocol does not waist energy and bandwidth on competition and collisions.

MC-LMAC [10] proposed a multi-channel scheme based on LMAC which allows the node to utilize new frequency channels on-demand, if the network reaches a density limit. This method is composed of two phases, one where the nodes try to select timeslots according to the single channel in LMAC rule and the second involves nodes which are unable to grab a timeslot in the first phase invite the neighbor nodes which are free to listen to them on an agreed channel or time slot.

TMCP [11] is a tree-based multi-channel protocol for data collection applications. The goal is to partition the network into multiple subtrees with minimizing the intra-tree interference. The protocol partitions the network into subtrees and assigns different channels to the nodes residing on different trees. TMCP is designed to support convergecast traffic and it is difficult to have successful broadcasts due to the partitions. Contention inside the branches is not resolved since the nodes communicate on the same channel.

In TFMAC [12], a channel scheduling mechanism is used to manage and decide when a node should switch channel to support the current communication requirements. TFMAC requires Time Synchronization and it uses single half duplex transceiver. This protocol divides each channel into time slots and the slot scheduling has been done for the medium access. The frame has been divided into contention access period where the slot scheduling and channel allocation has been done and contention free period where the data transfer has been done.

B. Contention-based multi-channel MAC protocols

Contention-based multi-channel MAC protocols use neither a predetermined transmission schedule nor the frame is divided into time slots. Instead, contention procedure is conducted at the beginning of each frame, beforehand every transmission, in order to avoid collisions. Contention-based MAC protocols allow small delay and high throughput in cases of low traffic.

MMSN [13] and TMMAC [14] have attempted to make use of multiple channels by assigning different channels to different nodes in a two-hop neighborhood to avoid potential interference. They use a different channel from its downstream and upstream nodes. Time slots are used to coordinate transmissions in these protocols. They also require precise time synchronization at nodes with frequent channel switching delays and scheduling overheads especially for high data traffic. In the multi-hop flow, nodes have to switch channels in order to receive and forward packets. This causes frequent channel switching and potential packet losses. In order to prevent packet loss these protocols use some negotiation or scheduling schemes to coordinate channel switching and transmission among nodes with different channels. They need many orthogonal channels for channel assignment in dense networks.

C. Hybrid protocols

These protocols combine principles from previous two methods. The frame is divided into time slots, but slots are assigned to receivers instead of transmitters. In the absence of traffic, hybrid protocols are more energy efficient then scheduled protocols, since each node need to be awaken to receive data only once per frame. Although hybrid protocols require contention of the potential transmitters at the beginning of each slot, contention mechanism is simpler and wastes less energy than with contention-based protocols since there is always only one receiver.

Y-MAC [15] is a hybrid multi-channel MAC protocol. to TDMA, Y-MAC divides time into frames and slots, where each frame contains a broadcast period and a unicast period. Every node must wake up at the beginning of a broadcast period and nodes contend to access the medium during this period. If there are no incoming broadcast messages, each node turns off its radio awaiting its first assigned slot in the unicast period. Each slot in the unicast period is assigned to only one node for receiving data. This receiver-driven model can be more energy-efficient under light traffic conditions, because each node samples the medium only in its own receive time slots.

III. Protocol Description

Our proposed MAC protocol is based on the Multi-Channel mechanism. Our proposed MAC protocol distributes the nodes in a network to multiple channels, so this method can reduce collisions between nodes in a network and improve network transmission efficiency [16, 17]. Our MAC protocol has two operation modes. First is ‘Normal’ mode and second is ‘Priority’ mode. Change of this operation mode is controlled by outside signal. In ‘Normal’ mode, nodes in a network operate on normal CSMA/CA what use Multi-channel manner. And, in ‘Priority’ mode, nodes in a network have priority for data transmission. High priority nodes can use more transmission channel for data send than low priority nodes. This method can guarantee successful transmission of important data generated by high priority nodes.

The Class of a node is determined by own sensed data, in ‘Priority’ mode; nodes have three degrees – Class A, Class B, and Class C. When sensed data at each node exceed specific threshold value, each node has specific class. High class node has low Backoff Exponents (BE) and can use more transmission channel. This mechanism allows that high class node has more transmission opportunity. It guarantees transmission of important data generated by high class nodes.

A. *Normal operation state*

In a general situation, proposed MAC protocol operates normal CSMA/CA. But, nodes in a network can use multiple channels, and perform not only time backoff but also channel backoff [18]. This twin backoff mechanism is more efficient to avoid collisions between nodes in a network. All nodes have same CW (Contention Window) and BE and can use three channels.

When there is a need to transmit sensed data, they first generate backoff value and select transmit channel randomly. And perform CCA (Channel Clearance Assessment) at on time and selected channel. If selected channel is idle, perform data transmit. Nor perform new time and channel backoff using new CW and increased BE value. Detail description about this procedure represented in Fig. 1.

B. *Priority State*

In Priority state, each node verifies own sensed value. If this value exceed specific threshold, that node has transmission priority. Nodes what get high priority, has more accessible data channel and low BE value. So, these high priority nodes can access media more easily than low priority nodes. Each node in a network can get one of 3 classes- Class A, B and C.

Class A node can access all channel and has minimum max BE value. These nodes can access media easily than other class nodes. Class B node can use 2 of 3 channel and Class B node only use one data channel and has maximum max BE value. Fig. 2 represents procedure of priority state. Each node decides own transmission priority when sending data generated by pre-determined threshold value. Follow own transmission priority, each node select BE and TX channel. After select BE and TX value, each node perform CCA on backoff period boundary. If channel is clear, nodes perform data sending sequence. But, if channel is buys nodes perform backoff sequence. In backoff sequence, a node only re-select TX channel.

Class A node can access not only own channel but also channel of class B and C. And, class B node use own channel and class C’s channel. All node in a network can access class C’s channel. So, in the class C’s channel occur many collision between all nodes in a network. We also propose channel selection weight factor for reduce this collision at common channel. Each node has weight factor for select own channel. Nodes in a network select own channel more frequently by effect of this weight factor.

IV. Simulation results

In this paper, we evaluate our proposed MAC protocol via simulation. We compare average media access delay and total number of successful media access. First, we compare all node in a network can access all channels with our proposed MAC protocol.

Fig. 3 shows average media access delay and total number of successful media access by channel allocation method. Uniform allocation method means all nodes in a network can access same multiple channels and classified allocation method mean some node has higher media access priority than other nodes. Like table 1, Class A node can access all data channel but class B node can access 2 of 3 data channel and class C node can access only 1 channel.

TABLE I. Number of Allocated Channel

	Number of allocated channel	
	Uniform allocation	***Priority based allocation***
Class A	3	3
Class B	3	2
Class C	3	1

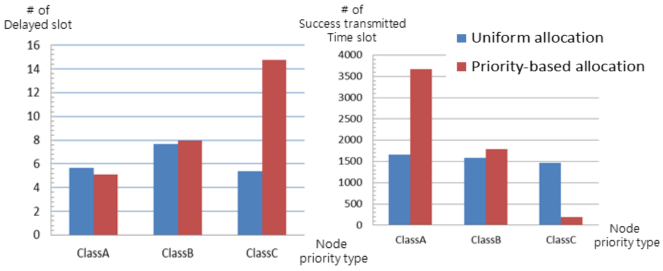


Figure 1. Average Delay and Number of Tx Success.

ISBN: 978-1-61208-207-3

In case of uniform channel allocation method, average channel access delay and total number of successful media access of each node in a network is very similar. But, in priority based allocation mechanism, number of successful media access of class A nodes what can use more channel than other low priority nodes is larger than other class nodes.

Next simulation result compare uniform channel allocation ratio with priority based channel allocation ratio. Allocation ratio represented in table 2.

TABLE II. CHANNEL ALLOCATION RATIO

	Uniform channel allocation ratio	Priority-based channel allocation ratio
Class A	1:1:1	1:1:3
Class B		1:2
Class C		1

Fig. 4 shows number of unused channel at each channel. In case of don't use weight factor for channel allocation, almost slot in #1 channel has been used but a lot of slot in #3 channel – only for class A nodes, don't used. But in case of we use weight factor for channel allocation, the number of unused slot in #3 channel is lower than don't use weight factor.

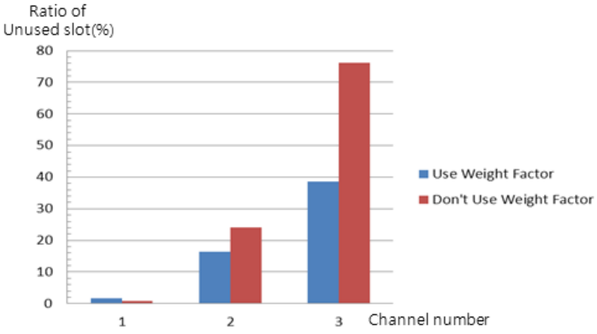
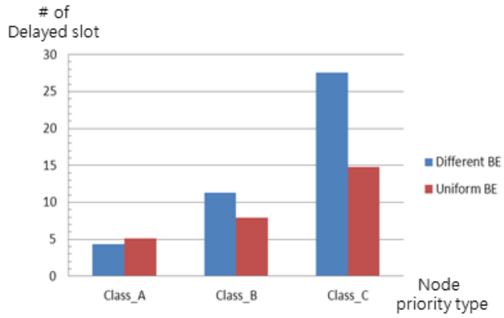


Figure 1. Ratio of Unused Slot at Each Channel.

Next Fig. 5 shows simulation result when each class has different maximum BE and minimum BE value. Table 3 shows this BE value.

TABLE III. MAXIMUM AND MINIMUN BE VALUE

	Same BE		Different BE	
	minBE	***maxBE***	***minBE***	***maxBE***
Class A	0	3	0	3
Class B	0	3	1	4
Class C	0	3	2	5



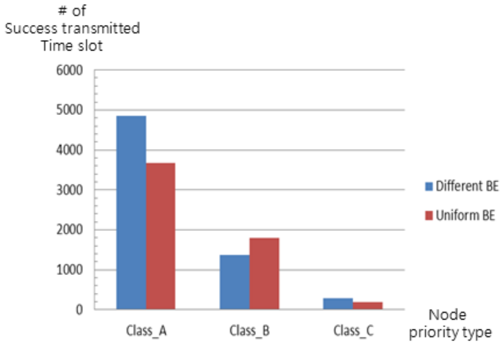


Figure 2. Average Delay and Number of Tx Success.

V. CONCLUSION

Real-time monitoring is one of the prime necessities of a flight test for the test and evaluation (T&E) of weapon systems. Data of a event area is the interested information which is reliable and quickly transmitted to sink than other area.

Existing MAC protocols in WSNs are not well-suited for real time monitoring of the flight test with large amount of data. Therefore we proposed the new multi-channel MAC protocol based on the modified CSMA/CA, which has two operation modes. In normal mode, nodes are operated on normal CSMA/CA. And nodes have different priority depending on a sensed signal level in priority mode. High priority nodes can use more transmission channel for data send and has lower backoff exponents than low priority nodes. It guarantees transmission of important data generated by high class nodes.

In the future, we plan to setup testbed sensor network system and evaluate the performance of the proposed MAC.

REFERENCES

[1] F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," Computer Networks, vol. 51, pp. 921-960, 2007.

[2] C. E-A. Campbell, I.A. Shah, K.K. Loo, "Medium access control and transport protocol for wireless sensor networks: an overview," International Journal of Applied Research on Information Technology and Computing, ISSN: 0975-8089

ISBN: 978-1-61208-207-3

(Online) and ISSN: 0975-8070 (Print), Vol. 1, No. 1, pp. 79-92, 2010.

[3] S. Misra, M. Reisslein, and G. Xue, "A survey of multimedia streaming in wireless sensor networks," IEEE Communications Surveys and Tutorials, vol. 10, pp. 18–39, 2008.

[4] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," IEEE/ACM Trans. Net., vol. 12, no. 3, pp. 493–506, June 2004.

[5] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," ACM SenSys, Los Angeles, CA, Nov, 2003.

[6] A. El-Hoiydi, and J. Decotignie, "WiseMAC: An ultra low power mac protocol for the downlink of infrastructure wireless sensor networks," Ninth IEEE Symposium on Computers and Communication, ISCC04, pp. 244–251, June 2004.

[7] M. Buettner, G. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless networks," 4th ACM Conf. on Embedded Networked Sensor Systems (SenSys), Boulder, CO, pp. 307–320, Nov. 2006.

[8] M. Ringwald, K. Romer, "BurstMAC - An efficient mac protocol for correlated traffic bursts," in Proc. of the 6th International Conference on Networked Sensing Systems 2009 (INSS 2009), pp. 1-9, Pittsburgh, Pennsylvania, USA, Jun. 2009.

[9] M. D. Jovanovic, G. L. Djordjevic, G. S. Nikolic, and B. D. Petrovic, "Multi-channel media access control for wireless sensor networks: a survey," Telecommunication in Modern Satelite Cable and Broadcasting Services(TELSIKS), 2011 10th International Conference, pp.741-744(2011)

[10] O. Incel, L. van Hoesel, P. Jansen, and P. Havinga, "MC-LMAC: A multi-channel mac protocol for wireless sensor networks," Ad Hoc Networks, pp. 73–94, 2011.

[11] Y. Wu, J.A. Stankovic, T. He, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," In Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM 2008), pp.1193-1201.

[12] M. Jovanovic, and G. Djordjevic, "TFMAC: Multi-channel mac protocol for wireless sensor networks," TELSIKS'07, Conference Proceedings, pp. 23-26, Nis, Serbia, 2007.

[13] G. Zhou, C. Huang, T. Yan, T. He, J. Stankovic and T.Abdelzaher, "MMSN: Multi-frequency media access control for wireless sensor networks," In Proc. Of IEEE Infocom, pp. 1-13, 2006.

[14] J. Zhang, G. Zhou, C. Huang, S. H. Son, and J. A. Stankovic, "TMMAC: An energy efficient multi-channel mac protocol for adhoc networks," In Proceedings of the 2007 IEEE International Conference on Communications (ICC 2007), pages 3554–3561, June 2007.

[15] Y. Kim, H. Shin, and H. Cha, "Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks," in IPSN '08: Proceedings of the 7th international conference on information processing in sensor networks. Washington, DC, USA: IEEE Computer Society, 2008, pp. 53–63.

[16] A. Adya, P.Nahl, J.Padhye, A.Wolman, and L. Zhou. "A multi-radio unification protocol for IEEE 802.11 wireless networks," In Proceedings of IEEE Broadnets'04, San Jose. CA, 2004.

[17] P. Kyasanur, and N.H. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," In Proceedings of IEE WCNC'05, New Orleans, LA 2005.

[18] R. Maheshwari, H. Gupta, and S. R. Das, "Multi-channel mac protocol for wireless networks," Sensor and AdHoc Communications and Network, 2006

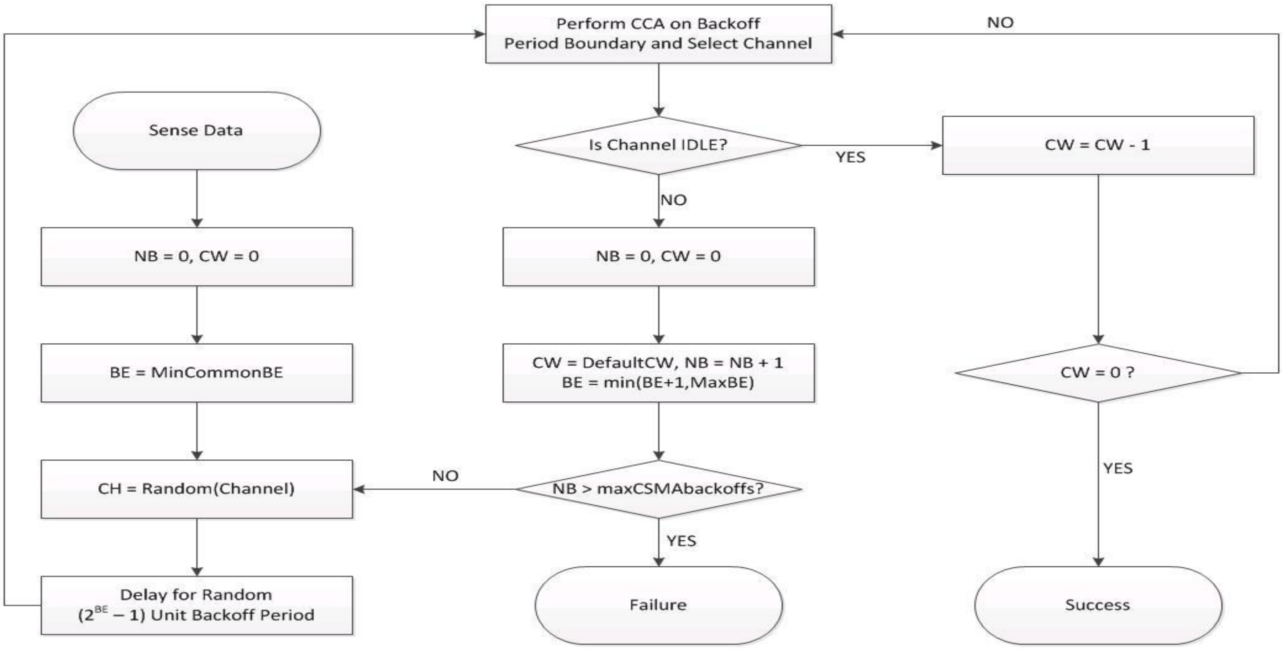


Figure 3. Normal State Procedure Flow Chart.

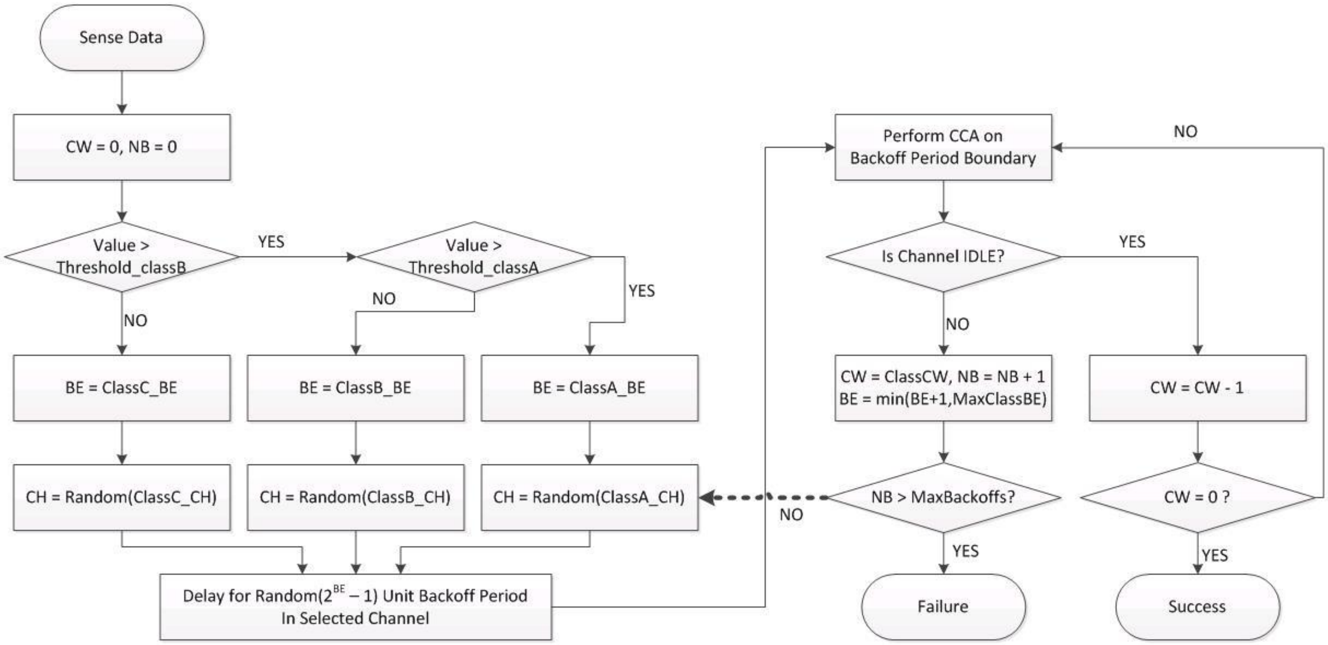


Figure 4. Priority State Procedure Flow Chart.

Cost-effective Sensor Nodes for Wireless Sensor Networks

[1, 2] Sergey Y. Yurish, [1, 3] Javier Cañete, [1] Francisco Puerta
[1]Technology Assistance BCNA 2010, S. L.
[2] International Frequency Sensor Association (IFSA),
[3] Universitat Politècnica de Catalunya (UPC, Barcelona)
Barcelona, Spain
e-mails: SYurish@sensorsportal.com, Javier.Canete@gmail.com, info@techassist2010.com

Abstract— **Cost reduction in wireless sensor networks becomes an important requirement to extend their application in fields where a great amount of sensors is needed. Traditional approach to use multichannel analog-to-digital converter and/or analog multiplexers for analog sensors will not give any reduction in price. Moreover, the analog multiplexer introduces additional measuring error. This paper describes in details the developed advanced, robust but cost-effective sensor nodes' architectures suitable for further integration in a node-on-chip. Such sensor nodes can work with any analog and quasi-digital sensors and transducers, and its sensing sub-system lets achieve the best metrological performances. A comprehensive comparative study of sensor node's architectures and sensing sub-systems are presented.**

Keywords-sensor nodes; wireless sensor networks, frequency-to-digital converter, universal sensors and transducers interface, node-on-chip

I. INTRODUCTION

Wireless sensors and sensor networks can be deployed almost anywhere at a far lower cost than can a wired system. With the recent advances in embedded systems and wireless technology, the hardware used is becoming more inexpensive and more widely available. Wireless sensor devices connect sensors wirelessly among each other as well as to monitoring and management setups. According to the MarketResearch.com the global market for wireless sensor devices used in end vertical applications totaled $ US 790 million in 2011 and expected to increase at a 43.1 % compound annual growth rate (CAGR) and reach an estimated $ US 4.7 billion by 2016 [1].

Because wireless sensor networking is built around low-power radios, the nodes that make up the network play a key role in wireless communication. From a physical perspective, the deployment of nodes may take several forms depending on the sensor application and the desired pattern of communication. Deployment may also be a one-time activity, where the installation and use of a sensor network are strictly separate activities. It can also be a continuous process where more nodes are deployed over the lifetime of the network [2]. The application can vary from a single sensor node to multiple sensor nodes.

A wireless sensor net is made up of a group of sensor nodes. Wireless sensor nodes are the essential building blocks in a wireless sensor network. Each sensor node possesses the ability to monitor some aspect of its environment, and each is able to communicate its observations through other nodes to a destination where data from the network is gathered and processed. Recent developments in wireless technologies and the semiconductor fabrication of miniature sensors are making wireless sensor networks (WSNs) smaller and more cost-effective for a growing number of uses [3].

Cost reduction in wireless sensor networks becomes a requirement to extend their application in fields where a great amount of sensors is needed [4], for example, industrial applications. In this case it should be a good solution to connect many existing low-cost sensors both: analog and quasi-digital to one sensor node to reduce the cost of nodes. Traditional approach to use multichannel analog-to-digital converter (ADC) and/or analog multiplexers for analog sensors will not give any reduction in price. Moreover, the analog multiplexer introduces additional measuring error. Hence, the analog signal must be preliminary converted to the quasi-digital signal (frequency, period, duty-cycle, time interval, phase-shift, pulse number or pulse-width modulated (PWM) output).

The described in [4] sensor interface transforms the voltage provided by various sensors with different output ranges to a pulse signal, which frequency will depend proportionally on the input voltage. The conversion of the sensor signal to a frequency value will bear much less sensitivity to interferences. The further frequency-to-digital conversion is performed by using the classical direct counting method. This technique counts the number of pulses N_x of a signal of unknown period $T_x=1/f_x$ during a gate time window T_0 determined by n periods of a signal of known, reference frequency f_0. The unknown frequency f_x is calculated by the number of pulses into the counting window:

$$N_x = n \cdot \frac{T_0}{T_x} \;\Rightarrow\; f_x = \frac{N_x}{n \cdot T_0}. \tag{1}$$

Such classical method has two well known disadvantages: the dependence of relative quantization error δ_x on frequency $f_{x,}$ and redundant conversion time determined by the constant gate time T_0. In order to achieve acceptable performance in the conversion to digital values of the frequency signal, it is necessary to the different sensor output ranges are converted into the same frequency range. For this, the conditioning electronics must be able to change sensor's gain and offset voltage depending on the sensor signal characteristics.

The circuit [4] consists of two operational amplifiers, analog multiplexer, two programmable potentiometers and voltage-controlled oscillator, which performs the transformation to the frequency range. The low resolution serial digital-to-analog converter is used for self-calibration purposes. When a node of the network is activated, the microcontroller selects the sensor to be read by means of the control lines of the analogue multiplexer and carry out the conditioning and the conversion. However, such sensor interface has some disadvantages. It introduces additional error, because of mainly based on analog electronics components.

Mantracourt Electronics Ltd. (UK) manufactures Wireless Telemetry Pulse Acquisition Module (T24-PA) for quasi-digital sensors and transducers with frequency range from 0.5 Hz to 3 kHz and relative error 0.15 ... 0.25 % [5]. It can not be used with various quasi-digital sensors and transducers, which as rule have very broad dynamic frequency range: form part of Hz to some tens MHz, and low relative error 0.01 % and better [6]. In order to get the optimal trade-off between metrological performances and price for the sensor node it is expediently to use other, universal, advanced solutions based on the novel frequency-to-digital conversion method.

In order to design cost-effective sensor nodes, which satisfy to modern requirements, the following measures must be realized. Instead of voltage or current sensors, so-called quasi-digital sensors with frequency, period, time interval, phase-shift, pulse number or PWM output must be used. The frequency-to-digital converter should be based on the advanced method for frequency measurements, which have not disadvantages, mentioned above. In case of analog output sensors, the intermediate voltage-to-frequency converter(s) should be used.

The aim of this paper is to describe in details the developed advanced but cost-effective sensor nodes' architectures suitable for further integration in a node-on-chip for wireless sensor networks. The paper consists of four parts and is organized as follows. The first part includes state-of-the-art review and task definition. The second part describes a design approach for sensor nodes architectures based on a Universal Sensors and Transducers Interface circuit (USTI) and are suitable for any quasi-digital sensors and analog sensors, and the third part is devoted to the further design of a node-on-chip based on the USTI-WSN IC. The forth part includes results of experimental investigation of sensing sub-system based on the USTI IC. The last part of the paper provides conclusions and future research directions.

II. SENSOR NODES' ARCHITECTURES

A sensor node in a wireless sensor network is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. Its architecture consists of the following components: sensing sub-system, processing sub-system, communication sub-system and power management sub-system. The sensing sub-system directly influences on metrological performances on the whole. However, during the last years a lot of publications have been devoted mainly to communication and power management sub-systems. This article will be focused on the sensing sub-system design, which satisfies to the modern requirements of relatively low cost, expansibility and power-aware [7]. The low cost means that sensor node should be cheap since wireless sensor network may have hundreds or thousands of sensor nodes. The expansibility signifies that hardware design must be expandable with a number of different quasi-digital and analog sensors to support a variety of applications. The power-aware means that hardware supports intelligent function, which allows algorithms to adapt themselves to the available power.

Let consider several advanced sensor node architectures for analog and quasi-digital sensors and transducers. All these architectures are based on novel USTI ICs developed by authors [8].

A. Sensor node architecture with analog multiplexer

A simple sensor node for analog and quasi-digital sensors with analog multiplexer and time division channeling is shown in Figure 1.

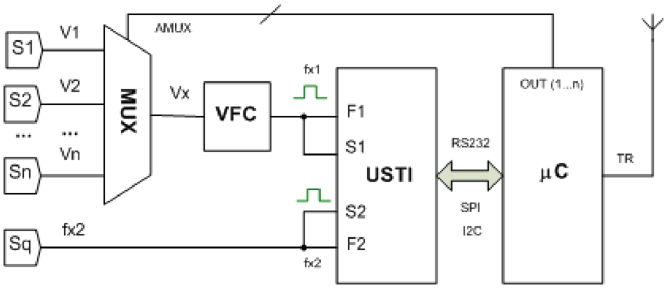


Figure 1. Sensor node with analog multiplexer.

Sensing sub-system in such architecture contains an analog multiplexes (MUX), voltage-to-frequency converter (VFC), and USTI ICs. A processing sub-system, communication sub-system and power management sub-system are realized on a separate microcontroller. The USTI is a core component of the sensing sub-system. It is based on advanced, modified patented frequency (period)-to-digital conversion method of the dependent count with a constant quantization error in all broad frequency range and non-redundant conversion time [8-10]. The dependence of relative error on conversion time is shown in Figure 2.

In this case it is not necessary to convert the different sensor output ranges into the same frequency range, as it was proposed in [4]. Only one voltage-to-frequency converter is used to convert sensor's output to frequency. As usually, integrated VFCs have broad frequency ranges and good metrological performances [11, 12]. In addition, one quasi-digital sensor can be directly connected to the second channel of USTI IC, and one sensing element (resistive,

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-207-3

capacitance or resistive bridge) can be also connected to this IC [13].

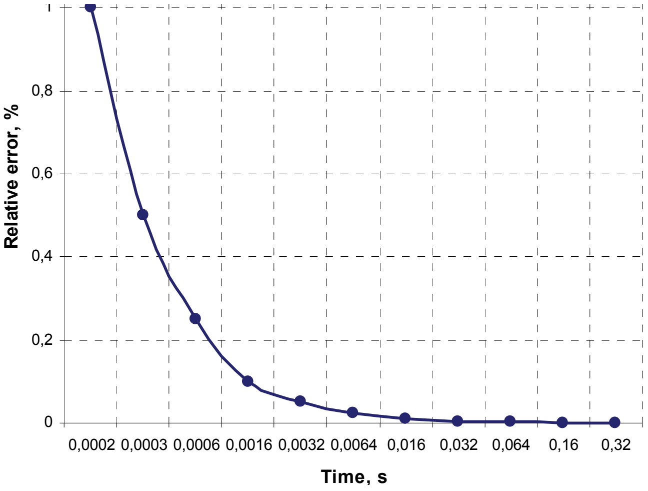


Figure 2. Relative error vs. conversion time.

B. *Sensor node architecture with digital multiplexer*

The analog multiplexer and VFC introduces additional measurement errors. To eliminate the error due to analog multiplexer, it is possible to convert voltage to frequency for each of analog output sensors before the multiplexer, and use a digital multiplexor (MX), instead of analog multiplexer. Such sensor node with the digital multiplexer and combined (time for digital signal domain and space for analog signal domain) channeling is shown in Figure 3.

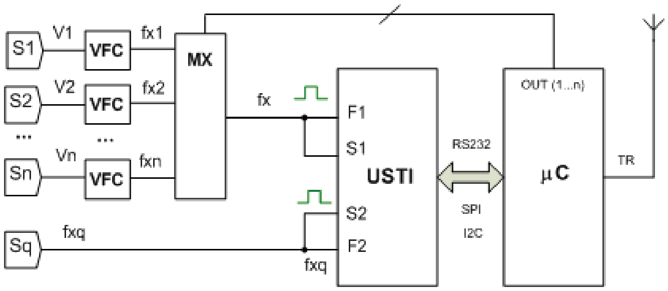


Figure 3. Sensor node with digital multiplexer.

For the time-division channeling, the cycle polling time τ can be calculated according to the following equation [14]:

$$\tau = n \cdot (T_{meas} + \tau_{delay1} + \tau_{delay2}) \,, \qquad (2)$$

where τ_{delay1} is the time delay between the ending of the conversion in the previous sensor and the command to poll the next sensor; τ_{delay2} is the time delay of the frequency conversion starting after the sensor connection; n is the number of sensors in the sensor node; T_{meas} is the measurement time. The measurement time T_{meas} for the USTI includes three main components: conversion rate (t_{conv}), communication time (t_{comm}) and calculations time (t_{calc}) [13].

The digital multiplexer does not introduce any additional error. However, the VFCs still do it. So, the solution with minimum possible hardware and high metrological performance is possible if instead of analog sensors to use quasi-digital sensors.

C. *Sensor node architecture for quasi-digital sensors*

A low-cost sensor node with high metrological performance for quasi-digital sensors and transducers is shown in Figure 4, which addresses the challenges of metrological performance improvement and node's cost reduction [4, 7, 13]. In this sensor node architecture no any VFC is necessary.

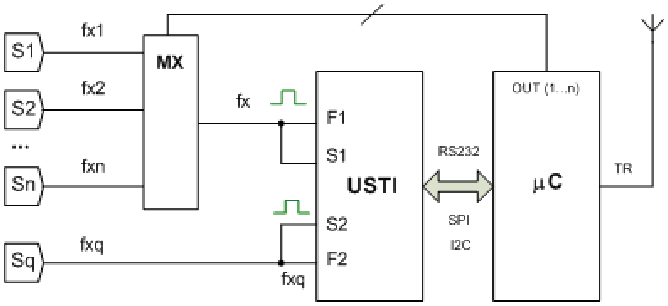


Figure 4. Low cost sensor node for qusi-digital sensors.

By this way it is possible to decrease the total measurement error, for example, from 0.14 % to 0.08 % (for the numerical example, described in [13]), and reduce the sensor node's price. For example, at the same price for analog and quasi-digital sensors, the core of sensor node - the 24-bit resolution, 8-channel ADC ADS1278 [15] costs 23.95 $ US (in quantities of 1,000) while the USTI IC with significantly better metrological performance and a digital 8-channel multiplexer costs only 18.95 $ US in the same quantities.

The space division and combining channeling also can be realized in this sensor node. In such sensor node instead of one USTI and the n-channel digital multiplexer, n ICs (according to the number of channels) and a microprocessor system with n inputs are used. That is, for simultaneous measurement of several frequencies, there is an independent channel with the USTI. The microprocessor simultaneously starts all converters and at the end of the measurement processes reads results. Quasi-digital sensors and transducers can be also connected in pairs to one USTI. In addition, one resistive, capacitive of resistive bridge sensing element can be also connected directly to the USTI. For one's turn, all USTI can be connected to a master microcontroller or microprocessor with the help of SPI or I2C buses. Each USTI IC can serves up to 3 channels by itself in a sensor node.

The cycle polling time τ for the space-division channeling is decreased approximately in k times in comparison with the time-division channeling and should be calculated as:

$$\tau = T_{meas} + t_{readout,} \quad (3)$$

where $t_{readout,}$ is the time for result reading by a microprocessor.

In the case of analog sensors, an addition VFC in each channel should be also used. Such solution lets achieve maximum possible speed at a little bit increased cost for a sensor node.

Another benefit to use quasi-digital sensors and transducers instead of analog sensors in WSN is a possibility to transmit frequency-time signals without preliminary conversion to digital. Two examples are described below.

The RF transmitter using pulse width modulation (PWM) method is reported in [16]. It does not use an analog to digital converter for the RF transmission of analog data. The transmitter consists of a pulse width modulator, a voltage-controlled oscillator (VCO) and on-chip antenna. The PWM method digitally encodes analog signal levels, but the PWM signal remains quasi-digital. By keeping the signal quasi--digital, noise effects are minimized. The modulated signals are inputted into the VCO using PWM. If the voltage of the modulated signal is the high level, the VCO is oscillated, and the RF carrier waves for transmission can be obtained. Then, the output of the VCO is transmitted by the on-chip antenna [16]. The transmitter was possible to transmit with low power dissipation of 0.75 mW the carrier of 315 MHz.

An interdigital capacitor based battery-free wireless pressure sensor is described in [17]. It consists of an interdigital capacitor (IDC) that serves as a pressure sensing element and an inductor, which works as a passive power source and data communication element. These two components work together as an LC resonator to realize the wireless pressure sensing and remote power to eliminate the need for wire connection in conventional pressure sensor. The sensing element is comprised of a set of linear parallel electrodes coated with Polyvinylidene Fluoride pressure sensing material on the top. The change of capacitance in the IDC is a function of the geometry of the electrodes and the electric properties of the sensitive layer. The sensor prototype demonstrated that it performs well in the range of 0 psi to 60 psi with an average pressure sensitivity of 25 kHz/psi [17].

III. NODE-ON-CHIP

The future step to reduce hardware expenses is to use the recently designed IC USTI-WSN of node-on-chip instead of the USTI IC and μC [13]. The USTI-WSN IC contains all sensor node's sub-systems (see Figure 5). The USTI-WSN IC prototype is shown in Figure 6.

The prototype can work with two various quasi-digital sensors or transducers and one sensing element at the same time. It has a high speed embedded transmitter with high data rate transceiver for the 2.4 GHz ISM band. The radio transceiver provides high data rates from 250 kb/s up to 2 Mb/s for wireless communications and provides frame handling, outstanding receiver sensitivity and high transmit output power enabling a very robust wireless communication. High performance RF-CMOS 2.4 GHz radio transceiver designed for industrial and consumer applications targeted for IEEE 802.15.4, ZigBee, IPv6/6LoWPAN, RF4CE, SP100, WirelessHART and ISM applications. Supply voltage is 3.6 V. Power consumption is less than 18.8 mA in active mode, and < 250 nA in sleep mode. The operation temperature range is -40 ºC to 85 ºC.

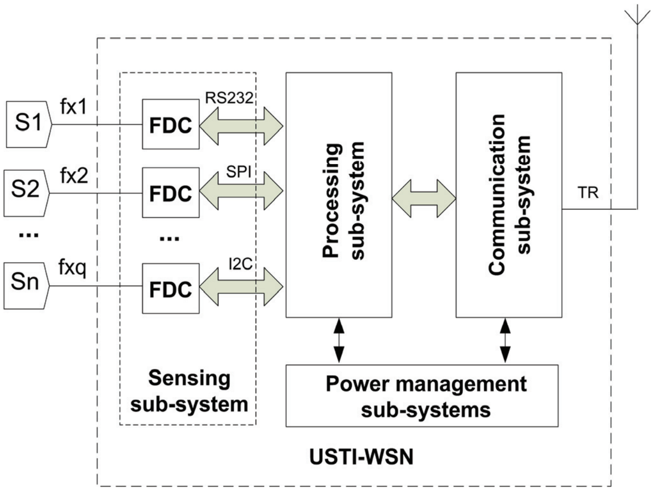


Figure 5. Block-diagram of node-on-chip.

Figure 6. USTI-WSN IC prototype in 64-pad QFN package.

IV. EXPERIMENTAL RESULTS AND FUTURE RESEARCH

The USTI IC has been tested with various quasi-digital temperature, humidity, acceleration, light, displacement, etc. sensors with frequency, period, duty-cycle and PWM outputs, and sensing elements such as resistive, capacitive and resistive bridges. The maximal possible input frequency of a square waveform pulse signal for the USTI was 9.1 MHz without prescaling, the minimal possible frequency was 0.04 Hz. The IC was programmed to measure frequency with the minimal possible relative error δ_x=±0.0005 %. Experimental results of measurements for 9 MHz and 0.05 Hz frequencies square waveform pulse signals are shown in Figure 7 (a) and 7 (b) respectively.

Before measurements, the USTI was calibrated in the working temperature range: +23.5 ºC … + 25.4 ºC with the purpose to compensate the quartz oscillator's systematic error [18]. The statistical characteristics are presented in Table I.

As it is visible from the table, the maximal relative error does not exceed the programmable δ_x < ± 0.0005 % in all frequency range including high and infralow frequencies.

For 0.05 Hz frequency the relative error does not exceed $\delta_x = \pm 0.00009\ \% < \pm 0.0005\ \%$.

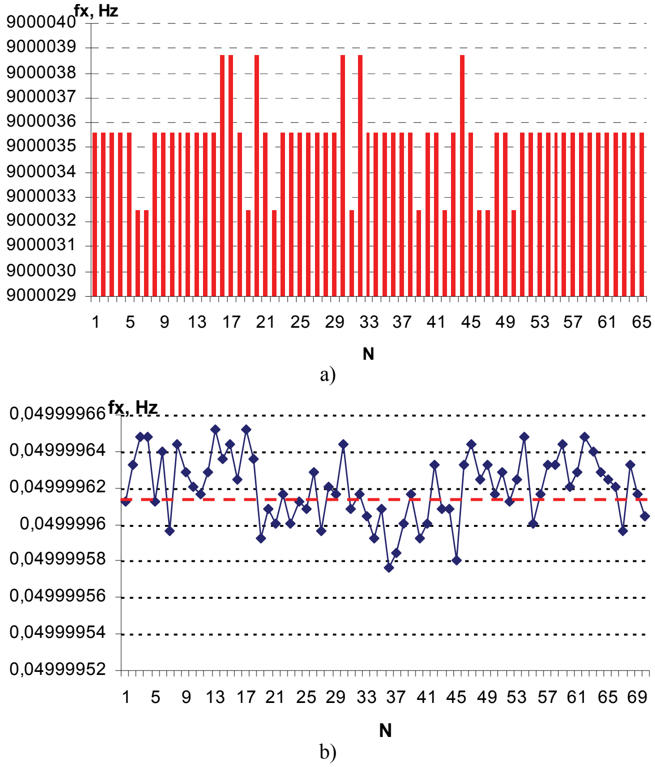


Figure 7. Experimental results for frequency measurements: 9 MHz (a), and 0.05 Hz (b).

TABLE I. STATISTICAL CHARACTERISTICS.

Parameter	Value *fx*	
	9 MHz	0.05 Hz
Number of measurements, *N*	65	70
Minimum f_x (min), Hz	9000032.48	0.049999576263
Maximum f_x (max), Hz	9000038.73	0.049999652243
Sampling Range, f_x (max)- f_x (min), Hz	6.2515	7.6E-0008
Median	0	0
Arithmetic Mean, Hz	9000035.42	0.04999962
Variance	2.405	3.4E-0016
Standard Deviation	1.5508	1.8E-0008
Coefficient of Variation	5803428.66	2709716.49
Relative error, %	0.00039 < 0.00050	0.00009 < 0.0005

The absolute and relative errors for infralow frequency measurements (f_x=0.05 Hz) are shown in Figure 8 (a) and 8 (b) respectively. The χ^2 test for goodness of fit test was applied to investigate the significance of the differences between observed data in the histograms and the theoretical frequency distribution for data from the Gaussian distribution law. The number of equidistant classes was calculated according to the following equation [19]:

$$k = 1.9 \times \mathrm{N}^{0.4}, \qquad (4)$$

where *N* is the number of measurements.

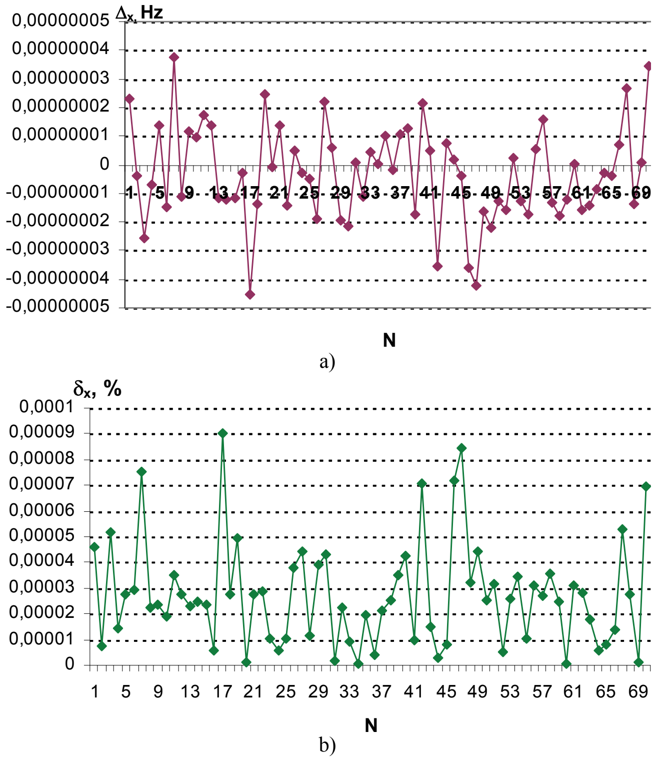


Figure 8. Absolure error (a), and relative error at 0.05 Hz frequency measurements.

At probability $P = 97$ %, and 5 equidistant classes (k=5), the hypothesis of Gaussian distribution law can be accepted for all sets of measurement data because of $S < \chi^2_{max}$, where $S = 1.7593$ is the sum of deviations between the dataset and the assumed distribution; $S < \chi^2_{max} = 7$ is the maximum possible allowable deviation in the χ^2 distribution. Hence, the hypothesis of normal (Gaussian) distribution can be accepted.

Experimental results confirmed high metrological performances and justified, that the USTI IC can be used with all quasi-digital sensors existing on the modern market [6]. The comparative analyze of proposed solution for sensor nodes and convenient solution described in [4] is shown in Table II.

The comparative metrological and technical performance of proposed solution and wireless module T24-PA available on the market [5] are shown in Table III. As it is visible from this table the wireless sensor node based on the USTI IC has significantly better metrological performance. In addition, the USTI IC has much wider functionalities and can work not only with frequency and period output sensors but also with any duty-cycle, pulse-width modulated, phase-shift, time interval, pulse number output sensors and transducers.

The further reduction of power consumption will be able due to the use of advanced method for frequency-to-digital converter with non-redundant, programmable reference frequency [20]. It will allow to change accuracy for power consumption and opposite dependent on sensor node's activity, measuring algorithm and available power.

TABLE II. COMPARISON RESULTS OF SENSOR NODE DESIGNS.

No.	Traditional Solution [4]	Proposed Solution	Benefits
1.	Analog sensors	Quasi-digital sensors	More robust and cheaper; less sensitive to interferences and noises
2.	Analog multiplexor	Digital multiplexor	No addition error
3.	Analog and mixed IC design	Digital IC design	Easy integration in standard CMOS technological processes
4.	Low metrological performances	High metrological performances	Wide applications
5.	Need a frequency range unification	Does not need any frequncy range unification	Lower hardware expenses
6.	Classical direct counting method	Modified method of the dependent count	Constant programmable quantization error; Non-redundant conversion time; Broad range of input frequencies
7.	Adaptation features: no	Adaptation features: yes	Self-adaptation; wide applications

TABLE III. COMPARATIVE PERFORMANCES.

	T24-PA	USTI IC
Relative error, %	0.15 ... 0.25	0.0005
Frequency Range, Hz	0.5 ... 3 000	0.04 ... 9 000 000
Time range, s	333E-06 ... 2	1.5E-06 ... 250
RPM range (presuming 1 pulse / rev), rpm	30 ... 180 000	3 ... unlimited
Power Supply Current, mA	35	9.5

V. CONCLUSIONS

The proposed sensor node architectures and design approach are suitable for any quasi-digital sensors and transducers. It is based on the designed USTI IC, which lets to achieve high metrological performances at relatively low cost, and get robust solution, less sensitive for various interferences, noises and distortions. Due to the USTI's broad frequency range of input signals and constant quantization error, it is not necessary to convert the different sensor output ranges into the same frequency range. The proposed sensor node architectures can also work with analog sensors. In this case the output voltage must be preliminary converted to the frequency with the help of voltage-to-frequency converter. In addition, the advanced, modified method of the dependent count for frequency measurements provides the best tradeoff between accuracy and operation time, giving a relative error less than ±0.0005 % at 0.32 s conversion time or ±1 % at 0.00016 s conversion time respectively.

The USTI IC is available on the market since 2011 from the Technology Assistance BCNA 2010 S. L., Spain. The USTI-WSN IC will be introduced on the market in 2013-2014.

ACKNOWLEDGMENT

This research and development were supported by the International Frequency Sensor Association (IFSA).

REFERENCES

[1] Market Research Projects Wireless Sensors Market Growth at 43 % CAGR Through 2016, MarketResearch.com, 29.02.12.

[2] R. Thusu, "Wireless Sensor Use is Expanding in Industrial Applications", Sensors Magazine, 1 June 2010.

[3] Comprehensive Analysis of Wireless Sensor Systems Market, Research and Markets, April 2006.

[4] A. Bayo, N. Medrano, B. Calvo and S. Celma, "A Programmable Sensor Conditioning Interface for Low-Power Applications", Proc. of the Eurosensors XXIV, 5-8 September, 2010, Linz, Austria, Procedia Engineering, vol. 5, 2010, pp. 53–56.

[5] Wireless Telemetry Pulse Acquisition Module T24-PA, Product Sheet, Mantracourt, UK, issue 1.0, 21.12. 2011.

[6] http://www.sensorsportal.com/ (last access: 11.08.12)

[7] M. A. M. Vieira, A. B. da Cunha, D. C. da Silva Jr., "Designing Wireless Sensor Nodes", SAMOS 2006, pp.99-108.

[8] Universal Sensors and Transducers Interface (USTI), Specification and Application Note, Technology Assistance BCNA 2010, S. L., 2011.

[9] Patent No. 81851 (UA), "Method of Frequency and Period Measurement of Harmonic Signal and Device for its Realization", Kirianaki N.V., Yurish S.Y., G01R 23/00, 2006.

[10] S.Y. Yurish, "Novel Modified Method of the Dependent Count for High Precision and Fast Measurements of Frequency-Time Parameters of Electric Signals", Proc. of 2008 IEEE International Instrumentation & Measurement Technology Conference - I^{2}MTC, Victoria, Vancouver Island, British Columbia, Canada, 12-15 May 2008, pp. 876-881.

[11] Manufacturers of Voltage-to-Frequency Converters at http://www.sensorsportal.com (last access 11.08.12)

[12] J.Williams, "Designs for High Performance Voltage-to-Frequency Converters", Application Note 14, Linear Technology, March 1986.

[13] S.Y. Yurish, "Digital Sensors and Sensor Systems: Practical Design", IFSA Publishing, 2011.

[14] N.V. Kirianaki, S. Y. Yurish, N. O. Shpak and V. P. Deynega, "Data Acquisition and Signal Processing for Smart Sensors", Chichester, UK, John Wiley & Sons, 2002.

[15] Presision Data Converters Guide, D011012, SLYT475, Texas Instruments, USA, 2012.

[16] W.Lee, Y.Nishida, K. Sawada and M. Ishida, "CMOS RF Transmitter using Pulse Width Modulation for Wireless Smart Sensors", submitted for IEEE Transaction for Components, Packaging and Manufacturing Technology, 2011.

[17] J. G. Villalobos, Z. Xu, and Y. Jia, "IDC Based Battery-free Wireless Pressure Sensor", Sensors & Transducers, vol. 121, issue 10, October 2010, pp. 121-132.

[18] S. Y. Yurish, "Advanced Automated Calibration Technique for Universal Sensors and Transducers Interface IC", Proc. of the IEEE International Instrumentation and Measurement Technology Conference (I^{2}MTC 2009), Singapore, May 5-7, 2009, pp.402-405.

[19] P.V. Novitskii and A.I. Zograf, "Measurement Errors Estimation", Leningrad, Energoatomizdat, 1991 (in Russian).

[20] Patent No. 79849 (UA), Method of frequency measurement and device for its realization, Yurish S.Y., G01R 23/00, 2005.

Dynamic Reconfiguration for Software and Hardware Heterogeneous Real-time WSN

Fabien Mieyeville, Mihai Galos, David Navarro
Ecole Centrale Lyon, Institute of Nanotechnology of Lyon
INL-UMR5270, CNRS, Ecole Centrale de Lyon, Ecully, F-69134, France
fabien.mieyeville@ec-lyon.fr, mihai.galos@ec-lyon.fr, david.navarro@ec-lyon.fr

***Abstract*—Wireless Sensor Network (WSN) technology has imposed itself in civilian and industrial applications as a promising technology for wireless monitoring due to its wireless connectivity, removing many hardware constraints. Initially used in low-frequency sampling applications, the increasing performances of electronic circuits has driven WSNs to integrate more powerful computation units, paving the way for a new generation of applications based on distributed computation. These new applications (process control, active control, visual surveillance, multimedia streaming) involving medium to heavy computation present real-time requirements at node level where reactivity becomes a primary concern as well as at the network level where latency must be bounded. In this paper, we present the implementation of a high-level language MinTax coupled with an in-situ compilation solution for real time Operating Systems enabling energy-aware dynamic reconfiguration while supporting hardware heterogeneity in Wireless Sensor Networks.**

***Keywords*-Wireless Sensor Network; dynamic reconfiguration; MinTax; real time; in-situ compilation.**

I. Introduction

Wireless Sensor Networks are highly distributed self-organized systems. A wireless sensor network is made of a large number of scattered tiny low-cost devices featuring strong constraints in terms of energy, processing, communications and memory capabilities. Common applications of WSN deployed on a given space are data collection from sensor node measurements that are transmitted to a specific node called the sink node. Typical deployment of wireless sensor network can be seen Figure 1.

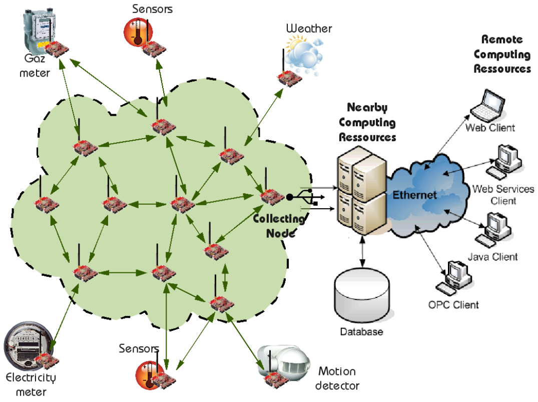


Fig. 1. Classical deployment of wireless sensor network.

First generation of Wireless Sensor Networks has been deployed in applications to remove wired connections and to offer new approaches in the physical deployment of distributed systems. Hardware node platforms of those WSNs offer limited computation ability, small memory capacity and energy constraints are so high that local demanding computations are alleviated (or sometimes optimized [1]) restraining WSN node activities to the sensing and data transmitting tasks. Yet with the advance of microelectronic technology, the new generation of hardware WSN nodes offers improved performances in power management domain as well as in computation field. Consequently, local processing on WSN node can be considered in a WSN design process flow. Thus, WSNs are now disseminating into the fields of high performance networked applications such as process control, multimedia streaming [2] and active control [3]. In this last domain, numerous successful implementations can be found, particularly in Structural Health Monitoring (SHM) [4] where realizations are numerous and distributed computation possibilities offered by WSNs are beginning to be explored [5]. These performance-critical applications require bounded delay latency and then can be referred to as real-time applications that generate new design constraints in WSN compared with conventional WSN applications.

In this paper, we will demonstrate the capability of a reconfiguration solution based on a high-level language called MinTax [6], [7] for current real-time applications. This solution takes benefit from new hardware node architecture to reduce reconfiguration consumption in WSN by in-situ compilation at node level and can be easily deployed on hardware and software heterogeneous nodes architecture in a real-time context. This paper falls in four parts: after a presentation of WSN reconfiguration state of the art, we will develop the stakes in real-time WSN (RTWSN). Then, we will develop MinTax and in-situ compilation and demonstrate by experiment that it can be used in a RTWSNs.

II. Reconfiguration: brief overview

A. A complex problem

Reprogramming a whole network can be addressed in numerous ways. The taxonomy for programming model (cf. Figure 2) established by R. Sugihara and R. Gupta [8] demonstrates the different hierarchical levels at stake in programming of WSN nodes. Our solution falls in the platform-centric category of the node-level abstraction. We will then focus on the reconfiguration of the node itself and the cost minimization of this task while complying with WSN design constraints (including real-time aspects) and not on the global process of disseminating the reconfiguration through the whole network [9] .

B. Challenges and stakes in WSN reconfiguration

The main challenges in WSN reconfiguration are [10]: (1) the energy cost of reconfiguration should be as small as possible, (2) minimization of the size of the code and small necessary memory usage to perform the reconfiguration because of the constrained hardware architectures used in WSN and last (3) the size of the code to be updated or created should be kept minimal. This last aspect has a huge impact on reconfiguration in two ways. First, minimizing the size of the code sent by RF to reconfigure node minimizes the consumption of the node. Secondly, wireless communications being unreliable due to possible signal collisions, interferences, and packet contentions, a short reconfiguration code improves the probability of being successfully received.

To those commonly recognized challenges we add the support for heterogeneity of end systems. WSNs are deployed for a long period of time: nodes can be replaced by new architectures with different hardware and software specifications compared with the initial deployment. Hence any reconfiguration solution for long-term deployed WSNs should provide support for both hardware and software heterogeneity.

Now that challenges are clearly established, we will put into context the common existing solutions for performing reconfiguration at node level.

C. Current solutions for WSN reconfiguration at node level

Three main approaches for dynamic reconfiguration are usually identified [7] as follows: use of machine code, use of bytecode for a Virtual Machine and diff-based approaches.

The first category is either associated with Operating Systems (OS-es) that can dynamically load/unload modules (modular) or those that cannot (monolithic). Monolithic OS-es are statically compiled and globally optimized into a single executable image. Part of the code cannot be reprogrammed independently and requires a complete rewriting of the entire code resident on the node. Modular OS-es offer a partial reprogramming paradigm that can take a part (module) of the functionality and link it to existing functionalities already running on the node. In general, operating systems do not offer support for hardware heterogeneous WSN and offer at best limited support for real-time [11]. The updates consist of machine code for a particular instruction set.

The second category is associated with Virtual Machines, which execute an intermediate form of information called bytecode. This bytecode is decoded and the resulting instructions are executed. If they offer a promise of hardware heterogeneity, they are either tied to a particular OS or deployed as a standalone solution. In the context of a software heterogeneous WSN, they do not provide support for different Operating Systems. Moreover, since the code is interpreted (decoded) on every run, the execution of a functionality implies overhead which means a greater energy expenditure.

The last category, the diff-based approach makes use of a difference computing algorithm that runs on the PC and generates a delta-file. This delta file contains the modifications between two versions of the software, the one already present on the node and the newer version. Then, this delta file is sent to the node and is added by a resident program to the targeted functions (improvement or creation). Functions being committed to be increased in size, a slop region between them can be provided so as to easily accommodate to those code size modifications. The diff-based approaches offer the advantage of extremely small updates, but do not offer either hardware or software heterogeneity.

D. Performances metric

Current reconfiguration solutions are based on a code processed by cross-compilation and then disseminated through the network. Hence, size of the code to be transmitted is critical since the RF transceiver is the most consuming part in WSN node. The metrics that are commonly used to evaluate performances of a reconfiguration solution are completion time, energy consumption and memory usage [10]. First, reprogramming a node is a non-trivial task which necessitates a quick completion so as to keep disruption of embedded software on the nodes to minimum, especially in case of partial reprogramming of a network. Secondly, since energy is at the heart of any WSN design process, reprogramming must ensure that the node can keep on working after the reconfiguration has been performed: this process must consume as little as possible so as not to exhaust the node. Lastly, embedded architectures in WSN nodes offer limited memory capacity: the program and data memories used in the reprogramming process must be kept to a minimum.

E. Specifications for an optimal reconfiguration solution

From our point of view, a good reconfiguration solution must provide the high level approach of a Virtual-Machine that can offer support for hardware heterogeneity and must minimize the size of the code to be sent by radio-frequency with the effectiveness of code-machine that reduces execution consumption of the code. Furthermore, the energy cost of reconfiguration must be minimal so any monolithic operating system should be avoided. Our solution that will be developed in the next sections mixes those three aspects. To those classical WSN specifications, we will add the constraints introduced by real-time applications that will be developed in the following section.

III. Reconfiguration for real-time WSN

A. Real-time WSN

Defining the real-time capacity of a wireless network requires a two-fold approach: if a quantitative notion of real-time capacity is often related to the amount of real-time data that the network is able to route in their deadlines, reactivity of the nodes themselves makes the real-time aspect fundamental in the node hardware platform itself (particularly in active control applications [5]). Most works in the field of RTWSNs focus on the quality of service with strong emphasis on the protocols [9], [12]: that is the reason why characterizing real-time ability of WSNs and evaluating performances is often linked to metrics related to network wireless communication

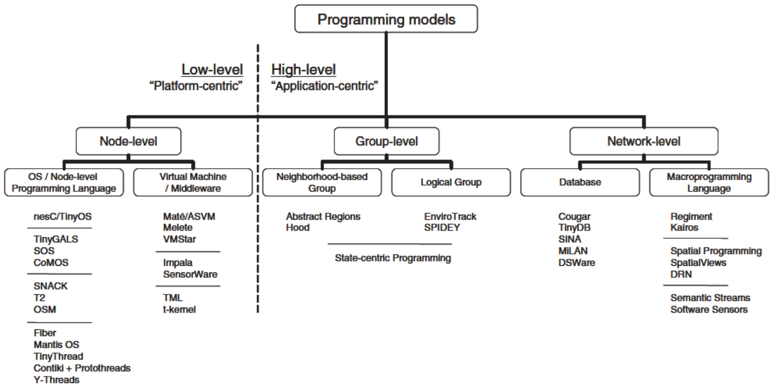


Fig. 2. A taxonomy of programming for WSN [8].

performances [13]. If such an approach is sufficient in traditional WSN applications based on low-frequency sampling rate, more demanding applications such as active control [5] or multimedia sensors [2], requiring local computation, necessitate real-time management at node level implemented in operating systems features [14], [15].

Very scarcely used in conventional WSN application, real-time is being increasingly used in new WSN applications and is very application-dependent. Then, implemented solutions will differ, both on the protocol aspect [16], [17] and at node level. Real time constraints support of software can be achieved by implementing adequate scheduling policies [12].

B. Real-time operating systems

Among existing operating systems for WSN, very few offer support for real-time [11], [18]. So as to establish the adequacy of our solution with real-time WSN, we have selected the following operating systems including energy-aware Real-Time Operating System (RTOS) kernels offering preemptive multithreading based on a traditional programming paradigm: FreeRTOS [19]. While Nano-RK (not presented in this work but currently being implemented) is the only RTOS dedicated to WSN and offering energy-awareness, FreeRTOS offers a small footprint and targets limited computation architecture: they can easily be deployed for WSN and are commonly used in WSN community [20].

FreeRTOS is a portable, open source, mini Real Time Kernel. FreeRTOS code base is small (classical kernel size from 4kBytes to 9kBytes) and is mostly written in standard C. Each task is assigned a priority and tasks with the same priority share the CPU time in a round-robin fashion. The FreeRTOS scheduler is preemptive so as to meet real-time behaviour required by the system. FreeRTOS is often used in WSN applications [20] and offers an extensive hardware heterogeneity support.

C. Reconfiguration specifications for real-time WSN

A real-time system must perform a set of actions within a certain time interval. In RTOS, tasks are executed periodically and must be completed within their deadlines. The way a RTOS manages the concurrent programming of these tasks is set by its scheduling policy that may be based on priority assignments. Reconfiguration of a real-time WSN node is critical since the new tasks to be embedded must comply by the existing deadlines. In this paper, we made the assumption that the application to be programmed is validated on a node so as to ensure the real-time integrity of the resulting programming. In particular, latencies of the system after reprogramming node must be verified on a test node before reconfiguration deployment. The resulting constraints that must be respected are the following: the RF communication must be shortened so as to ensure a viable transmission of code to be implemented and the duration of the reconfiguration of the node must be kept as small as possible.

IV. MINTAX FOR REAL-TIME WSNS

MinTax [6], [7] is a high-level programming language designed and tailored precisely for energy-aware software updates in Wireless Sensor Networks. It is compiled dynamically on the node (in-situ) after its deployment. The resulting machine code is then written to the microcontroller memory and is available as a new functionality.

Its high-level semantics mean that the code is not dependent on the underlying node hardware, as is machine code. As a consequence, when an update on a hardware heterogeneous WSN is performed, a single update for all present architectures is broadcast once. MinTax can also be considered as a generic reconfiguration method for WSN from a hardware point of view. Furthermore, the MinTax compiler residing on the node does not interact with the Operating System that may be present on the node, and the update written in MinTax does not contain any information pertaining to the OS, software heterogeneity is thus supported. What is more, because the updates need to take a short amount of time, they take the form of modules that a modular OS links to its kernel and makes them coexist with functionalities already present on the node. The MinTax compiler supports modular compilation and software heterogeneity.

A. Features of MinTax

MinTax was inspired from C, and offers a subset of the C language. Function calls, with up to 4 bytes as formal parameters, as well as returns are present. Iteration clauses such as "while" and "for", as well as branching ("if" and "switch-case") clauses are also supported. Other features include analog/digital port read/write (analog read and PWN output) and arithmetic clauses. Future versions of MinTax will probably support matrix manipulation, which would significantly improve the scope of the applications that can be deployed using it.

B. The MinTax compiler

A classical compiler is composed of an Analysis and Synthesis part, presented in Figure 3. The Analysis part is responsible for reading the input file, divide it into atoms (or indivisible parts) and constructing a parse tree. This tree contains information on the order the instructions are to be interpreted. The splitting into atoms is called lexical analysis and the construction of the parse tree is done by the syntactical and semantical analysis. Next, the Synthesis part is the part that does the actual machine code generation. In the case of the MinTax compiler, the lexical analyser is generated using a program called Re2C [21] (regular expression to C). Lexical rules are given in the form of regular expressions, and a C file is generated with the language's corresponding lexical analyser. In what concerns the Semantical and Syntactical part, the parser is generated with a parser generator called Lemon [22]. Lemon takes semantic rules in the form of an input file in which the grammar of the MinTax language has been implemented. During the parsing process, a Symbol Table with the all the symbols present in the MinTax file (functions, variables) is created. It will be used to generate the code in the generation phase.

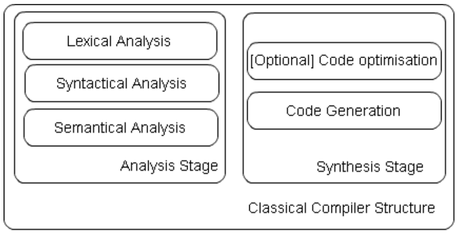


Fig. 3. Classical compiler structure

For the Synthesis part, the instruction set for every supported architecture (AVR and MSP430) has been implemented as function primitives. These primitives take formal parameters as arguments and generate the instructions they model.

The process described above is described in Figure 4.

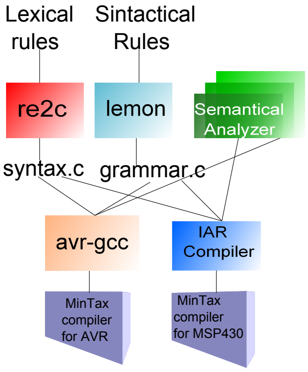


Fig. 4. The process required to compile the MinTax compiler

For a more detailed presentation of MinTax features, the reader may refer to previous works [6], [7].

C. Supported hardware architectures

One of the key feature of MinTax is its support of heterogeneity: the major microcontroller families of WSNs are then supported: the AVR ATMEGA128 family and MSP430 and MSP430x architectures. A cross-compilation feature is also provided so as to enable development and debugging on computer before deployment. Some platforms supported by the MinTax compiler are presented in Table I.

TABLE I
MINTAX COMPILER SUPPORT FOR COMMERCIAL WSN PLATFORMS

WSN Node	Flash usage	RAM usage	RF Transceiver
Mica2	26kB	150bytes	CC1000, 433Mhz
AVRRaven	26kB	150bytes	AT86RF230, 2.4Ghz
Zolertia Z1	22kB	168bytes	CC2420, 2.4Ghz

V. EXPERIMENTAL VALIDATION

We have validated our approach on a point-to-point communication between two different hardware platforms implementing two different real-time OSes (funkOS and FreeRTOS).

A. Hardware and software configuration

A Zolertia Z1 node under funkOS [7] sends a functionality written in MinTax to an AVR Raven node from ATMEL running the FreeRTOS kernel. This node receives the functionality, compiles the function and links it to its existing functions. In the same time it resends the received data to another node so as to emulate a dissemination. Measurements are processed through the use of a home-made current-sense amplification circuit, which enables us to correlate measurements with the different phases of microcontroller activity. This RTOS offering no support for RF communication, we have implemented an IEEE 802.15.4 RF packet format since it is the common protocol used in real-time WSN [9].

B. Code example

The code implemented is a loop corresponding to the following pseudo-code: *count to led until 11, send RF, delay*. This example uses a great number of functionalities of MinTax (calculation, jump, use of hardware feature of microcontroller, etc.) and illustrates the robustness of this high-level language solution. The MinTax translation of this pseudo-code is the following (code is commented as pseudo-code to help the understanding of MinTax syntax):

```
 inputMinTax[]={
"bCa{" // function b, char parameter a
"Ea<11;"  //if a>11
"s@a;"  // call send a
"#" //end if
"};" //end function b

"aa{" //function aa
"Wa<11;" //while a<11
"a+;" // increment a
"#" //end while
"};" //end function aa


"a{" //function a
"WT" //while true
"b=$a;"  //read port a
"b@a;"  // send
"b+;" // increment
"$a=b;" // while porta
"aa@;"  // delay
"#" //end while true
"};" //end function a


};
```

The function realized by this code has a compiled sized (machine code obtained by cross-compilation) of 96 bytes. The MinTax formulation of this code occupies 60 bytes. On this quite simple solution, the MinTax abstraction enables a gain of 33% on the size, which should result in reduced RF transceiver consumption.

TABLE II
DETAILED OF DURATION AND ENERGY CONSUMPTION OF REPROGRAMMING PROCESS.

State	Duration	Power	(mJ)	Total (mJ)
RX	40,8	127,15	4,986	17,547
Compile	111,28	83,33	9,273	
Flash	23,3	80,66	1,879	
TX	9,28	130,01	1,207	

C. Experimental results

Figure 5 shows the activity on the microcontroller of the AVR Raven node and durations and energy consumption of each state are summarized in the Table II. Beyond the success of reprogrammation of the function that is seamless merged with other existing functionalities, a thorough analysis of these results enables us to evaluate the performances of our solution. Four successive phases are to be observed:

1) the receiving of the data whose shortness enables to solicit at minimal the consuming RF transceiver,
2) the code is compiled by the in-situ compiler and linked to the FreeRTOS kernel,
3) the generated machine code is written in the flash memory,
4) broadcast of the MinTax code to the other nodes so as to reconfigure the whole network. This step is accomplished at the end of the process so as to validate the integrity of the received file. Hence, no corrupted data, and then no useless configuration file, is emitted through the network at the cost of a slight latency in the global reconfiguration process. This implementation offers a supplementary robustness to the reconfiguration process.

At the end of the RF-broadcasting, the experimental traces show the execution of the new functionality.

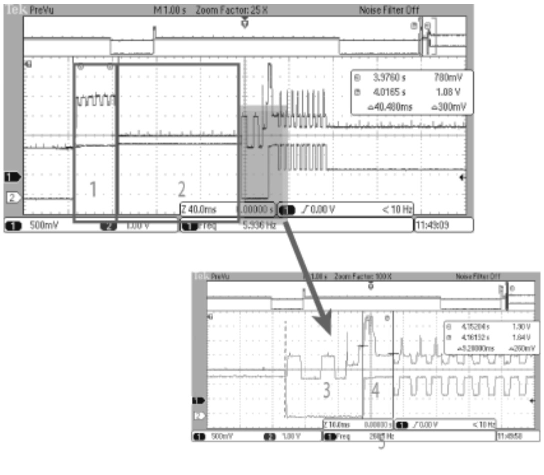


Fig. 5. Physical measurement of the reconfiguration process.

D. MinTax and in-situ compiler performances analysis

MinTax used with in-situ compilation enables the fast reconfiguration of node in approximately 184.66 ms with less than 18 mJ. The transmission duration and energy budget is from far inferior to the receive phase: indeed, if we use an IEEE 802.15.4 protocol for initial sending of data from Zolertia Z1 to AVR Raven node, the resending of data is accomplished without any protocol layer.This choice was made to evaluate the cost of the protocol overhead on our solution. As a result, the strong overhead necessary for an IEEE 802.15.4 protocol-based communication stands for 75% of the energy necessary to the RF transmission of the reconfiguration data. This cost has obviously a strong impact on the reconfiguration process. Hence, if MinTax enables a gain of 33% on the size of the transmitted code, it represents only a gain of 5ms on the 40.8ms of the global RF communication. MinTax should then be used with a lightweight RF protocol such as the RF layer used for Contiki [23]. Moreover, it is important to notice that the presented example is quite simple and the advantages of our solution grow with the size of the application code to reconfigure. If for small applications, compilation could remain consuming, for big applications, compilation energy will be smaller compared with transceiver energy expenditure since offering a power consumption reduced by one third.

Otherwise, the global performances of MinTax present an improvement for reconfiguration compared with the literature. For example, the reconfiguration of a blink application using run-time linking of ELF files in the Contiki operating system [24] takes 972ms for a total consumption of 19.92mJ.

E. Compression implementation for improved performances

To improve latency performances of MinTax in a real-time applications context, we have implemented the support for compression. After the generation of the MinTax code, we proceed to a compression using a Huffman-based compression algorithm.

This approach has been used with the example previously described. From a size of code of 60 bytes for MinTax, we obtain a size of 49 bytes, i.e., a gain of 12%. From the previous conclusion, it is obvious that this gain will not have a huge impact on the reconfiguration cost in a classical WSN configuration using a IEEE 802.15.4 algorithm. We, nevertheless, performed the experimentation so as to evaluate the cost of the decompression on the reconfiguration process. The experiment results are to be seen on Figure 6.

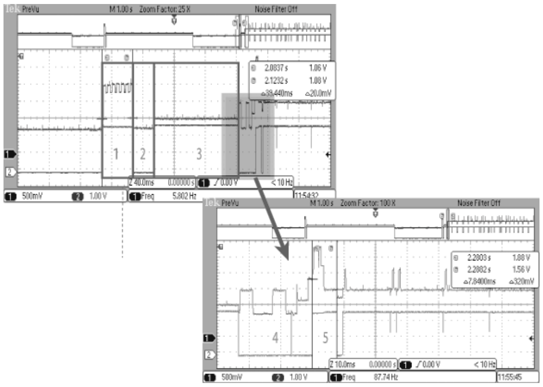


Fig. 6. Physical measurement of the reconfiguration process using compression.

The decompression phase is added to the previous behaviour, resulting in five successive phases to be observed: (1)

TABLE III
DETAILED OF DURATION AND ENERGY CONSUMPTION OF REPROGRAMMING PROCESS USING COMPRESSION.

State	Duration (ms)	Power (mW)	Energy (mJ)	Total (mJ)
RX	39,44	126,43	4,986	18,129
Decompress	25,84	52,46	1,356	
Compile	108,24	82,03	8,879	
Flash	23,44	78,91	1,850	
TX	7,84	134,96	1,058	

receiving of the compressed MinTax code, (2) decompression of the archive by a resident algorithm on the node, (3) compilation of the code, (4) writing of the generated machine code in the flash memory and (5) broadcast of the MinTax code to the other nodes so as to reconfigure the whole network.

Durations and energy consumption of each state are summarized in the Table III.

From a global point of view, the overall performance of the reconfiguration is very similar to the precedent section where no compression was applied. The decompression cost is not really high in term of energy consumption (only 7.4% of the global energy consumption) but is more impacting on the latency with 25.84ms, the same duration of the flash writing, that is to say about 12.6% of the total duration of the reconfiguration (204.76 ms).

The loss of performances compared with the case where no compression is implemented is about 15% as well on energy consumption as on duration of reconfiguration. Yet the slightly reduced duration time for receiving and transmitting of the reconfiguration code could justify the use of compression. Furthermore, the application case is here quite simple. If we consider more complex applications such as distributed computation on node requiring reconfiguration, the over-cost of compression should be absorbed by the gain in code size and in duration of dissemination.

VI. CONCLUSION AND PERSPECTIVE

We have here successfully demonstrated the use of MinTax for real-time operating systems with hardware heterogeneous support. Furthermore, the energy cost of reconfiguration is kept minimal and the duration of the reconfiguration is small compared with the current reprogramming solutions, making it particularly suitable for real-time systems. The latencies introduced by reconfiguration (key parameter in real-time systems) are currently being explored so as to establish the upper-bounds: more complex functions such as distributed computation algorithms being used in active control applications are under test since MinTax support high complexity [6] code involving loop, conditional evaluations, nesting, etc.. This work will validate the quality of services offered by MinTax as well as the validity of compression approach that will take benefit from the increased size code. MinTax is also currently being developed for Nano-RK [14] a popular reservation-based real-time operating. The support of this RTOS should offer to MinTax and its compiling solution an extended coverage of RTOS for WSN.

REFERENCES

[1] L. Gu *et al.*, “Lightweight detection and classification for wireless sensor networks in realistic environments,” in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, ser. SenSys ’05. New York, NY, USA: ACM, 2005, pp. 205–217.

[2] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “A survey on wireless multimedia sensor networks,” *Computer Networks*, vol. 51, pp. 921–960, 2007.

[3] I. Akyildiz and M. Vuran, *Wireless sensor networks*. John Wiley & Sons Inc, 2010.

[4] J. Lynch and K. Loh, “A summary review of wireless sensors and sensor networks for structural health monitoring,” *Shock and Vibration Digest*, vol. 38, no. 2, pp. 91–130, 2006.

[5] Y. Wang *et al.*, “Decentralized civil structural control using real-time wireless sensing and embedded computing,” *Smart Structures and Systems*, vol. 3, no. 3, pp. 321–340, 2007.

[6] M. Galos *et al.*, “Energy-aware software updates in heterogeneous wireless sensor networks,” in *9th IEEE International NEWCAS Conference*, June 2011.

[7] M. Galos *et al.*, “Reprogramming hardware-software heterogeneous wireless sensor networks,” in *The 14th International Symposium on Wireless Personal Multimedia Communications (WPMC’11)*, Brest, France, Oct. 2011.

[8] R. Sugihara and R. K. Gupta, “Programming models for sensor networks: A survey,” *ACM Trans. Sen. Netw.*, vol. 4, pp. 8:1–8:29, April 2008.

[9] X. Feng *et al.*, “A survey of adaptive and real-time protocols based on IEEE 802.15. 4,” *International Journal of Distributed Sensor Networks*, vol. 2011, 2011.

[10] Q. Wang, Y. Zhu, and L. Cheng, “Reprogramming wireless sensor networks: challenges and approaches,” *IEEE Network*, vol. 20, no. 3, pp. 48–55, 2006.

[11] M. O. Farooq and T. Kunz, “Operating Systems for Wireless Sensor Networks: A Survey,” *Sensors*, vol. 11, no. 6, pp. 5900–5930, May 2011.

[12] O. Chipara, C. Lu, and G. Roman, “Real-time query scheduling for wireless sensor networks,” in *Real-Time Systems Symposium, 2007. RTSS 2007. 28th IEEE International*. Ieee, 2007, pp. 389–399.

[13] P. Pagano *et al.*, “Simulating real-time aspects of wireless sensor networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, p. 2, 2010.

[14] A. Eswaran, A. Rowe, and R. Rajkumar, “Nano-rk: An energy-aware resource-centric rtos for sensor networks,” *26th IEEE International RealTime Systems Symposium RTSS05*, vol. 0, pp. 256–265, 2005.

[15] X. L. K. Z. R. C. W. Dong, C. Chen and J. Bu, “Fit: A flexible, lightweight, and real-time scheduling system for wireless sensor platforms,” *IEEE Trans. Parallel Distributed Syst. (TPDS)*, vol. 21, no. 1, pp. 126–138, 2010.

[16] S. Bansal, D. Juneja, and S. Mukherjee, “An analysis of real time routing protocols for wireless sensor networks,” *International Journal of Engineering Science*, vol. 3, 2011.

[17] Z. Teng and K. Kim, “A survey on real-time mac protocols in wireless sensor networks,” *Communications and Network*, vol. 2, no. 2, pp. 104–112, 2010.

[18] A. M. V. Reddy *et al.*, “Wireless sensor network operating systems; a survey,” *Int. J. Sen. Netw.*, vol. 5, pp. 236–255, August 2009.

[19] R. Barry, *FreeRTOS Reference Manual — API Functions and. Configuration Options*. Bristol, UK: Real Time Engineers Ltd., 2010.

[20] A. Schoofs *et al.*, “A framework for time-controlled and portable wsn applications,” *Sensor Applications, Experimentation, and Logistics*, pp. 126–144, 2010.

[21] P. Bumbulis and D. D. Cowan, “Re2c: a more versatile scanner generator,” *ACM Lett. Program. Lang. Syst.*, vol. 2, no. 1-4, pp. 70–84, 1993.

[22] “The Lemon Parser-Generator,” in *http://www.hwaci.com/sw/lemon/ (accessed August 11, 2012).*.

[23] A. Dunkels, B. Gronvall, and T. Voigt, “Contiki - a lightweight and flexible operating system for tiny networked sensors,” in *LCN ’04: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2004, pp. 455–462.

[24] A. Dunkels *et al.*, “Run-time dynamic linking for reprogramming wireless sensor networks,” in *Proceedings of the 4th international conference on Embedded networked sensor systems*, ser. SenSys ’06. New York, NY, USA: ACM, 2006, pp. 15–28.

Towards the Design of a Component-based Context-Aware Framework for Wireless Sensor Networks

Manik Gupta
School of Electronic Engineering and Computer
Queen Mary University of London
London, UK
e-mail: manik.gupta@eecs.qmul.ac.uk

Eliane Bodanese
School of Electronic Engineering and Computer
Queen Mary University of London
London, UK
e-mail: eliane.bodanese@eecs.qmul.ac.uk

***Abstract*—Context-awareness is one of the prerequisites in order to design adaptable applications and services. As Wireless Sensor Networks get more pervasive in nature and cater to diverse application they need to incorporate context-awareness. A lightweight sensor node level context processing framework is desirable. A component software programming-based approach has been proposed to design the context-aware architecture. A reference implementation has been provided and potential advantages in terms of low software reconfiguration overhead have been highlighted.**

Keywords-context-awareness; component-based software; middleware;adaptive sampling.

I. INTRODUCTION

Context-awareness is important for Wireless Sensor Networks (WSNs) since there are several WSN applications that need to make decisions on the basis of the prevailing contextual environment. For the current work, air pollution monitoring for urban streets application is being worked upon and the sensor nodes need to provide fine grained resolution pollution measurements for variables like carbon monoxide, etc. The sensor nodes can also provide support for measuring variables like temperature, air pressure, humidity, wind speed, location etc. These additional variables can be used as contexts for adapting the sensor application. An adaptive sampling application for changing the sampling rate of the sensor nodes is an example of such a context-aware application; wherein the sampling rate needs to be adapted as and when sensor nodes undergo certain environmental changes using an intelligent algorithm.

Hence, as the WSNs get more pervasive in nature and need to incorporate more intelligence in order to facilitate node level decision making, a generic lightweight context-aware framework design and implementation for the wireless sensor nodes becomes an imperative need to design smarter and adaptable applications. One of the challenges to be addressed is to make the framework customizable, allowing runtime reprogramming and dynamic reconfiguration, in order to address the application and environmental diversity, to which the WSNs generally cater.

Recently, component-based software development [1][2][3] has been advocated for WSN programming. Software componentization provides a high level programming abstraction through interface-based interactions between modules. A component represents a single unit of functionality and deployment. Components can be compiled separately and then composed into a system. A component can communicate with its outside through a well-defined interface and receptacle. An interface defines a set of operations provided by a component to others, while a receptacle specifies the set of operations a component requires from others. The system can be reconfigured by switching from an old component to a new one implementing the same interfaces.

Dynamically deployable and reconfigurable software components offer a lot of advantages for WSN since they are typically large in scale and operate for long periods of time in the face of dynamic environmental conditions and changing application requirements. Software development becomes easier since during the development cycle, changes to only one part of the system (single algorithm or function) are done at a time, so the rest of the software remains unaffected. Software updates become very convenient since only the updated components need to be transmitted through the network. It is possible to implement dynamic and reconfigurable applications since the functionality can be altered by reprogramming only parts of the system.

Hence, the main contribution of the paper is to propose a component-based context-aware architecture for sensor nodes. This framework bridges the gap between component-based software and context-awareness for WSNs by enabling a lightweight solution for context management. Though the use of component-based software development in the WSN field is not new at all, but the majority of the sensor nodes today are not capable of managing the contexts (other than merely acting as a context collector). This sensor node-based framework will aid not only in collection, but also in processing and use of the contexts for decision making. In a traditional context-aware application where all the contexts are collected and processed in a single monolithic application, a change or modification related to one of the contexts will lead to the complete application

ISBN: 978-1-61208-207-3

change. On the contrary, in case of the component-based context-aware application design, only the context which needs to be changed can be updated by means of incorporating changes to that particular component. The specific requirement to make sensor nodes smarter and more practical by means of managing multiple contexts in an online manner is the main driver behind the proposed architecture.

Rest of the paper is structured as follows. Section II gives details about various context-aware frameworks and component-based middleware and software systems available for WSNs. Section III gives details about architectural design of the component-based context-aware framework. Section IV gives the reference implementation details and Section V draws a conclusion on the paper.

II. RELATED WORK

Context-awareness has been acknowledged as a very important step in ubiquitous computing and there are a lot of context-aware systems which exist in the literature [4][5][6]. Most of these context-aware systems typically cater to the needs of pervasive computing, but are not suitable for sensor node level context processing in WSN. Most of these systems are heavyweight and need back end processing. Given the nature of WSNs, a context-aware framework for WSNs cannot be heavyweight and processing/memory intensive. Therefore, new designs are required which are lightweight and are suitable for the needs of WSN applications.

Hence an investigation into component-based software paradigm for wireless sensor networks was carried out. Both component-based middleware solutions for WSN as well as component-based software reconfiguration and operating system exist for WSNs in the literature. RUNES [7], GridKit [8] and LooCI [9] are examples of component-based middleware's for WSN, but they do not address context-awareness specifically. RUNES and GridKit are more suitable for network level reconfiguration, while Java-based implementation of LooCI limits its usage. On the contrary, in this research work, the aim is to develop a context-aware framework built using a lightweight component middleware which can work on resource constrained nodes. Wisekit [10] is the only node level component-based middleware solution for WSNs found in literature that addresses context-awareness by providing application adaptation. It is a distributed component-based middleware solution which addresses context-awareness by making the adaptation and reconfiguration of WSN applications possible, but actual implementation details are not mentioned in the paper. Figaro [11] for fine grained software reconfiguration and Lorien [12], dynamic component-based operating system are examples of use of component software paradigm in WSNs, but again, they do not address context-awareness specifically.

Based on the literature survey on both the context-aware framework and component-based software for WSNs, it was found from the survey that a gap exists in these two domains in terms of integration by means of developing a component-based context-aware framework. Hence, in the current work, it has been proposed to adopt the component-based software programming paradigm for the development of a lightweight and reconfigurable context-aware framework. The architecture design has been explained in the following section.

III. DESIGN OF THE COMPONENT-BASED CONTEXT-AWARE FRAMEWORK

In this section, the design architecture for a component-based context-aware framework has been proposed. The context-aware framework needs to be built using the services of a component-based middleware. A component-based middleware called MIREA [13], developed at UCL, and has been used in this research work to build the context-aware framework. MIREA is specifically targeted at real-time embedded systems. MIREA is light-weight, component-based, and supports flexible reusability of software components. The aim of building the context-aware framework on top of the component-based middleware in this research work has been to lend adaptability at the middleware level that in turn will lead to adaptation at the application level.

A component can specify functionality that it requires and/or provides using a well-defined interface. The model as shown in Fig. 1 consists of *ComponentTypes*, *Components*, *Interfaces*, *Receptacles*, and *Connectors*. A *Component* is a runtime instance of a ComponentType. A *ComponentType* can export one or more *Interfaces*, through which a given component provides a set of functionalities to other components (i.e., in the form of a set of C/C++ functions in the middleware). A *Component* can have any number of *Receptacles*, through which a set of required functionalities are specified. A component can also have an associated component wide state that is only accessible from within the containing component. *Connectors* are a specialized form of component that performs intermediary actions if required, for instance, in order to monitor, log or encrypt data for security reasons.

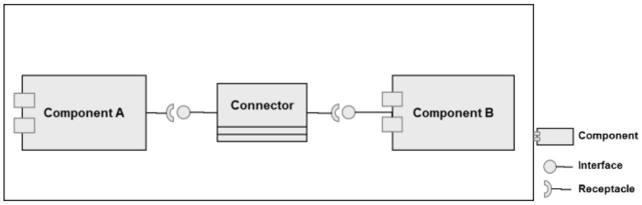


Figure 1. Elements of a component-based system.

MIREA provides the following categories of core services:

1. Loading and unloading of *ComponentTypes*
2. Instantiation and destruction of *Components*

3. Registration and acquisition of *Interfaces* and *Receptacles*
4. Connection between *Interfaces* and *Receptacles*
5. Registration and acquisition of Components' *States*
6. Destruction of *Connectors*

All of the services above are runtime activities whereas the process of defining a new *ComponentType*, *Interface*, *Receptacle*, *Connector* and *State* are static in nature - defined at an application design stage. Connections between components can be reset and reconfigured by first destroying an existing connector instance and then reconnecting them to a new type of interface and receptacle. After this, any invocations made on the given *Receptacle* will be redirected to the newly connected *Interface* instance, hence a new/different *Component* instance. More details about MIREA can be found in [13].

Most of the context-aware systems have a generic architecture consisting of context collector, context reasoner and a context database. In the component-based architecture, each of these tasks is going to be performed by an individual standalone component that will have a well-defined interface for interaction with the other components. These reusable components can be loaded and unloaded during runtime. Also, once loaded, the components can be instantiated at run time and their respective interfaces and receptacles can be registered and connected to each other using the MIREA API's. The architecture of the component-based context-aware framework has been shown in the Fig. 2.

The various components and their respective functionality in this architecture are explained as follows:

- Sensor Manager will be responsible for interacting with the physical sensors and will hide the hardware specific details from the application. The context collector will invoke the sensor manager for data from a particular sensor depending on a particular application and the sensor manager will provide the data to the context collector.
- Context Collector will be the main component responsible for collecting the application specific contexts and use interfaces provided by other components to gather data from sensors and map it into contextual information, store data in buffers and provide contextual data to the context reasoner.
- Context Database provides storage service for the various contexts and depending on the application requirement, the data can either be offloaded to the base station or be stored in external storage available on the sensor node.
- Context Reasoner is the component responsible for carrying out the reasoning over the various collected contexts and make adaptive decisions according to the application requirements. This component will also be responsible for performing data processing tasks like prediction or clustering analysis of the gathered contexts in order to facilitate the decision making process.

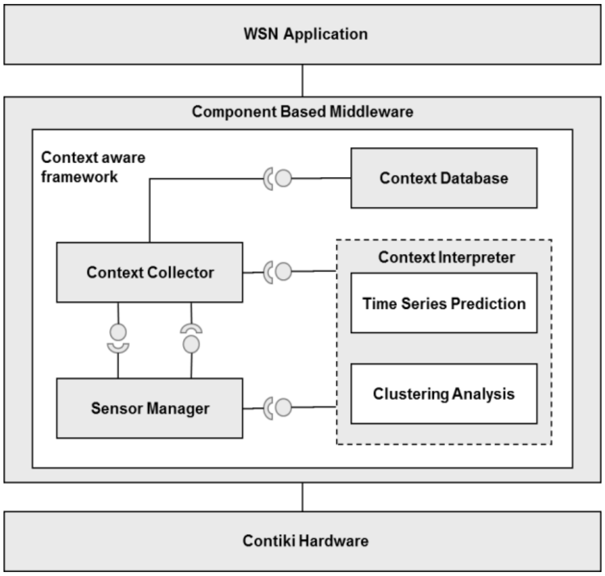


Figure 2. Architecture of component-based context-aware framework.

IV. Reference Implementation

The reference implementation for an adaptive sampling based data collection application is shown in the Fig. 3, which defines the various interfaces/receptacles between the various components. The adaptive sampling technique used in the reference implementation has been proposed as a part of on-going research work and more details can be found in [14]. The adaptive sampling technique based upon time series forecasting can adapt the sampling rate of a sensor node according to the prevailing contextual environment.

The implementation has been done using the Contiki [15] operating system on the T-mote sky platform. Contiki was chosen as the operating system of choice because of its support for dynamic loading and linking of loadable software modules [16]. Each loadable module in Contiki is in Compact Executable and Linkable format (CELF) format which is a modification of the common object code format , Executable and Linkable format (ELF). The dynamic loader/linker (elfloader) in Contiki parses the ELF format and is able to perform dynamic loading, linking and relocation of ELF object code files. Initially the components can be stored in the external EEPROM by programming either using the serial interface or over the air programming. The various steps involved in implementing the context-aware framework on the T-mote Sky are as follows:

- The MIREA middleware was ported onto Contiki.
- The components were implemented and built using the platform specific compiler.

- The components are loaded on the external flash of the T-mote sky using the serial interface.
- The MIREA application driver program loads, connects and unloads the components using the MIREA API's.

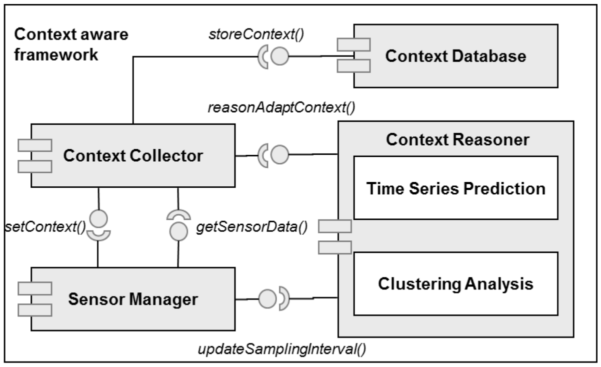


Figure 3. Interface/receptacle definition for an adaptive sampling application.

The memory footprints for each of the components (w.r.t. both program and data memory) for T-mote sky are shown in Table I. The memory footprints have been evaluated by looking at the size of binary images after compilation. The *text* section represents the code size, *data* section is the size of initialized memory, and *bss* section is the size of uninitialized memory.

TABLE I. MEMORY FOOTPRINTS FOR VARIOUS COMPONENTS

Component Name	**Memory Footprints**		
	Text(bytes)	***Data(bytes)***	***BSS(bytes)***
sensorManager	554	10	12
contextCollector	632	10	0
contextDatabase	256	10	0
contextReasoner	984	22	14

Another experiment was carried out to compare the Contiki image size with context-aware framework implementation as a single software module (monolithic context-aware framework) vs. Contiki image with MIREA implementation and the results are shown in Table II In case of monolithic implementation, once the context-aware framework is programmed on the node, it is difficult to reconfigure and update the software program. Any software reconfiguration would require communicating full application image of ~23K bytes to the node. On the contrary, in case of the component-based implementation, once the node is programmed with MIREA middleware that occupies ~33K bytes, less than ~1K bytes (refer to Table I) need to be transmitted to the node to enable component runtime reconfiguration or reprogramming.

TABLE II. COMPARISON OF MEMORY FOOTPRINTS FOR DIFFERENT IMPLEMENTATIONS

Different Implementations	**Memory Footprints**		
	Text(bytes)	***Data(bytes)***	***BSS(bytes)***
Contiki image with monolithic context-aware framework	23054	178	5164
Contiki image with MIREA middleware	32936	170	7658

V. CONCLUSIONS

In this paper, a design of a component-based context-aware framework has been proposed for WSN. This framework provides integration between the component software and context-awareness technologies for the WSNs. This design architecture has several advantages in terms of ease of programming, software updates and reconfiguration, dynamic application development etc. This architecture is lightweight in nature and suitable for sensor node level context-aware processing. Most of the context-aware systems existing in the literature do not cater to the needs of sensor nodes and are architecturally and functionally more sophisticated and heavyweight in nature. The component-based software technology is suitable for programming context-aware WSN applications and a proof of concept reference implementation using the Contiki OS that enables dynamic linking/loading of software components has been carried out.

ACKNOWLEDGMENT

The authors would specially like to thank our partners, Prof. Steve Hailes and Dr. Jagun Kwon from University College London for providing their full support and help on the MIREA middleware. This project was funded by India-UK Advanced Technology Centre (IUATC).

REFERENCES

[1] Karl H. Johnasson, John Lygeros, Anthony Tzes, Karl-Erik Arzen, Antonio Bicchi, Gianluca Dini, Stephen Hailes, A component-based approach to the design of networked control systems. *Proceedings of the 33rd Midwest Symposium on Circuits and Systems*.

[2] Barry Porter, Utz Roedig, Francois Taiani, Geoff Coulson, A comparison of static and dynamic component models for Wireless Sensor Networks. *Computing, 224460*.

[3] Geoff Coulson, Gordon Blair, Paul Grace, Francois Taiani, Ackbar Joolia, Kevin Lee, Jo Ueyama, Thirunavukkarasu Sivaharan, A generic component model for building systems software. *ACM Trans. Comput. Syst.* 26, 1, Article 1 (March 2008), 42 pages.

[4] Daniel Salber, Anind K. Dey, Gregory D. Abowd, The context toolkit: aiding the development of context-enabled applications. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit* (CHI '99). ACM, New York, NY, USA, 434-441.

[5] Manuel Román, Christopher Hess, Renato Cerqueira, Anand Ranganathan, Roy H. Campbell, Klara Nahrstedt, A Middleware Infrastructure for Active Spaces. *IEEE Pervasive Computing* 1, 4 (October 2002), 74-83.

[6] Tao Gu, Hung Keng Pung, Da Qing Zhang, A service-oriented middleware for building context-aware services. *J. Netw. Comput. Appl.* 28, 1 (January 2005), 1-18.

[7] Paolo Costa, Geoff Coulson, Cecilia Mascolo, Gian Pietro Picco, Stefano Zachariadis, The RUNES middleware: a reconfigurable component-based approach to networked embedded systems, In *Proceedings of the 16th Annual IEEE International Symposium Personal, Indoor and Mobile Radio Communications, 2005.*

[8] Paul Grace, Geoff Coulson, Gordon Blair, Barry Porter, Danny Hughes, Dynamic reconfiguration in sensor middleware. In *Proceedings of the international workshop on Middleware for sensor networks* (MidSens '06). ACM, New York, NY, USA, 1-6.

[9] Danny Hughes, Klaas Thoelen, Wouter Horr, Nelson Matthys, Javier Del Cid, Sam Michiels, Christophe Huygens, Wouter Joosen, LooCI: a loosely-coupled component infrastructure for networked embedded systems. In *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia* (MoMM '09). ACM, New York, NY, USA, 195-203.

[10] Amirhosein Taherkordi, Quan Le-Trung, Romain Rouvoy, Frank Eliassen, WiSeKit: A Distributed Middleware to Support Application-Level Adaptation in Sensor Networks. In *Proceedings of the 9th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems* (DAIS '09), Twittie Senivongse and Rui Oliveira (Eds.). Springer-Verlag, Berlin, Heidelberg, 44-58.

[11] Luca Mottola, Gian Pietro Picco, and Adil Amjad Sheikh. 2008. FiGaRo: fine-grained software reconfiguration for wireless sensor networks. In *Proceedings of the 5th European conference on Wireless sensor networks* (EWSN'08), Roberto Verdone (Ed.). Springer-Verlag, Berlin, Heidelberg, 286-304.

[12] Barry Porter, Geoff Coulson, Lorien: a pure dynamic component-based operating system for wireless sensor networks. In *Proceedings of the 4th International Workshop on Middleware Tools, Services and Run-Time Support for Sensor Networks* (MidSens '09). ACM, New York, NY, USA, 7-12.

[13] Jagun Kwon, Stephen Hailes, MIREA: Component-based middleware for reconfigurable, embedded control applications, In *Proceedings of the IEEE International Symposium on Intelligent Control* (ISIC,2010),

[14] Manik Gupta, Lamling Venus Shum, Eliane Bodanese, Stephen Hailes, Design and evaluation of an adaptive sampling strategy for a wireless air pollution sensor network, In *Proceedings of the IEEE 36th Conference on Local Computer Networks*, (LCN, 2011).

[15] Adam Dunkels, Bjorn Gronvall, Thiemo Voigt, Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks* (LCN '04). IEEE Computer Society, Washington, DC, USA, 455-462.

[16] Adam Dunkels, Niclas Finne, Joakim Eriksson, and Thiemo Voigt. 2006. Run-time dynamic linking for reprogramming wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems* (SenSys '06). ACM, New York, NY, USA, 15-28.

Applications Development on a Rule-Based WSN Middleware

Jiaxin Xie, Zixi Yu, Xiang Fei, Partheepan Kandaswamy
Department of Computing,
Coventry University
Coventry, UK
{xiej, yuzi, x.fei, kandaswp}@coventry.ac.uk

***Abstract*— Wireless Sensor Network (WSN) middleware eases the WSN application development by providing an application programming interface (API). Rule-based WSN middleware enables the applications and users to program the behavior of the sensor nodes. REED (Rule Execution and Event Distribution) is such a middleware solution that allows sensor networks to be programmed at run time. In this paper, we propose a method of developing WSN applications that uses finite state machine (FSM) as a bridge between application logics and the rules running on the REED, and demonstrate that for applications, if their behaviors can be described using finite-state machine (FSM), they can be directly described using the rules and thus implemented on the REED; further, we argue that rule-based middleware is useful for implementing bio-inspired mechanisms, such as self-organization, on WSN systems. Two WSN applications are implemented, as examples, on the REED: one is the de-synchronization of sensor nodes, and the other is the clustering-based self-organization. This paper is not aimed to study a specific application or control mechanism on WSNs, but to, via two prototype implementations, show that rule-based middleware such as the REED is useful and flexible enough to support the development of WSN applications, especially for bio-inspired mechanisms.**

Keywords-wireless sensor network; rules; finite state machine

I. INTRODUCTION

The advance in Wireless Sensor Networks (WSN) technology has led to a variety of WSN applications. One example is PROSEN [1] (PROactive SENsing) research project that aimed at developing a WSN system for proactive wind farm condition monitoring. The features of WSN systems, such as the distribution and heterogeneity of sensor nodes, the constrained processing power, memory, and energy for each sensor node; and the error-prone wireless links over which sensor nodes communicate, make the WSN application development a challenging task [2]-[4]. To ease the wireless sensor data collection, delivery and query, WSN middleware is introduced that provides an application programming interface (API) to shield the application (developer) from the complexities arising from the WSN. Rule-based WSN middleware enables the applications and users to programme the behaviour of the sensor nodes. Conceptually, a *rule* takes the form of <*event*, *condition*, *action*> where:

- an *event* is received from any other component in the system. This is can be an event carrying data values, or other events such as a timeout event, a sleep or wake-up event, and so on.
- a *condition* is a Boolean expression that will be evaluated when the *event* occurs.
- an *action* is executed if the above *condition* is true when the *event* is received. The action may manipulate or store data. It may also generate another *event* to other components in the system, such as an *event* to trigger other *rules*.

Fig. 1 shows the general architecture of the rule-based middleware [7]. The *Rule-Base* stores all the rules derived from the application while the *Fact-Base* stores the states of the node and the events that have occurred. The *Event-Manager* is responsible for receiving *events* and passing them to the *Rule-Engine*. The *Rule-Engine*, based on the current event and the states stored in the *Fact-Base*, executes matching rules stored in the *Rule-Base*. The results of the execution could be the update of the *Fact-Base*, or sending an event to other components in the system via the *Event Manager*.

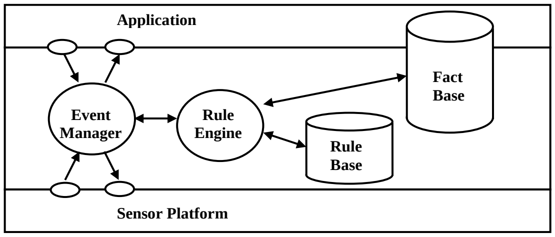


Figure 1. Architecture of rule-based middleware

A Rule Execution and Event Distribution (REED) middleware, originally for the PROSEN project, has been designed and implemented. REED is based on the general architecture described in Fig. 1; but further, enables programmability at run time, i.e., the system behaviour can be programmed by applications at run time, so as to be adaptive to the changing application goals and changing environment [5][6].

Provided the rule-based middleware such as the REED, for application developers, their main focus should be on constructing the rules that describe the logics of the application tasks, and this leads to the question of how to effectively and efficiently express the application specific domain behaviour using rules. In this paper, we propose a method of developing WSN applications that uses finite state machine (FSM) as a bridge between application logics and the rules running on the REED. Further, we argue that rule-based middleware such as the REED is especially useful for implementing bio-inspired mechanisms for WSNs as bio-inspired mechanisms imply that only simple rules are needed to be running on each node in order to achieve the emergent behaviour (such as self-organization) of the whole WSN system.

The rest of the paper is organized as follows: Related work is discussed in Section II. Then, the REED for the PROSEN project is briefly described in Section III. In Section IV, a general method of constructing WSN rules is proposed, followed by explaining the advantages of rule-based middleware for bio-inspired WSN mechanisms. The implementation of two bio-inspired mechanisms on the REED, i.e. sensor nodes clustering and de-synchronization, are described in Section V and tested and evaluated in Section VI. Section VII concludes the paper.

II. RELATED WORK

Fei and Magil [5] and Fei and Yu [6] have listed the related work on the rule-based middleware for WSNs, including a general architecture of the rule-based middleware proposed by Terfloth, Wittenburg, and J.Schiller [7]. Fei and Magil [5] also developed a rule-based middleware, called REED, for the PROSEN project. In addition, a survey across a broad array of WSNs and middleware including rule-based middleware has been provided by Mottola and Picco [2]. A systematic study on the same topic has been given in Terfloth [10]. As mentioned in Section I, for application developers, directly constructing rules for the WSN systems is still non-trivial. Using FSM, a primitive and useful graphic model for describing system behaviours, as a bridge between application logics and the rules running on the WSN systems will make it more straightforward for the application developers to construct rules running on the rule-based WSN middleware such as the REED. To the knowledge of the authors, this paper is the first to build up the relationship between FSMs and the rules running on the WSN systems.

Some bio-inspired mechanisms on WSNs have been proposed. The biologically-inspired clustering algorithm proposed by Wokoma, Shum, Sacks and Marshall [11] was inspired from quorum sensing, a biological process that is carried out within communities of bacterial cells. Based on how fireflies and neurons spontaneously synchronize, Geoffrey, Geetika, Ankit, Matt and Radhika [12] developed a fully distributed time synchronization mechanism among TinyOS-based motes; likewise, Julius, Ian, Ankit and Radhika [13] developed a fully distributed time de-synchronization mechanism. Boonma and Suzuki [14] developed the BiSNET, a biologically-inspired sensor networking architecture, to address issues such as scalability, energy efficiency, self-healing, etc. However, they were not implemented on rule-based middleware. We argue and demonstrate in this paper that rule-based middleware, plus the FSM-based rules construction method, will ease the applications development on WSN systems.

Jacobsen, Zhang and Toftegaard [15] provided an overview of bio-inspired principles and methods applicable to sensor network design. This overview also mentions that by using simple rules for the behaviour and the interaction among individuals a global optimum can be achieved on a large, system-level scale. However, no real case of rule-based bio-inspired mechanism is provided. In this paper, two bio-inspired mechanisms will be implemented on a rule-based middleware to demonstrate the effectiveness of the rule-based bio-inspired mechanisms development on WSNs.

III. REED FOR PROSEN

A. *PROSEN Architecture*

Fig. 2 illustrates the system architecture for PROSEN where REED lies in. The system consists of a Policy Server (PS), a PN (Processing Node) for each wind-turbine, and sensors to measure parameters such as temperature, wind-speed, wind-direction, and gearbox temperature. The PS interacts with users and operators to obtain the goals for the system. Such goals might describe a desirable power output or response to poor weather conditions. The PS converts the goals to a set of policies. These policies in turn are converted to low-level rules. These rules are then distributed to the REEDS on the PNs.

For the Rule-Engine, in addition to executing the *rules* in response to received *events* as described in Section I, in order to support on-line programmability, its functionality also includes:

- Managing a rule database to allow the adding, removing, and overriding of rules
- Verifying rule consistency
- Merging and filtering rules.

For detailed information on the REED and its language description, refer to [5].

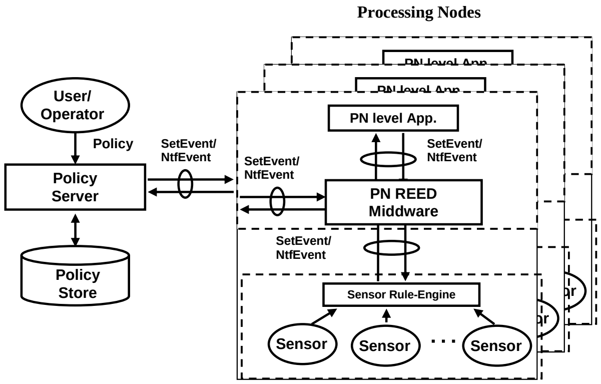


Figure 2. PROSEN system architecture

B. *General Prototype Structure*

The general prototype system structure, as shown in Fig. 3, consists of one PC functioning as a PS, one PC as a Gateway [6] and PNs that are the combination of PCs (as PN emulators) and a GumstixTM [8]. GS400K-XM is a miniature full function Linux motherboard based on low power Intel XScale® technology. It has 16MB flash memory which can accommodate JamVM [9], which is a compact JVM (Java Virtual Machine). REED is developed using Java and running on PNs (PCs and the Gumstix). The PS and the gateway are connected via the Internet, and the Gateway and the PN are connected via a 174MHz wireless link. The sensor reading is simulated via random number generator.

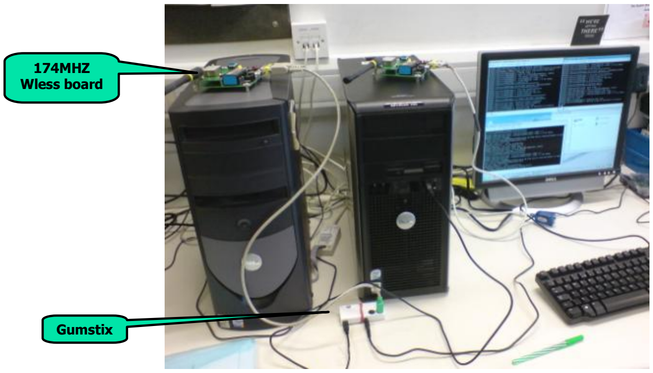


Figure 3. General prototype system structure

IV. APPLICATIONS DEVELOPMENT ON THE REED

A. *A General Method of Constructing WSN Rules*

Rule-based middleware, such as the REED, to some extent ease the development of WSN applications However, as each rule is encoded by a structure of textual data format; application developers may still face difficulties in constructing the rules that describe the behaviour of the application tasks. For them, graphical models, such as FSMs and UML statechart diagrams, are the convenient ways of expressing the behavior of the application. We argue that the FMS as shown in Fig. 4 has the direct relationship with the rules supported by the REED, and the mapping from the FMS to the rules are as follows:

- The Event part in the FSM can be mapped to the event of a rule;
- The State part in the FSM can be mapped to the condition of a rule;
- The Action part in the FSM can be mapped to the action of a rule plus an extra action that is to update the current state;
- The number of the $\frac{\text{Event}}{\text{Action}}$ in the FSM is equal to the number of the *rules* for an application.

So, as long as the application behaviour can be described using finite-state machine (FSM), the construction of rules from FSM is straightforward. Two cases in Section V provide examples of how to construct rules via FSMs.

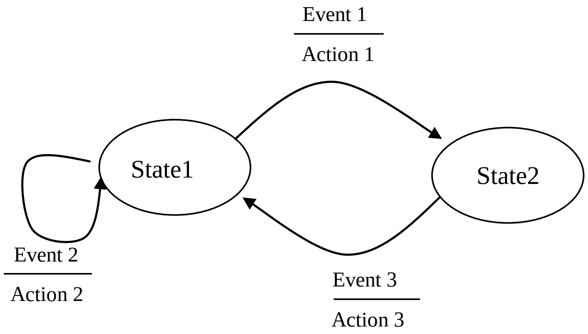


Figure 4. Finite state machine

B. *Support for Bio-inspired WSN Mechanisms*

Biological systems achieve complex goals reliably via the collaboration of a large number of cheap, unreliable components. Such collaboration is based on each component executing simple tasks in respond to the stimuli. For WSN, sensor nodes share the similar features with the components in biological systems: distributed, resource limited, unreliable, and etc.

Bio-inspired mechanisms, such as self-organization, have been applied to WSN. If the simple tasks executed on each node in response to the stimuli can be expresses as a small set of *rules*, as mentioned above, rule-based middleware such as the REED will be advantageous in facilitating the implementation of the bio-inspired mechanisms on WSNs. Both two cases in Section V are bio-inspired.

V. TWO CASE STUDIES

A. *Rule-Based Sensor Nodes Clustering*

Clustering sensor nodes is one of the self-organisation mechanisms applied to WSNs. There exist quite a few clustering mechanisms for WSNs [11]. The clustering algorithm implemented in this paper features first the clusters are formed dynamically and updated periodically; second, the process overhead imposed on the cluster heads is balanced among all the sensor nodes.

To describe the behaviour of the clustering algorithm, its FSM is drawn first as in Fig. 5. As ‘UpdateCurrentState’ is the default action, due to limitations of space, it is not listed in the *action* part of the *rules*. According to the Section IV, the rules can be directly derived from the FSM as shown in Table 1. Further, it is found that if the *action* set for an *event* is always the same across the whole states, the *condition* part of the rules for that event can be simply replaced by “TURE”; this is especially the case if for an events, it occurs only when the PN is in a specific state. This replacement not only makes the rule set concise, but also reduce the processing time for the PNs to check the *condition* part of the rules. So for R1, R2, and R3, their *condition* part is simply “TRUE”.

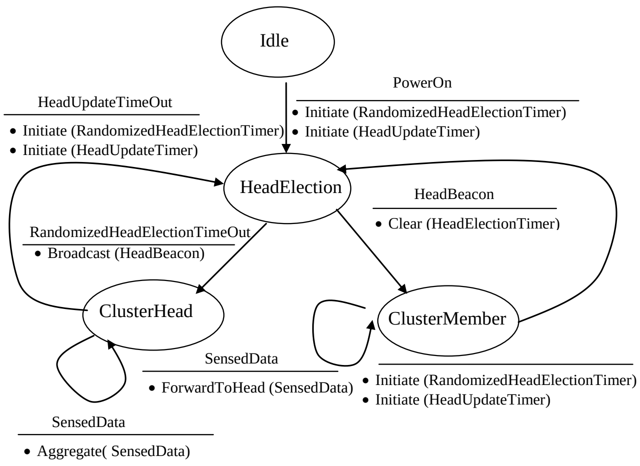

Idle
PowerOn
• Initiate (RandomizedHeadElectionTimer)
• Initiate (HeadUpdateTimer)
HeadUpdateTimeOut
• Initiate (RandomizedHeadElectionTimer)
• Initiate (HeadUpdateTimer)
HeadElection
HeadBeacon
• Clear (HeadElectionTimer)
RandomizedHeadElectionTimeOut
• Broadcast (HeadBeacon)
ClusterHead
ClusterMember
SensedData
• ForwardToHead (SensedData)
• Initiate (RandomizedHeadElectionTimer)
• Initiate (HeadUpdateTimer)
SensedData
• Aggregate(SensedData)

Figure 5. FSM for the clustering algorithm

B. Rule-Based Sensor Nodes De-synchronization

De-synchronization implies that sensor nodes perfectly interleave periodic events to occur in a round-robin schedule. It is useful in that it enables the sensor nodes to evenly distribute sampling burden in a group of nodes, schedule sleep cycles, or organize a collision-free TDMA schedule for transmitting wireless messages [13]. DESYNC, proposed by Julius, Ian, Ankit and Radhika [13], is a biologically-inspired self-maintaining algorithm for de-synchronization in a single-hop network. In this paper, the algorithm of the DESYNC will is implemented on the REED. Due to limitations of space, for detailed description of the DESYNC, please refer to [13].

Fig. 6 describes the behaviour of each sensor node in order to achieve de-synchronization, from which it can be seen that only four rules running on each node are enough to achieve the emergent de-synchronization of the sensor nodes in a fully distributed way. Table 2 lists the four rules derived from Fig. 6.

TABLE I. RULES FOR SENSOR NODES CLUSTERING

Rule 1 is triggered when the sensor node is powered on.

R-1 = power_on
[TRUE; Initiate (RandomizedLeaderElectionTimer),
Initiate (LeaderUpdateTimer)]

Rule 2 is triggered for cluster head election.

R-2 = leader_election_timeout
[TRUE; Broadcast (LeaderBeacon)]

Rule 3 is triggered for cluster members.

R-3 = leader_beacon
[TRUE; Clear (LeaderElectionTimer)]

Rule 4 is triggered for cluster head to aggregate data.

R-4 = sensed_data
[Head; Aggregate (SensedData)]

Rule 5 is for cluster members to send data to its head.

R-5 = sensed_data
[Member; ForwardToHead (SensedData)]]

Rule 6 and 7 are triggered for cluster head updating.

R-6 = leader_update_timeout
[Head; SendToHost (AggregatedData),
Initiate (RandomizedLeaderElectionTimer),
Initiate (LeaderUpdateTimer)]

R-7 = leader_update_timeout
[Member; Initiate (RandomizedLeaderElectionTimer),
Initiate (LeaderUpdateTimer)]

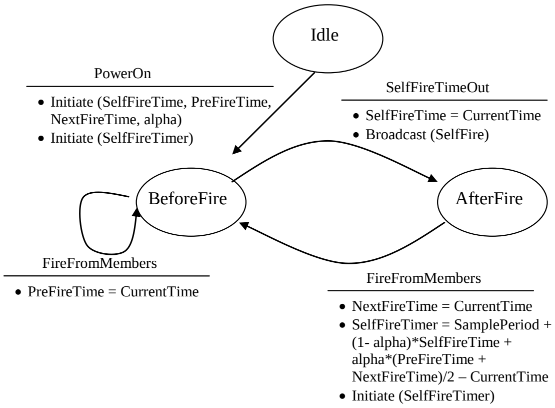

Idle
PowerOn
• Initiate (SelfFireTime, PreFireTime, NextFireTime, alpha)
• Initiate (SelfFireTimer)
SelfFireTimeOut
• SelfFireTime = CurrentTime
• Broadcast (SelfFire)
BeforeFire
AfterFire
FireFromMembers
• PreFireTime = CurrentTime
FireFromMembers
• NextFireTime = CurrentTime
• SelfFireTimer = SamplePeriod + (1- alpha)*SelfFireTime + alpha*(PreFireTime + NextFireTime)/2 – CurrentTime
• Initiate (SelfFireTimer)

Figure 6. FSM for the de-synchronization algorithm

TABLE II. RULES FOR SENSOR NODES DE-SYNCHRONIZATION

Rule 1 is triggered when the sensor node is powered on.

R-1 = power_on
[TRUE; Initiate (SelfFireTime, PreFireTime,
NextFireTime, alpha),
Initiate (SelfFireTimer)]

Rile 2 is triggered when the sensor node receives firing signals from its neighbour before it fires

R-2 = fire_from_members
[BeforeFire; PreFireTime = CurrentTime]

Rile 3 is triggered when the sensor node fires

R-3 = self_fire_timeout
[TRUE; SelfFireTime = CurrentTime]

Rile 4 is triggered when the sensor node receives a firing signal after its neighbour before it fires

R-4 = fire_from_members
[AfterFire; NextFireTime = CurrentTime,
SelfFireTimer = SamplePeriod + (1 - alpha) * SelfFireTime + alpha * (PreFireTime + NextFireTime)/2 – CurrentTime,
Initiate (SelfFireTimer)]

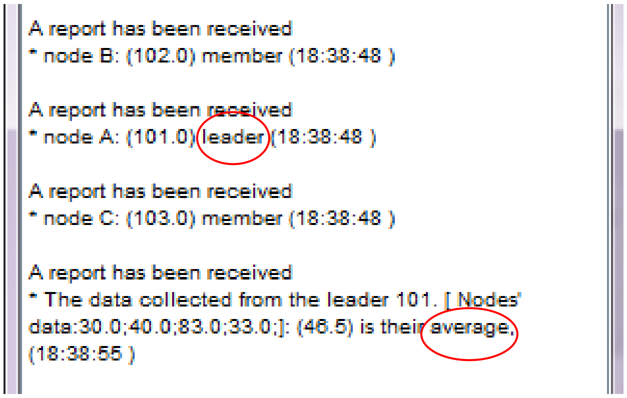
VI. PROTOTYPE IMPLEMENTATION AND EVALUATION

The prototypes of the two cases have been implemented based on the general structure as shown in Fig. 3. To ease the implementation, the gateway is not included and the underlying communications among the PS and PNs are TCP/IP. However, the interfaces provided by the REED middleware are kept the same.

A. *Evaluation of the Clustering Algorithm*

To evaluate the rule-based clustering mechanism, a prototype containing three PNs in a single-hop network has been implemented. The cluster head update period is set as 12 seconds and the sampling period is 5 seconds. Two tests are carried out:

1. Formation of cluster heads: Fig. 7 shows the debugging information on the clustering algorithm, from which it can be seen that in this specific period, node A, with its ID being 101.0, becomes the cluster head/leader, and aggregated information (average of the sensed data) has been collected and sent by node A, the cluster head/leader.
2. Distribution of the cluster head: the system is tested for one hour and 20 minutes which equals to 400 cluster head update periods. Fig. 8 provides the information on the distribution of the cluster head across the PNs, where Node A was elected 135 times, Node B 138 times and Node C 137 times. It presents an overall uniform distribution which demonstrates that the process overhead imposed on the cluster heads are balanced among all the sensor nodes.
3. The test results show that the clustering mechanism, with the features being those mentioned in Section V, has been implemented on the REED.



```
A report has been received
* node B: (102.0) member (18:38:48 )

A report has been received
* node A: (101.0) leader (18:38:48 )

A report has been received
* node C: (103.0) member (18:38:48 )

A report has been received
* The data collected from the leader 101. [ Nodes'
data:30.0;40.0;83.0;33.0;]: (46.5) is their average,
(18:38:55 )
```

Figure 7. Debugging information on the clustering mechanism

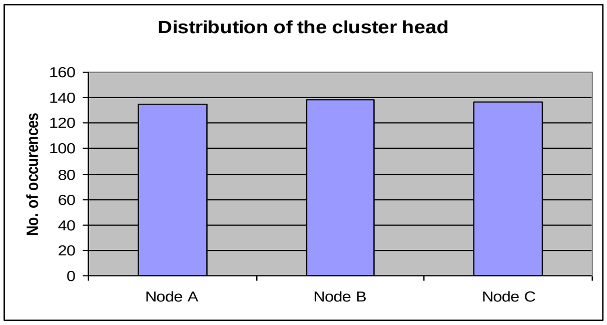
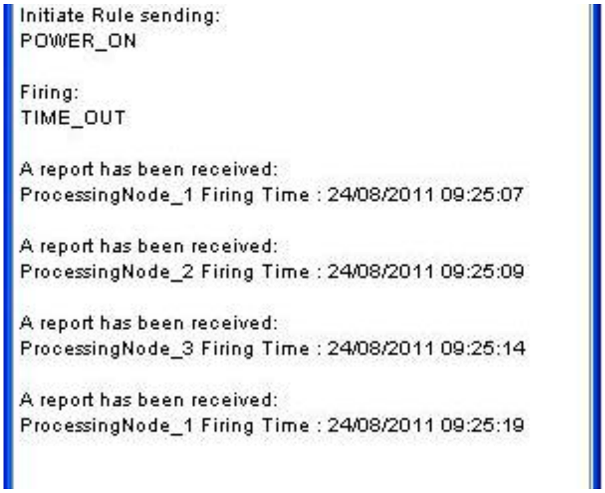


Figure 8. Distribution of the cluster head

B. *Evaluation of the De-synchronization Algorithm*

To evaluate the de-synchronization algorithm, a prototype containing three PNs in a single-hop network has been implemented. The sampling period for each PN is set as 12 seconds and alpha is set as 0.95. By dividing the sampling period by the number of the PNs, the desired time slot size, which in this case is 4 seconds, can be obtained. Fig. 9 shows the debugging information on the de-synchronization algorithm, from which the firing time difference between two time-wise adjacent nodes, i.e. the time slot size, can be derived. Time slot size is the core evaluation metric for the de-synchronization mechanism.

Fig. 10 illustrates the firing time differences between two PNs over time. It can be seen that at the very beginning, deviation from the desired slot size is non- negligible. This is because each PN initially starts their sampling tasks at random time. By running the de-synchronization algorithm, the firing time differences converge to the desired slot size quickly.



```
Initiate Rule sending:
POWER_ON

Firing:
TIME_OUT

A report has been received:
ProcessingNode_1 Firing Time : 24/08/2011 09:25:07

A report has been received:
ProcessingNode_2 Firing Time : 24/08/2011 09:25:09

A report has been received:
ProcessingNode_3 Firing Time : 24/08/2011 09:25:14

A report has been received:
ProcessingNode_1 Firing Time : 24/08/2011 09:25:19
```

Figure 9. Debugging information on the de-synchronization

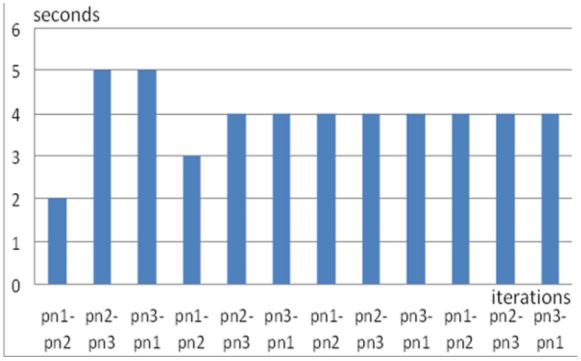


Figure 10. Firing time differences between two nodes

The system was running for 10 hours to test the dynamic behaviour of the de-synchronization algorithm. It was observed that the time slot size may oscillate when reaching the desired value, as shown in Fig. 11. This is mainly due to the choice of alpha, the degree of accuracy provided by the Java libraries, and etc. To get rid of the oscillations, for R-4, no adjustment on the next firing time will be applied if:

$$\left| SelfFireTime - (PreFireTime + NextFireTime) / 2 \right| \leq Threshold$$

In our case, the threshold is set as 55 milliseconds. After this modification, on noticeable oscillations were observed when the system is running for 10 hours.

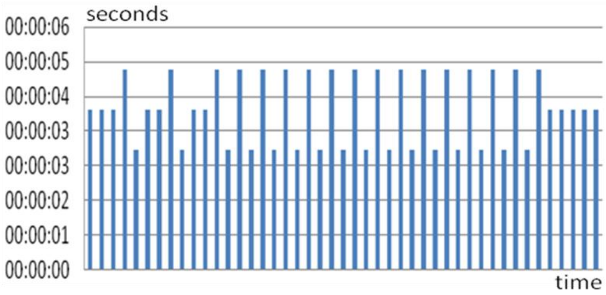


Figure 11. Oscillations of the time slot size

It should be noted that to realize the bio-inspired de-synchronization, only four rules are needed to be running on each node, which makes easier developing bio-inspired mechanisms on resource constraint sensor nodes.

VII. CONCLUSION AND FUTURE WORK

In this paper, the REED, a rule-based WSN middleware is briefly described. A general method to develop WSN applications is given which uses FSM as a bridge between application logics and the rules running on the REED. Especially we argue that rule-based middleware is especially useful for implementing bio-inspired mechanisms, such as self-organization, on WSNs. Prototypes of two WSN applications: sensor nodes clustering and de-synchronization, are implemented on the REED following the proposed developing method. The test results demonstrate the usability of the REED, the effectiveness of the proposed developing method, and especially the advantages of the rule-based realization of bio-inspired mechanisms. Further, we stress-out that the REED provides a framework that makes the application development more straightforward.

To further evaluate the effectiveness of the application development method proposed by this paper, in the future, we aim to extend our research by on the one hand, implementing more existent bio-inspired mechanisms on the REED; and on the other hand, developing real-world applications using the proposed FSM-based rules construction method.

REFERENCES

[1] PROSEN: http://www.cs.stir.ac.uk/~kjt/research/prosen/ [retrieved: August, 2012]

[2] L. Mottola and G. P. Picco, "Programming Wireless Sensor Networks: Fundamental Concepts and State of the Art", ACM Computing Surveys (CSUR) Volume 43 Issue 3, April 2011

[3] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey", Computer Networks,Vol. 52 , Issue 12, pp. 2292-2330, 2008

[4] M. Kuorilehto, M. Hannikainen, and T. D. Hamalainen, "A Survey of Application Distribution in Wireless Sensor Networks", Journal on Wireless Communications and Networking, Vol. 5, pp. 774–788, 2005

[5] X.Fei and E. Magil, "Rule Execution and Event Distribution Middleware for PROSEN-WSN", SENSORCOMM-2008, pp.580-585, 2008

[6] X. Fei and Z. Yu, "Development of a Rule-Based Wireless Sensor Network Middleware", Proceedings of the 16th International Conference on Automation & Computing, pp. 13-18, Sept. 2010

[7] K. Terfloth, G. Wittenburg, and J.Schiller, "FACTS - A Rule-Based Middleware Architecture for Wireless Sensor Networks", COMSWARE 2006, New Delhi, India, January 2006

[8] Gumstix: http://gumstix.com/ [retrieved: August, 2012]

[9] JamVM: http://jamvm.sourceforge.net/ [retrieved: August, 2012]

[10] K. Terfloth, "Doctoral Dissertation: A Rule-Based Programming Model for Wireless Sensor Networks", Freie Universitat, Berlin. June 2009.

[11] I. Wokoma, L. Shum, L. Sacks, and A. Marshall, "A Biologically-Inspired Clustering Algorithm Dependent on Spatial Data in Sensor Networks", Proceeedings of the Second European Workshop on Wireless Sensor Networks, pp. 386 - 390, 2005

[12] W. A. Geoffrey, T. Geetika, P. Ankit, W. Matt, and N. Radhika, "Firefly-Inspired Sensor Network Synchronicity with Realistic Radio Effects", Sensys'05, pp. 142–153, November, 2005

[13] D. Julius, R. Ian, P. Ankit, and N. Radhika, "DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks", IPSN, pp. 11-20, April 2007.

[14] P. Boonma and J. Suzuki, "BiSNET: A biologically-inspired middleware architecture for self-managing wireless sensor networks", International Journal of Computer and Telecommunications Networking, Vol. 51 Issue 16, pp. 4599-4616, November, 2007

[15] R. H. Jacobsen, Q. Zhang, and T. S. Toftegaard, "Bioinspired Principles for Large-Scale Networked Sensor Systems: An Overview", Sensors, Vol. 11, pp. 4137-4151, 2011

ISBN: 978-1-61208-207-3

A Java Library for Event-Driven Communication in Power-Manageable Reactive Sensor Nodes

Emanuele Lattanzi and Alessandro Bogliolo
Department of Base Sciences and Fundamentals - DiSBef
University of Urbino - Italy
Email: emanuele.lattanzi, alessandro.bogliolo@uniurb.it

***Abstract*—The energy efficiency of wireless sensor networks strongly depends on the possibility of exploiting the idleness of their nodes. In principle, idle periods could be fully exploited by making use of ultra low power micro controller units (MCUs) and power manageable network interfaces which provide a wide range of sleep states with sub micro Watt power consumption. One of the key issues, however, is to avoid to keep sensor nodes busy when they could be idle, thus reducing the opportunity of dynamic power management. This issue is particularly serious in case of sensor nodes running a virtual runtime environment, since the virtual machine (VM) is seen by the scheduler of the underlying operating system (OS) as a process which is always active in spite of the idleness of the threads running on top of it. On the other hand, the benefits of virtualization in terms of abstraction and usability motivates the development of sensor nodes with power manageable virtual runtime environments. Promising results have been recently achieved in this direction by using a modified version of the Darjeeling VM on top of Contiki OS. This paper moves a step forward by introducing *VirtualSense*, an event-driven communication library for the Darjeeling VM which exhibits two distinguishing features. First, it is general enough to enable the implementation of advanced communication protocols in Java. Second, its event-driven nature makes it possible for a Java thread to react to incoming messages without keeping the MCU busy while waiting.**

***Keywords*-Event drive, communication library, reactive sensor, low-power;**

I. INTRODUCTION

The energy efficiency of a wireless sensor network (WSN) depends on the capability of its nodes to adapt to time-varying workload conditions by turning off unused components and by dynamically tuning the power-performance tradeoff of the used ones. *Dynamic power management* (DPM) is a wide research field which has brought, on one hand, to the design of power manageable components featuring multiple operating modes and, on the other hand, to the development of advanced DPM strategies to exploit them. The best DPM strategy is the one which meets the performance constraints imposed by the application with minimum power consumption. Apart from the fine tuning of the power-performance tradeoff of the active states, the main power-saving opportunities come from idle periods, which allow the power manager to take advantage to ultra low-power inactive modes. Exploiting idleness is mandatory in wireless sensor nodes, which spend most of their time waiting for external events or for monitoring requests, and which are often equipped with *energy harvesting* modules which promise to grant them an unlimited lifetime [5] as long as their average power consumption is lower than the average power they take from the environment.

A suitable answer to power management needs in WSNs is provided by state-of-the-art ultra low power micro controller units (MCUs), which feature a wide range of active and inactive power states while also providing enough memory and computational resources to run a virtual machine (VM) on top of a tiny operating system (OS). Virtualization adds to the simplicity and portability of applications for WSNs at the cost of increasing the distance between hardware and software, which might impair the effectiveness of DPM both for the limited control of the underlying hardware offered by the virtual runtime environment, and for the limited visibility of the actual activity offered by the VM. In fact, the VM is usually viewed by the scheduler of the embedded OS as a process which is always active in spite of the possible idleness of its threads. Two solutions have been proposed to address these issues. The first one is provided by *bare-metal* VMs, which runs directly on top of the MCU without any OS [6], [7], [8], at the cost of loosing portability. The second one is provided by full-fledged software stacks specifically designed for power manageable sensor nodes in order to make it possible to take DPM decisions directly from the runtime environment and to grant to the OS scheduler full visibility of the idleness of the virtual tasks. This is the approach adopted in this paper, starting from a recently proposed architecture [4] base on Darjeeling VM [2] and Contiki OS [1].

The effectiveness of DPM risks to be impaired, however, by the paradigm adopted for inter-node communication. Although a sensor node is primarily designed to sense a physical quantity and to send a message to the sink to report the measured value, most of the nodes in the network act as routers to relay other nodes' messages towards the sink. Moreover, in self-adapting sensor networks all the nodes have to be able to receive *interest* messages from the sink and to broadcast them to their neighbors in order to be assigned a task and to update their own routing tables

[9]. While all other activities can be either scheduled or triggered by external interrupts, receiving a message is an asynchronous event which needs to be carefully handled. In the Darjeeling VM a thread waiting for a message is suspended and rescheduled at next timer interrupt (by default, after 10ms) to look for an incoming message in the network buffer. This polling mechanism keeps the MCU busy regardless of the idleness of the node.

This problem could be solved by implementing a smarter protocol in the Contiki OS, but this would fail in the attempt of making the sensor node completely programmable in Java. An alternative approach would consist of increasing the timer interrupt in order to grant to the OS enough time to put the MCU into an inactive state, but this would reduce time resolution and increase the average response time of the node.

This paper presents a communication library for the Darjeeling VM which makes available event-driven send and receive primitives to achieve two goals: to provide the generality required to enable the implementation of advanced communication protocols in Java, and to make it possible for a Java thread to react to incoming messages without keeping the MCU busy while waiting. Experimental results show that the proposed library enables the full exploitation of the low-power states of the MCU with a negligible time overhead.

The rest of the paper is organized as follows: Section II presents the architecture of a node with virtual runtime environment, Section III introduces the Java library, Section IV presents a representative case study and some measurements, Section V concludes the work.

II. Node Architecture

We consider the architecture of a sensor node composed of: a power-manageable MCU, an embedded OS (namely, Contiki OS [1]), and a tiny Java-compatible VM (namely, Darjeeling VM [2]).

As briefly discussed in the introduction, the presence of an OS provides a suitable decoupling between HW and SW which makes the approach described in this paper independent of the underlying MCU. Nevertheless, in the following we refer to Texas Instruments' MSP430F54xxa MCUs, which are highly representative of state-of-the-art power-manageable MCUs providing four categories of power states: *Active*, *Standby*, *Sleep*, and *Hibernation*. In Standby mode the CPU is not powered, but the clock system is running and the unit is able to wakeup itself by means of timer interrupts. In Sleep mode both the CPU and the clock system are turned off, so that the unit wakes up only upon external interrupts. In Hibernation even the memory system is switched off, so that there is no data retention and a complete reboot is required at wakeup.

Contiki [1] is a real-time embedded OS particularly suited for sensor nodes. It has an inherent event-driven structure which reduces the overhead of periodic wake-ups by making the interrupt handler aware of the next time at which a process has to be resumed by a timer interrupt. This allows the MCU to go back to sleep without invoking the scheduler in case of premature wakeup. The only exception is represented by processes waiting for external events, which need to be resumed whenever an interrupt arrives.

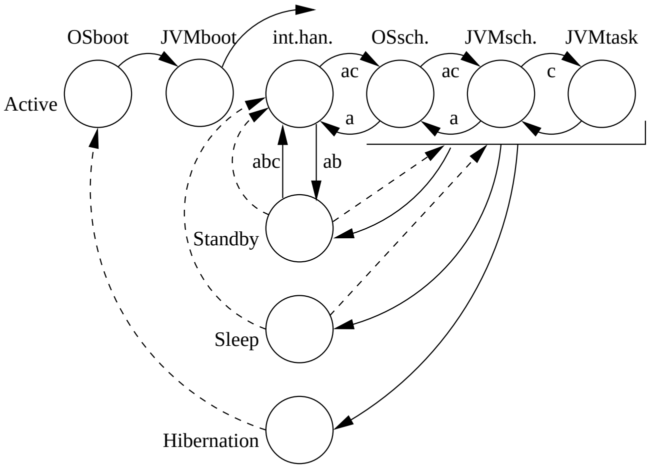


Figure 1. Power state diagram of a power-manageable MCU running the Darjeeling VM on top of Contiki OS.

Darjeeling [2] is a VM for wireless sensor networks which supports a significant subset of the Java libraries while running on 8-bit and 16-bit MCUs with at least 10kbytes of RAM. The VM runs on top of Contiki as a single process, in spite of its multi-threading support. Hence, the VM has its own scheduler to switch among the active threads according to a preemptive round-robin policy. Whenever the running thread is suspended the scheduler waits for the a timer interrupt before resuming execution.

A. Power-manageable virtual sensor node

Figure 1 shows the power-state diagram of a power-manageable MCU running the Darjeeling VM on top of Contiki OS. All the states aligned at the top of the diagram represent the macro steps required at wake-up to resume the execution of a Java thread. Wake-ups can be triggered either by timing interrupts (solid arrows) or by external events (dashed arrows). It is worth noticing that self wake-ups are enabled only in Standby mode, while external events are required to trigger wake-ups from Sleep and Hibernation.

Contiki puts the MCU in Standby whenever all its running processes are waiting for external events or timer interrupts. In order to keep control of the elapsed time it sets a periodic timer interrupt which wakes up the MCU every 10ms to allow the interrupt handler to evaluate if there are processes to be resumed. If this is the case the control is passed to the scheduler, otherwise the MCU is turned off again. As previously stated, the VM needs to be resumed at each timer

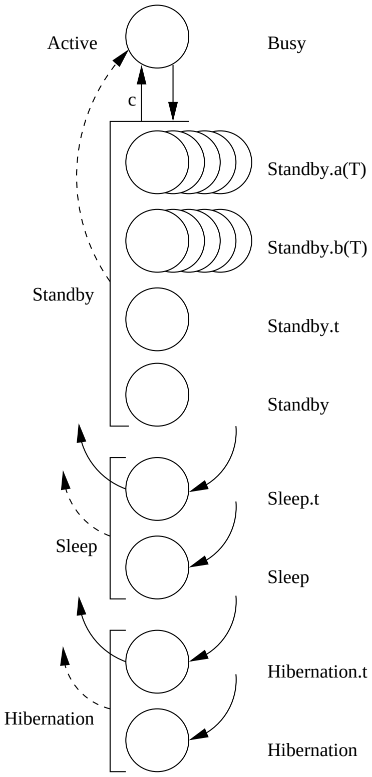


Figure 2. Power state diagram of a power-manageable virtual sensor node [4].

interrupt in order to check for the status of its threads which are not visible from the OS.

Figure 1 makes use of labels a, b, and c to denote the transitions taken upon a timer interrupts in case of: a) VM with no tasks to be resumed, b) OS with no processes to be resumed, and c) virtual task to be resumed. While case c) represents a useful wakeup, cases a) and b) should be avoided in order to save power. To this purpose, a modified software stack was recently proposed [4] which: i) avoids periodic wake-ups by making the OS aware of the time of the next event scheduled by the VM and by tuning the timer interrupt accordingly, ii) supports hibernation by saving and restoring the heap of the VM, and iii) maintains timing information in deeper low power states by means of an external ultra low-power real-time clock.

The modified software stack builds a power manageable virtual sensor node which makes directly available from the Java runtime environment all the power states represented in Figure 2. The first two Standby states are parameterized by the timer interrupt (T) which can be adjusted to explore the tradeoff between power consumption and reactivity. Suffix t is used to denote the usage of the external real-time clock to provide accurate timing information in spite of the lack of the internal clock. Table I reports the power-performance tradeoffs offered by the power states of the software stack

State name	**Power [μW]**	**WUt [ms]**	**WUe [μJ]**	**SDt [ms]**	**SDe [μJ]**
Active	6600μW	n.a.	n.a.	n.a.	n.a.
Standby.a(T)	$4.5 + 153.57/T$	23.41	153.57	–	–
Standby.b(T)	$4.5 + 0.33/T$	23.41	153.57	–	–
Standby.t	$4.5 + 0.3$	23.41	153.57	–	–
Standby	4.5	23.41	153.57	–	–
Sleep.t	*1.5+0.3*	23.41	153.57	–	–
Sleep	*1.5*	23.41	153.57	–	–
Hibernation.t	*0.1+0.3*	560	2312.8	78.8	606.76
Hibernation	*0.1*	560	2312.8	78.8	606.76

Table I
Characterization of the power states of the virtual sensor node architecture [4] run on a Texas Instruments' MSP430F2618 powered at 3V and clocked at 16MHz with a 4Kbyte virtual machine heap.

run on top of a Texas Instruments MSP430F2618 MCU powered at 3V and clocked at 16MHz. Each inactive state is characterized by: power consumption (Power), wake-up time and energy (WUt and WUe), and by shut-down time and energy (SDt and SDe). Transitions energies (times) lower than 0.01μJ (0.01ms) are not reported. software each power state

B. Communication issues

Communication across the radio channel is handled by the `Radio` class of the Darjeeling VM, which makes available a `receive()` method to be invoked by any Java thread waiting for a message. As soon as the method is invoked, the Java thread is suspended by the scheduler of the VM. If there are no other threads ready to execute, the Darjeeling process is suspended as well and rescheduled by the OS at next timer interrupt (i.e., at most after 10ms). Referring to the state diagram of Figure 1, as long as the message does not arrive, the MCU keeps waking up at each timer interrupt and executing the interrupt handler routine, the OS scheduler, and the VM scheduler before deciding to go back in Standby mode. This is a power-consuming self loop which is labeled with a in Figure 1 and schematically represented by macro state Standby.a(T) in Figure 2. No other low power states can be exploited while waiting for a message.

Looking at Table I, it is worth noticing that power consumption of Standby.a(T) is several orders of magnitude higher than that of Sleep and Hibernation. Moreover, the wake-up time is larger than 10ms, so that the MCU would stay always while waiting for a message unless a longer timer interrupt was set in the modified stack. The minimum timer interrupt which could allow the exploitation of Standby mode is $T = 25ms$.

The event-driven communication library presented in next section solves this issue by enabling the exploitation of all the low-power states of a power-manageable virtual sensor node waiting for incoming messages.

ISBN: 978-1-61208-207-3

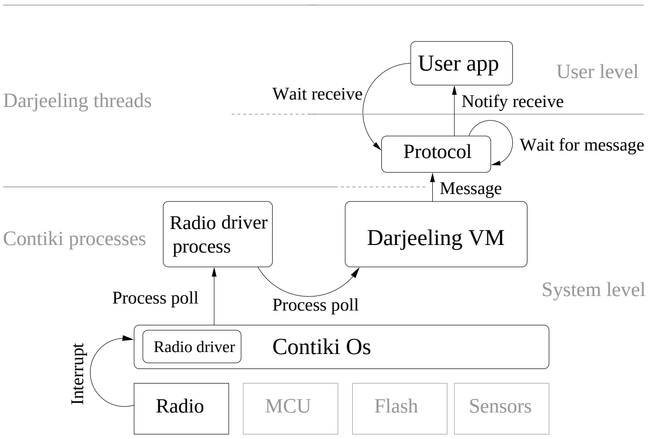


Figure 3. VirtualSense software architecture

III. VirtualSense communication

The software architecture of the proposed communication framework is shown in Figure 3, where arrows are used to represent the event chain triggered by the reception of an incoming packet. The figure points out the interactions between user-level and system-level execution flows, as well as those between Contiki processes (namely, `Radio driver process` and `Darjeeling VM`) and Darjeeling threads (namely, `Protocol` and `User app`).

While waiting for an incoming packet all the processes are blocked and they do not consume any computational resource. When a packet is received by the radio device, the `Radio driver` interrupt handler issues a `PROCESS_EVENT_POLL` for the `Radio driver process` which was waiting for it. At this point the scheduler of Contiki wakes up the `Radio driver process` which: takes the packet from the radio device buffer, forwards it to the Contiki network stack, issues a new `PROCESS_EVENT_POLL` for the `Darjeeling VM`, and releases the CPU while waiting for next packet. The CPU is then taken by the `Darjeeling VM` process, which resumes the execution of the `Protocol` thread which was blocked for I/O. The `Protocol` plays the role of consumer by popping the incoming message from the Contiki network stack, which acts as a buffer in the producer-consumer interaction.

The event chain described so far is general enough to enable the implementation of any kind of communication protocol either within the `Protocol` thread or at application level. Depending on the protocol adopted and on its implementation, received packets can either be handled directly by the `Protocol` thread or be forwarded to the `User app` waiting for them.

Sending a packet is much simpler than receiving it: the `User app` which needs to send a message invokes the `send()` method of the `Protocol`, which puts the packet on the Contiki network stack without involving the `Radio driver process`.

In the following we outline the three packages developed to extend the Darjeeling Java libraries in order to support the event chains described above: i) `javax.virtualsense.radio`, containing the static native methods used to communicate with the radio device; ii) `javax.virtualsense.network`, making communication primitives available to user-level Java threads; iii) `javax.virtualsense.concurrent`, providing synchronization primitives. A simplified class diagram is shown in Figure 4.

A. Radio package

The `radio` package contains the `Radio` class (represented in Figure 4) and some other classes used to handle exceptions. The `Radio` class exports static native methods which directly interact with the platform radio driver and with the network stack of Contiki OS: a method to perform radio device configuration and initialization (`init()`), unicast and broadcast send methods (`send()`, and `broadcast()`), a blocking receive methods (`receive()`, and two methods to get the sender and receiver IDs (`getSenderId()` and `getReceiverId()`). All the methods are protected, in order to be used only through the `Protocol` class, which is part of the `network` package.

Unicast and broadcast send methods make use of the Contiki `unicast_conn` and `broadcast_conn` network connections from the `rime` network stack. The `receive()` method suspends the calling Java thread by putting it in a waiting queue and acquires a lock on the radio device preventing the power manager to shut down the network device. Whenever a radio message is received from the Contiki network stack two different call-backs are activated depending on the nature of the received message: broadcast call-back or unicast call-back. Both call-backs wake up the suspended Java thread, set the `senderId` and `receiverId` attributes, and release the device lock.

B. Network package

The `network` package acts as a middleware layer which lies between the system level radio package and user-level applications. In particular, this package contains an abstract `Protocol` class, providing a communication protocol skeleton, and a `Network` class, providing a public interface to make communication primitives available to user-level threads.

The `Network` implements the *singleton* pattern, so that it can only be instantiated by means of its `init()` method, which can be invoked with or without a given protocol (i.e., an instance of a subclass of `Protocol`. If no protocol is specified, then the `NullProtocol` is used by default. After network initialization, user level threads can call

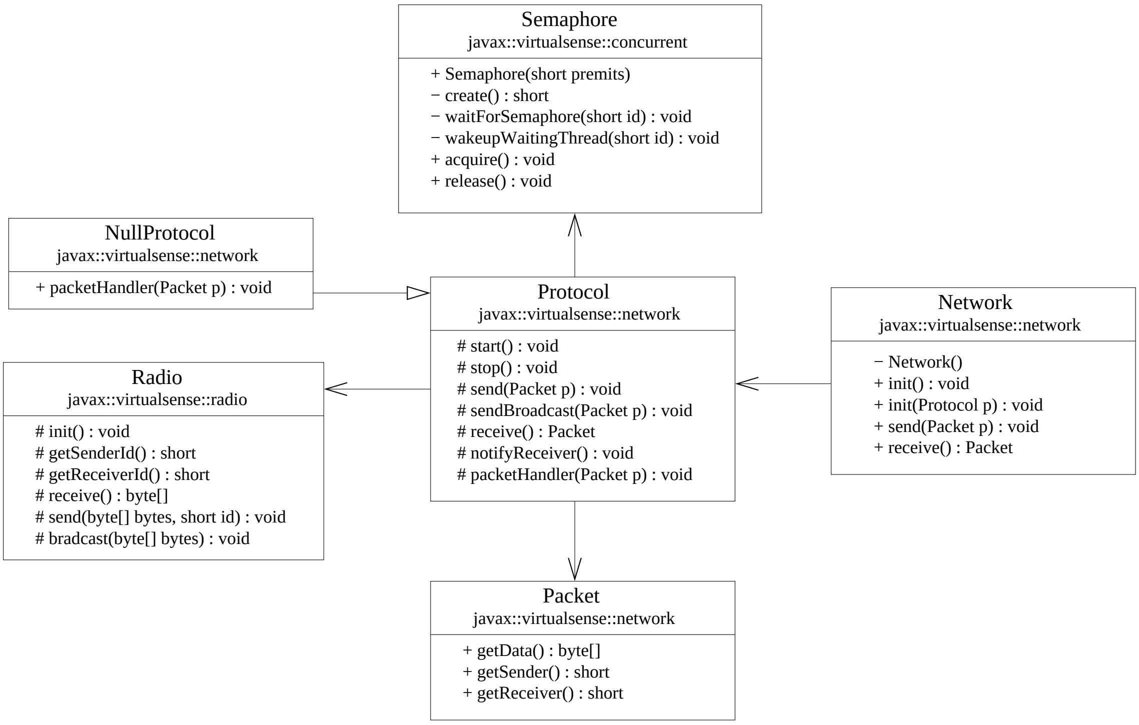

Semaphore
javax::virtualsense::concurrent
+ Semaphore(short premits)
− create() : short
− waitForSemaphore(short id) : void
− wakeupWaitingThread(short id) : void
+ acquire() : void
+ release() : void
NullProtocol
javax::virtualsense::network
+ packetHandler(Packet p) : void
Protocol
javax::virtualsense::network
# start() : void
# stop() : void
# send(Packet p) : void
# sendBroadcast(Packet p) : void
# receive() : Packet
# notifyReceiver() : void
# packetHandler(Packet p) : void
Network
javax::virtualsense::network
− Network()
+ init() : void
+ init(Protocol p) : void
+ send(Packet p) : void
+ receive() : Packet
Radio
javax::virtualsense::radio
# init() : void
# getSenderId() : short
# getReceiverId() : short
# receive() : byte[]
# send(byte[] bytes, short id) : void
# bradcast(byte[] bytes) : void
Packet
javax::virtualsense::network
+ getData() : byte[]
+ getSender() : short
+ getReceiver() : short

Figure 4. VirtualSense communication class diagram. Public methods are denoted means ”+”,l while protected and private methods are denoted by ”#” and ”-”, respectively.

`send()` and `receive()` methods to communicate. These two methods provide a public interface to the corresponding methods of the `Protocol` class.

`Protocol` is an abstract class which has to be subclassed in order to implement specific routing strategies. The class maintains as local properties the routing table and the queue of received packets. In order to decouple system-level and user-level packet reception tasks, the `Protocol` class provides a dedicated thread (instantiated and launched within the class constructor) which runs a loop containing a call to Radio.receive(). The thread is suspended on this call until a packet is received, as described in Section III-A. Upon reception of an incoming packet the thread resumes execution and it calls the `packetHandler()` method, an abstract method that has to be implemented in any Protocol subclass.

Methods `receive()` and `notifyReceiver()` provide the means for using the event-driven reception mechanism from user-level threads. To this purpose, an application which needs to receive a packet from the radio device invokes the `Network.receive()` method which, in turn, calls `Protocol.receive()` which suspends the calling thread on a counting semaphore. Upon reception of a packet to be forwarded to the waiting application, the `Protocol` invokes `notifyReceiver()` to release a permit on the semaphore. From the implementation stand point, the invocation of `notifyReceiver()` has to be placed inside `packetHandler()`, which is the method where the actual routing protocol is implemented. The default `NullProtocol` does nothing but invoking this method to forward to the applications all incoming packets.

C. *Concurrent package*

The `concurrent` package provides a robust and efficient way to manage thread synchronization. In particular the `Semaphore` class implements a standard counting semaphore based on a waiting queue. Any thread waiting for a semaphore permit is suspended by the VM and moved to the semaphore waiting queue. In this way it allows the power manager to shutdown the MCU. As soon as a new permit is available on the semaphore, the waiting thread is woken up by removing it from the waiting queue. Thread suspension and wake up are implemented through native methods `waitForSemaphore()` and `wakeupWaitingThread()`, respectively, which directly interact with the VM scheduler and manage thread displacement.

IV. Case Study and Experimental Results

In this section we show, with a practical example, how to use the communication library presented in Section III. We use as a case study a sensor network programmed to perform a periodic monitoring task: each node in the network senses the target physical quantity once per second and sends the measured value to the sink. The sink is nothing but a sensor node connected to a desktop PC by means of the serial port. All other sensor nodes act also as routers, implementing a self-adapting minimum path routing protocol.

```
01 import javax.virtualsense.network.*;
02
03 public class MinPathProtocol extends Protocol{
04
05   private short minHops = Short.MAX_VALUE;
06   private short epoch   =  0;
07
08   protected void packetHandler(Packet p){
09      if(p instanceof InterestMsg){
10         InterestMsg interest = (InterestMsg)p;
11         if(interest.getEpoch() > this.epoch){
12            this.epoch = interest.getEpoch();
13            super.bestPath = -1;
14            this.minHops = Short.MAX_VALUE;
15         }
16         if(interest.getHops() < this.minHops){
17            this.minHops = interest.getHops();
18            super.bestPath = interest.getSender();
19            interest.setHops(interest.getHops()+1);
20            super.sendBroadcast(interest);
21         }
22      }else if(p instanceof DataMsg) {
23         DataMsg data = (DataMsg)p;
24         if(data.toForward())
25            super.send(data);
26         else
27            super.notifyReceiver();
28      }
29  }// end method
30 }//end class
```

Figure 5. Minimum path algorithm implementation

The sink collects all the measurements and triggers period updates of the routing tables by sending an broadcast *interest* message (`InterestMsg`) to the network according to a *directed diffusion* paradigm [9]. The interest contains a progressive counter, called `epoch`, which is used by the nodes which receives and forward the interest message to verify its freshness. In addition, it contains the number of hops from the sink, which is incremented at each hop to allow sensor nodes to identify the best path. Figure 5 reports the Java code of the `MinPathProtocol` class which extends the `Protocol` and overrides abstract method `packetHandler()` to implement the minimum path directed diffusion algorithm.

Whenever a new packet is received, the `packetHandler()` checks if it contains an interest message (Figure 5, line 09) or a data message (line 22). In case of an interest, its epoch is compared with the previous one (line 11) in order to reset the routing table in case of new epoch (lines 12-14). In the directed diffusion min path protocol the routing table is nothing but the ID of the neighboring node along the best path to the sink. Such an ID is stored in `bestPath`, which is updated with the ID of the sender of last interest message whenever the number of hops annotated in the message is lower than the current value of `minHops` (lines 16-19). In this case the interest message is also forwarded (line 20).

Data packets are either to be forwarded to the sink through `bestPath` (line 25) or to be notified to user-level applications possibly waiting for them (line 27). According to the directed diffusion algorithm sensor nodes never play the role of recipients of data messages. Nevertheless, line 27 has been added in Figure 5 as an example of user-level communication.

A. Performance overhead

The proposed architecture was instrumented in order to measure the software overhead introduced by the high-level implementation of the communication protocol.

In particular we measured Contiki and Darjeeling execution times as separate contributions to the reception event-chain starting from the sleep state. Contiki overhead was measured as the time between the reception of a radio interrupt and the corresponding Darjeeling VM process poll. Darjeeling overhead was measured as time between the wake up of Darjeeling VM process and the delivery of the incoming packet to the user-level application. The results obtained in a prototyping board equipped with an MSP430F5435a running at 16MHz where respectively 3.7ms and 14.4ms for Contiki and Darjeeling software overheads resulting in a total overhead of 18.1ms.

On the other hand the software overhead introduced by the proposed Java library in the sending chain was of 3.4ms.

It is worth mentioning that the Java library introduced in this paper allows the power manager to exploit the waiting times to put the sensor nodes either in Sleep or in Hibernation, depending on the DPM policy adopted. For instance, in case of a sensor node acting as a router handling on average two data packets per second, the exploitation of the Sleep state would reduce the power consumption from the 6600μW of the Active state of our MCU to a measured value of 1620μW, which would tend to a few μW as the monitoring rate decreases.

This example shows how the proposed network library allows the programmer to implement a routing protocol with a few lines of code, enabling the full exploitation of the low-power states of the MCU without impairing the reactivity of the sensor node.

V. Conclusion

The availability of ultra low-power MCUs able to run a virtual machine makes it possible to design power manageable sensor nodes that can be programmed in Java. The separation between the virtual runtime environment and the underlying MCU has been recently bridged by means of a modified software stack which retains the key benefits of virtualization without impairing the effectiveness of dynamic power management. Any node in a wireless sensor network, however, spends most of its operating time waiting for incoming packets. In spite of the idleness of a waiting node, the capability of reacting to an incoming message is often implemented by means of polling mechanisms which keep the MCU busy and avoid the exploitation of its low-power inactive states. This is what happens with the communication primitives made available by the Darjeeling VM running on top of Contiki OS.

This paper has presented an event-driven communication library, called VirtualSense, which provides energy-aware `send()` and `receive()` methods which allow the MCU to exploit inactive low-power states while waiting for incoming packets and to resume the execution of the Java thread as soon as a packet is received. A simple case study has been presented to show that the proposed library makes it possible (and easy) to implement a communication protocol on top of the Java runtime environment without impairing the effectiveness of dynamic power management in ultra low-power sensor nodes.

ACKNOWLEDGMENT

The authors would like to thank Andrea Seraghiti, Massimo Zandri, and NeuNet Cultural Association (http://www.neunet.it/) for their fundamental contribution to the development of the prototype.

REFERENCES

[1] A. Dunkels, B. Gronvall, and T. Voigt, ”Contiki - a lightweight and flexible operating system for tiny networked sensors”, in *Proc. of the IEEE Conf. on Local Computer Networks*, pp. 455-462, 2004.

[2] N. Brouwers, P. Corke, and K. Langendoen. ”Darjeeling, a Java compatible virtual machine for microcontrollers”, in *Proc. of the ACM/IFIP/USENIX Middleware Conference Companion*, pp. 18-23, 2008.

[3] Texas Instruments. ”MSP430x5xx/MSP430x6xx Family User’s Guide”, URL *http://www.ti.com/lit/ug/slau208j/slau208j.pdf*, 2012.

[4] E. Lattanzi and A. Bogliolo, ”Ultra-Low-Power Sensor Nodes Featuring a Virtual Runtime Environment”, to be presented at IEEE ICC E2NETS-2012, 2012.

[5] E. Lattanzi and A. Bogliolo, ”WSN Design for Unlimited Lifetime”, In Yen Kheng Tan (Ed.), *Sustainable Energy Harvesting Technologies: Past, Present and Future*, InTech, 2011.

[6] Rene Muller, Gustavo Alonso, and Donald Kossmann. 2007. ”A virtual machine for sensor networks”. SIGOPS Oper. Syst. Rev, pp. 145-158, 2007

[7] Doug Simon, Cristina Cifuentes, Dave Cleal, John Daniels, and Derek 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 (VEE ’06). ACM, pp. 78-88, 2006.

[8] Philip Levis, David Gay, and David Culler. ”Active sensor networks”. In *Proceedings of the 2nd Symposium on Networked Systems Design & Implementation*, pp. 343-356, 2005.

[9] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. ”Directed diffusion for wireless sensor networking”. IEEE/ACM Trans. Netw. 11, pp. 2-16, 2003.

The Availability and Statistical Properties of Ambient Light for Energy-Harvesting for Wearable Sensor Nodes

Oleg Nizhnik, Ucu Maksudi, Sayaka Okochi, Kohei Higuchi and Kazusuke Maenaka
JST ERATO Maenaka Human-sensing Fusion Project
8111, Shosha 2167, Himeji-shi, Hyogo, Japan
e-mail: oleg@eratokm.jp

***Abstract*—Data for the availability of ambient light for a wrist-attached senor node was gathered. The average energy harvest totaled 0.074 J/cm^2/day when using a 1.8% efficiency, 0.6V DSSC (dye-sensitized solar cell) and 0.48 J/cm^2/day when using a standard 16% efficiency, 0.65V mono-crystalline Si cell. The irradiance averaged for both indoor and outdoor activity of office employees was found to be equivalent to 0.78 W/m^2 of solar spectrum light. At lower irradiation, the DSSC produced a higher open-circuit voltage but harvested less power than the Si solar cell.**

Keywords - solar cell; sensor network; sensor node; wearable; energy harvesting

I. INTRODUCTION

Solar power may be a viable solution for powering wearable sensor nodes. However, there is a scarcity of data on the actual harvestable power for wearable sensor nodes using photovoltaic cells. The actual power depends on the following factors:

Primary (light-related) factors:
The total irradiance (W/m^2)
The angular distribution of light from light sources
The spectrum of the light sources
The daily variability of the light sources
The seasonal variability of light sources

Secondary (harvester-related) factors:
Type of solar cell
Lifestyle of wearer
Harvesting location on body

Because of the many unknowns, solar cells have been considered too unpredictable to serve as a power source for wearable sensor nodes without using additional power management methods [1] [2]. Even for the total irradiance metric, wide and conflicting ranges have been reported, as summarized in Table 1. Therefore, experimental data on solar energy harvesting for wearable sensor nodes attached to the users' wrists is provided in this paper. The reason irradiation data for wearable nodes has not been gathered before is because of the obtrusiveness and unreliability of past wearable sensor nodes. Previous sensor nodes required frequent maintenance, charging, and data downloading, thus affecting behavioral patterns of the person wearing a node. Because the behavioral patterns affect the irradiation measurements, a smaller, more autonomous sensor node was necessary for unbiased measurements.

Recently, there have been several attempted implementations [3][4][5] of a wearable sensor node system utilizing solar cells. Therefore, a detailed assessment of the properties of ambient light is necessary for the proper selection of solar cell type – either high-efficiency at high irradiation level (Si cell) or more stable at lower irradiation and lower efficiency (DSSC).

TABLE I. PREVIOUSLY REPORTED IRRADIANCE LEVELS

Parameters		
Sensor node type	***Indoor irradiance, W/m^2***	***Outdoor irradiance, W/m^2***
Building management [6]	6.7	500
Wearable [7][8]	0.4-38	10-100
Building management[9]	1-5	-
Wearable[10]	1	1000

In this paper, Section II describes measurement conditions and algorithms used for data processing. Section III presents the irradiance measurement results and calculations of the harvestable energy for the DSSC and Si solar cells. Section IV concludes that although the Si solar cell was more effective in overall energy harvesting, the DSSC cell provides more reliable energy harvesting.

II. MEASUREMENT PROCEDURE

The measurement of light levels was done using the Indy2050 DSSC [11] attached to the wearable sensor node prototype LM03 provided by the University of Hyogo. The location of the experiment was Himeji, Japan. A photograph of the sensor node with solar cell is provided in Figure 1. The solar cell was connected directly to the voltage input of the ADC of the sensor node. To convert photocurrent (and hence the irradiance) to a voltage signal, a resistor was connected in parallel with the solar cell.

ISBN: 978-1-61208-207-3

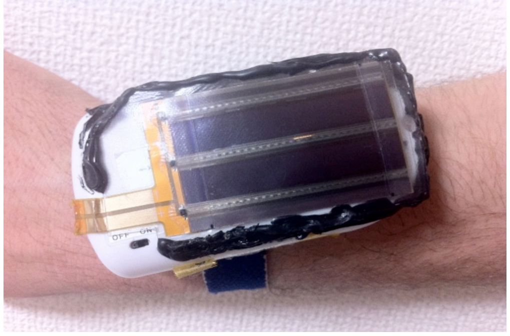


Figure 1. Photo of the measurement setup.

Because the ADC embedded in the sensor node had insufficient dynamic range, two setups were prepared: first with a 100 Ohm resistor, giving a measurement range of 1 to 400 W/m^2, and second with a 10 kOhm resistor, giving a measurement range 0.01 to 4 W/m^2. The two setups were worn by different users (both working in the same lab), and were swapped between users twice per week. A larger number of experimental setups was prepared, but severe hardware problems of the highly-experimental LM03 sensor node resulted in only two setups being operational. The ADC was sampled 20 times per second, and data was stored on a MicroSD card inside the sensor node. Ten days of recorded data (from April 1 to April 11) were downloaded to the PC on April 12, 2012 and processed. Processing included the removal of anomalous data (whenever time stamp showed discontinuity due sensor node being shut down) and simple averaging of samples down to a data rate of 1 sample per second. Also, the data from both setups was merged into a single dataset using the measured photocurrent-to-voltage gain and the ADC's offset voltages. The solar cell efficiency versus irradiance for the Indy2050 DSSC solar cell and typical Si cell was measured before starting energy harvesting measurements. The data on the efficiency of the Indy2050 solar cell and typical Si cell (extracted from [1]) is shown in Figure 2.

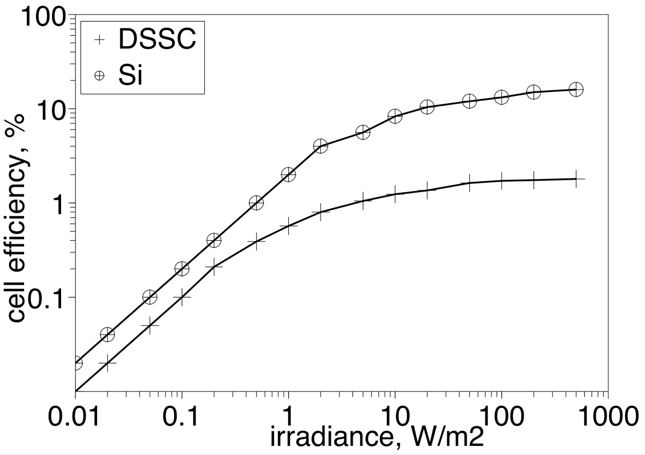


Figure 2. Efficiency of the DSSC and Si cells versus irradiance.

Power efficiency of the solar cell is not the only important property for energy harvesting. The open circuit voltage is also important. Since dark current is the dominant cause of efficiency loss, the open-circuit voltage drops to half of the nominal value at a corresponding power efficiency of 25% of nominal. At low irradiation levels, the DSSC cell [11] produces higher open-circuit voltages (0.6V), enabling the use of less demanding power converters in the sensor node. Below that point, energy harvesting for the solar cells becomes complicated because the supplied voltage becomes insufficient to drive CMOS low-leakage transistors above their threshold voltage. For the DSSC, the voltage-limited harvesting threshold was found to be 0.7 W/m^2 compared to 2.0 W/m^2 for the mono-crystalline Si cell.

III. Measurement results

The raw irradiance data averaged over 10 days was normalized for 1-day intervals (86,400 seconds) with a sampling rate of 1 sample/s (see Figure 3) and binned using thresholds listed in Table 2. The data from Table 2 is plotted on Figure 3. The probability peak at 0.2-0.5 W/m^2 most likely corresponds to indoor conditions, while the smaller peak at 2-5 W/m^2 may be attributed to outdoor light. The averaged time series of the irradiance can be seen in Figure 4. The peak at about 9am corresponds to transit to the workplace. The broad peak about midday corresponds to daylight leaking through the office windows and to lunchtime activity, and the peak around 5pm corresponds to the transit from the workplace to home.

TABLE II. Tabulated irradiance data

Bin #	Parameters		
	Min. irradiance, W/m^2	*Max. irradiance, W/m^2*	*Seconds/day*
1	0	0.01	36776
2	0.01	0.02	54
3	0.02	0.05	113
4	0.05	0.1	286
5	0.1	0.2	5011
6	0.2	0.5	28681
7	0.5	1	6039
8	1	2	3002
9	2	5	3344
10	5	10	1916
11	10	20	819
12	20	50	323
13	50	100	35
14	100	200	0
15	200	500	0

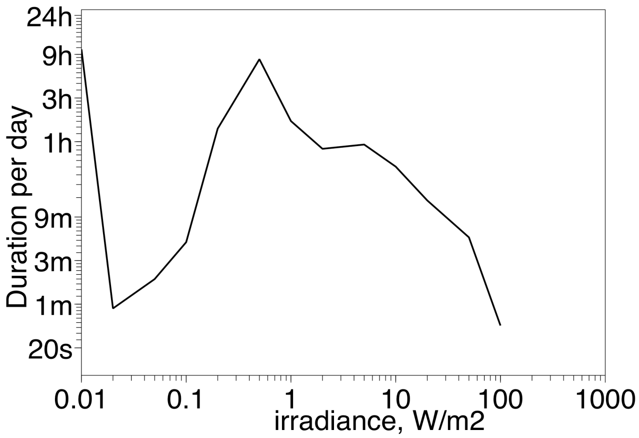


Figure 3. The probability distribuiton of irradiance.

Multiplying the probability distribution from Figure 3 with the geometric average of the power flux (irradiance) in a given bin results in a new metric: the harvestable energy per irradiance bin. Integrating the harvestable energy per irradiance bin results in the cumulative harvestable energy metric. This metric is useful if it is necessary to decide at which irradiance level the solar battery should start harvesting and to determine what losses are expected from not harvesting at lower irradiance levels. The harvestable energy and cumulative harvestable energy plots are shown in Figure 5. From Figure 5 it is shown that 99% of the harvestable radiant flux occurs at irradiances above 0.2 W/m^2 and 50% occurs at irradiancies above 5 W/m^2.

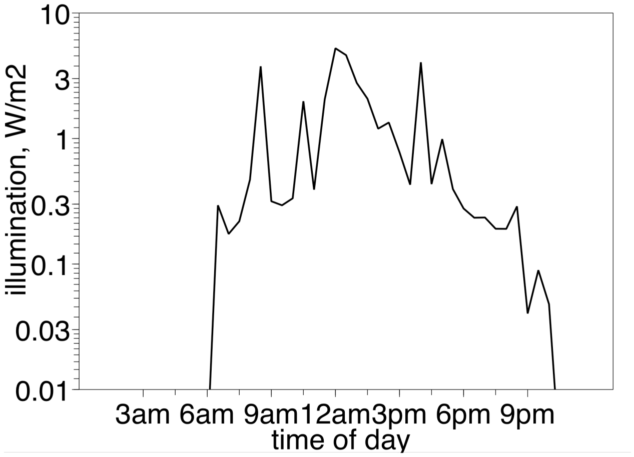


Figure 4. Daily variation of the irradiance.

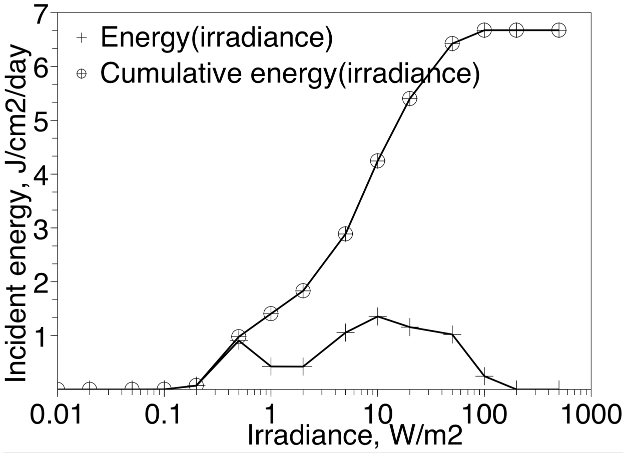


Figure 5. Harvestable energy and cumulative harvestable energy.

Finally, the effect of the reduced efficiency of solar cells at lower irradiance levels should be taken into account. Multiplying solar cell efficiency from Figure 1 by the harvestable energy from Figure 5 results in harvested energy as a function of irradiance. Figure 6 shows that because of the low performance of existing solar cells in low light conditions, most energy is harvested at irradiance 10-50 W/m^2. The DSSC cell used (designated by maker as an indoor energy harvester) performs more consistently at various light levels compared to Si cell, but it still fails to harvest energy efficiently at irradiation range 0.2-0.5 W/m^2 corresponding to typical indoor lighting conditions.

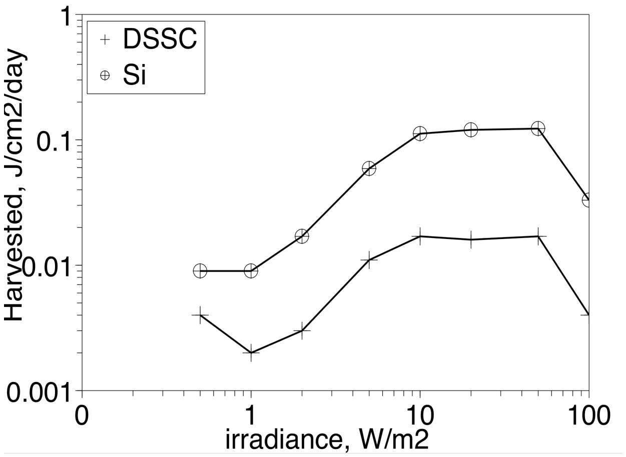


Figure 6. Harvested energy as function of irradiance.

IV. Conclusion

The parameters of harvestable light for wearable sensor nodes worn on the wrist of a typical office worker were measured. Harvestable solar energy for the DSSC and mono-crystalline silicon solar cell was estimated. The results are written in Table 3.

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-207-3

TABLE III. SUMMARY OF THE AMBIENT LIGHT ENERGY HARVESTING FOR THE WEARABLE SENSOR NODE

Parameters	Values
Indoor irradiance, W/m^2	0.2-0.5
Outdoor irradiance, W/m^2	2-5
Average irradiance on wrist, W/m^2	0.78
Energy harvested with DSSC [11], J/day/cm^2	0.074
Energy harvested with 16% efficient Si cell, [1] J/day/cm^2	0.48

For the tested DSSC solar cell, average power available at the solar cell terminals was 7.7uW (for cell area of 9 cm^2). Because the LM03 sensor node does not have any facility for power conversion, estimation of the conversion efficiency and storage efficiency was not attempted. Typical conversion and storage efficiency reported in literature was 11-18% [12].

The authors are continuing to gather data in order to reduce random errors and ultimately to acquire the dependence of the harvested energy on the season of year.

The authors believe that DSSC is more promising for indoor light energy harvesting. The important property of DSSC is the ability to deliver high voltages at low irradiation levels. Harvesting energy at low levels of irradiation reduces the probability of sensor node brownout, because low-light conditions (0.2-0.5 W/m^2) are very common for a typical office employee lifestyle (see Figure 2). Therefore, DSSC-based sensor node may have a smaller, cheaper battery subsystem. To more fully utilize energy from indoor lighting, further effort is necessary to reduce the dark current by 65% to 75% and simultaneously increase the efficiency of the commercially available DSSC to at least 8%.

REFERENCES

[1] E. M. Yeatman, “Advances in Power Sources For Wireless Sensor Nodes,” Proceedings of International Workshop on Wearable and Implantable Body Sensor Networks, Apr. 6-7, 2004, pp. 20-21

[2] V. Joan and C. Kaushik, “Markov Modeling of Energy Harvesting Body Sensor Networks,” 22nd IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), in Toronto, Canada, Sep. 11-14, 2011, pp. 2168-2172

[3] Faruk Yildiz, “Potential Ambient Energy-Harvesting Sources and Techniques,” Journal of Technology Studies, vol. 35, No. 1, Fall 2009, pp. 40-47

[4] Fang Tang and Amine Bermak, “An 84 pW/Frame Per Pixel Current-Mode CMOS Image Sensor With Energy Harvesting Capability,” IEEE Sensors Journal, vol. 12, No.4, April 2012, pp. 720-726

[5] Sujesha Sudevalayam and Purushottam Kulkarni, “Energy Harvesting Sensor Nodes: Survey and Implications,” IEEE Communications Surveys&Tutorials, No.3, Third Quarter 2011, pp. 443-461

[6] S.W. GLunz, J. Dicker, M. Esterie and al., “High-Efficiency Silicon Solar Cells for Low-illumination Applications,” Conference Record of the Twenty-Ninth IEEE Photovoltaic Specialists Conference, in New Orleans, USA, May 19-24, 2002, pp. 450-453

[7] C. O Mathuna, T. O`Donnell, R. V. Martinez-Catala and al., “Energy Harvesting for long-term deployable wireless sensor networks,” Talanta, vol. 75, 2008, pp. 613-623

[8] J. M. Rabaey, M. J. Ammer, da Silva Jr. and al., “PicoRadio supports ad hoc ultra-low power wireless networking,” IEEE Computer Magazine, vol. 33, No. 7, Jul. 2000, pp. 42-48

[9] N. B. Bharatula, S. Ossevoort, M. Stager and G. Troster, “Toward Wearable Autonomous Microsystems,” PERVASIVE 2004 conference, in Vienna, Austria, Apr. 18-23, 2004, pp. 225-237

[10] E. Romero, R. O. Warrington and M. R. Neuman, “Energy scavenging sources for biomedical sensors,” Physiol. Meas., vol. 30, No.9, Aug. 2009, pp. 35-62, doi: 10:1088 /0967-3334/30/9/R01

[11] “Dye Sensitized Indoor Photovoltaic Module,” G0083, Iss01, datashhets from G24i, http://www.g24i.com/filebase/files/57/g24i-indoor-modules-series-2000.pdf, [retrieved: June, 2012]

[12] Ahman Ahnood and Arokia Nathan, “Flat-Panel Compatible Photovoltaic Energy Harvesting System,” IEEE Journal of Display Technology, vol. 8, No.4, April 2012, pp. 204-211

Active Cameras Resources Management Assisted by Wide Field of view Fixed Cameras for the Surveillance of Multiple Moving Targets

Yacine Morsly
Department of Robotic
Ecole Militaire Polytechnique
Algiers, Algeria
ymorsly@yahoo.fr

Mohand Said Djouadi
Department of Robotic
Ecole Militaire Polytechnique
Algiers, Algeria
msdjouadi@gmail.com

Nabil Aouf
Department of Informatics and
SensorsCranfield University
Defence Academy, UK.
n.aouf@cranfield.ac.uk

***Abstract*—In this paper, we propose a novel approach to manage an active resources for a centralized active vision system assisted by a wide field of view fixed cameras (WFOV-FC). Indeed, since the WFOV-FC can provide only large coverage with low resolution, these later are used to generate spatiotemporal observation requests from all detected and tracked target in the surveillance zone. The information gathered will be used to schedule the set of active Pan-Tilt-Zoom cameras (PTZ-AC) in order to collect high-resolution videos suitable for further biometric analysis. Based on the output of this biometric analysis, the same used set of active cameras is requested to maximize at the same time the coverage with close-up views of every target identified as a threat. We formulate PTZ multi-cameras assignment and handoff as a planning problem whose solution achieves optimal cameras assignment in real time. Simulation results, show the efficiency of the proposed policy in satisfying both objectives at the same time**

Keywords-Multi-cameras systems; active and fixed cameras; assignment; online scheduling

I. Introduction

There is an ever increasing demand for security monitoring systems in the modern world. Visual surveillance is one of the most promising areas in security monitoring for several reasons. It is easy to install, easy to repair, and the initial setup cost is inexpensive when compared with other sensor based monitoring systems, such as audio sensors, motion detection systems, thermal sensors, etc. [1].

Video surveillance systems are installed in locations ranging from multinational banking organizations to public institutions to small local stores, and there is a similar disparity in the level of sophistication of installed systems. The use to which these systems are put also varies widely with the intentions of the operators and the budget available for the implementation of the system.

Earlier works in the field of cameras and videos technologies have made it possible to network numerous video cameras together by an operator in order to provide visual coverage of small and medium spaces such as banks and shops. However, as the size of the multi-cameras system grows and the level of activity in the public space increases, it becomes infeasible for human operators to monitor the multiple video streams and identify all events of possible interest, or even to control individual cameras in performing advanced surveillance tasks, such as zooming on a moving subject of interest to acquire one or more facial snapshots. Moreover, the cost of employment of a human operator outpaces the cost of installing and maintaining the multi-camera systems. Consequently, a timely challenge for computer vision researchers is to design multi-cameras systems capable of performing visual surveillance tasks automatically or at least with minimal human intervention. We regard the design of an autonomous multi-cameras system as a problem of resource allocation and scheduling, where the cameras are treated as resources required to complete the desired sensing tasks.

Autonomous multi-cameras systems using only WFOV-FC can provide large, low resolution coverage of the scene. However, recognition and identification of targets usually require close-up views at high resolution which need PTZ-AC. The resulting proposed autonomous multi-cameras system is based on a set of WFOV-FC's and a set of PTZ-AC's as illustrated in Figure 1. The major challenge in this work is the control and scheduling of the set of PTZ-AC so that satisfying the tradeoff between three competing objectives:

- Capture high quality video for as many as possible, preferably all, of the targets in the scene
- Observe each target for as long or as many times as possible, since the chances of identifying and classifying a target improve with the amount of data collected about that target.
- Maximize the coverage time of targets identified as a threats during their stay in the surveillance zone.

Not considering one of the three objectives will conduct to the situation where each camera follows a single target for their entire stay in the scene, ignoring all other pedestrians. The second situation is that a camera briefly observes every target in turn and repeatedly, thus spending most of the time transitioning between different pan, tilt, and zoom settings. The third one leads to ignore appeared threats.

This paper is organized as follows. Section II gives comprehensive background of the current and emerging approaches for camera selection and handoff. This is followed by presenting the adopted system architecture for an accurate autonomous video surveillance. In Section IV, we present the formulation of the challenge targeted as a machine scheduling problem. We next propose our scheduling policy in Section V. Finally, we describe simulation setup and results in Section VI and conclude the paper in Section VII.

II. Related Work

Previous work on multi-camera systems has dealt with issues related to low and medium-level computer vision, namely identification [2, 3], recognition [4, 5], detection and tracking of moving objects [6-13]. The importance of accurate detection, tracking, and data association is obvious, since tracking information is needed as the initial stage for controlling one or more PTZ cameras to acquire high-resolution imagery.

However, in addition to detection and tracking, the acquisition of high-quality imagery, particularly for biometrics purposes, requires accurate calibration between the fixed and PTZ cameras in order to focus attention on interesting events that occur in the scene. The control or the schedule of active cameras set when there are more objects to be monitored in the scene than the active cameras is also a challenge for many researchers. Some of themes employ a WFOV-FC to control an active tilt-zoom camera. This configuration is often termed in the literature as master–slave. Many researchers use a master–slave camera configuration with two or more [19] cameras. In particular, most of the methods strongly rely on a direct camera calibration step. Basically these approaches are not autonomous since they need a human to deal with calibration marks. Nevertheless, few exceptions are discussed, where in order to track targets across a fixed and a PTZ camera, they used an affine transformation between consecutive pair of frames for stabilizing moving camera sequences, and an homography transformation for registering the moving and stationary cameras with the assumption that the scene is planar.

A camera scheduling algorithm [14], would typically utilize tracking information, provided by one or more fixed cameras performing detection and tracking [16, 17], for computing a schedule that controls the assignments of targets to PTZ cameras over time. Each PTZ camera would then servo, based on calibration data, to aim itself at different targets in a timely fashion as specified by the schedule.

A similar approach involving calibrated static and pan/tilt cameras is presented in [9]. Data from multiple static cameras is fused to estimate the 3D location of the pedestrian. An active camera uses calibration information to bring the target into the center of its view. After initial repositioning, the active camera autonomously tracks the target [6, 7], thereby avoiding the communication overhead associated with master-slave configurations. The active camera periodically sends its pan/tilt settings to the static cameras. The static cameras can use this information to decide whether or not the active camera is tracking the correct target. If it is not, the active camera is repositioned.

Other authors, such Lim et al. [20], have proposed a number of different camera scheduling algorithms designed for different application goals, where they include, for example a scheme for scheduling available cameras in a task-dependent fashion , static and non static priority policies [15].

The work presented here, differs from the previous existing scheduling multi-cameras works in the following points:

It can handle several PTZ-AC's

The different PTZ-AC's are modeled as autonomous agents that are not driven by the WFOV-FC's.

The scheduling strategy supports both high quality videos recording and coverage insurance of targets and threats.

III. System Architecture

Figure 1 shows the architecture of the system. The system considered consists of "$\mathrm{N_a}$"($\mathrm{N_a} > 1$) PTZ-AC's and WFOV-FC's. The WFOV-FC's detect and label all moving objects in the scene. The states of the objects (e.g., size, position and velocity) in the 2D image space are tracked and predicted. Based on the prediction, the different observation requests are generated. Then the request assignment process assigns a subset of the targets/requests to each PTZ-AC by computing the relevance of the different PTZ-AC's to the observed targets in the surveillance zone. Each PTZ-AC camera tracks the objects assigned to it by selecting the PTZ-AC parameter settings that best satisfy these requests to capture high resolution images/videos of the targets.

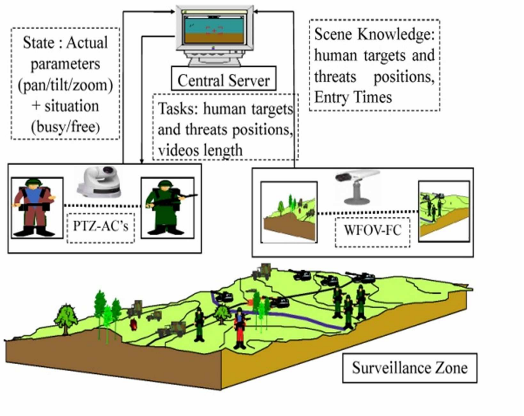

State : Actual parameters (pan/tilt/zoom) + situation (busy/free)
Central Server
Tasks: human targets and threats positions, videos length
Scene Knowledge: human targets and threats positions, Entry Times
PTZ-AC's
WFOV-FC
Surveillance Zone

Figure 1. System Overview

IV. Problem Statement

We consider a multi-cameras system including "N_f"calibrated WFOV-FC's and "N_a" PTZ-AC's.

Let $O = \{o_i \backslash i = 1,2, \dots\}$denote the set of targets observed at a given time by the multi-cameras system. At that time, the state of a target is given by $o_i^t = (x_i^t, v_i^t)$ where "x_i" and "v_i" represent the position and velocity, respectively, of observed target *i*.

Let each PTZ-AC be described by a tuple $(P, \alpha_{min}, \alpha_{max}, \beta_{min}, \beta_{max}, Z_{min}, Z_{max})$. We assume therefore that the "3D position P of each PTZ-AC is known a priori. $[\alpha_{min}, \alpha_{max}]$, $[\beta_{min}, \beta_{max}]$ and $[Z_{min}, Z_{max}]$ represent the pan tilt and zoom limits, respectively, for each PTZ-AC.

The problem is to find a schedule on "$\mathrm{N_a}$" PTZ-AC that minimizes the total unit penalty when target "I" with deadlines "$\mathrm{d_i}$" are released at time "$\mathrm{r_i}$" The targets require arbitrary processing times and pre-emption (pmtn) is allowed. So, minimizing the total unit penalty is akin to maximizing the number of targets successfully recorded prior to their deadlines.

We can describe the camera scheduling problem proposed as:

$$E^{*} = argmax\ q(E_i^t) \qquad (1)$$

$$E_i^t \in E_a$$

$$E_a = \{E_i^t / t \in [t_{current_{time}}, t_p], i \in [1, N_a]\} \qquad (2)$$

where E denote a feasible event defined as follows:

$$E = R(c_i^{PTZ}, o_i) \cup R(c_i^{PTZ}, ot_i) \qquad (3)$$

$R(c_i^{PTZ}, o_i) -$ The recording of human target by a PTZ-AC.

$R(c_i^{PTZ}, ot_i) -$ The recording of human target identified as a threat by a PTZ-AC.

$t_p -$ Preset time that indicates the number of predicted plans depending on the predicted states of humans targets observed.

The complexity of problem (1) is NP-hard [18]. Hence, we resort to a greedy algorithm for scheduling cameras to observe targets.

The obvious non-clairvoyant online algorithms are Round Robin (RR) and Shortest Elapsed Time First (SETF). The idea of the RR is to devote identical processing resources to all jobs, whereas SETF devotes all resources to the job that has been processed the least. As SETF is known to perform poorly when jobs are not fully parallelizable [19], we used the weighted RR.

The policy of using a weighted RR scheduling scheme is to assign jobs to multiple processors with different load capacities. Each processor is assigned a weight indicating its processing capacity and more jobs are assigned to the processors with higher weights.

We model each PTZ camera as a processor whose weights, which quantify the suitability of a camera with respect to observing a target, are adjusted dynamically.

These weights are assimilated to a combination of several quality measures.

The computation of the relevance of a PTZ-AC to the task of recording close-up videos of selected targets for further identification and/or classification process, encodes an intuitive observation which is formalized by describing the relevance of a PTZ-AC to the task of observing a target in terms of quality factor"q". omitting superscripts t, i and E , the global quality is expressed as follows:

$$\mathrm{q} = \begin{cases} 1 & \text{if } c_i^{PTZ}\ is\ idle \\ \mathrm{q}_{\alpha\beta\mathrm{Z}} \cdot \mathrm{q}_{\mathrm{o}} \cdot \mathrm{q}_{\mathrm{d}} \cdot \mathrm{q}_{\sigma} \cdot \mathrm{q}_{\mathrm{c}} \end{cases} \qquad (4)$$

where the different sub qualities are defined in the following subsections:

A) PTZ limits "$q_{\alpha\beta Z}$" *:*

The turn and zoom limits of cameras should be taken into account when assigning a camera to observe a target. A camera that has more leeway to turn and zoom may be able to follow a target for a longer period of time. The mechanical limitation for each PTZ camera on its Pan Tilt Zoom parameter range is defined by: $(\alpha_{min}, \alpha_{max}, \beta_{min}, \beta_{max} Z_{min}, Z_{max})$.

$$\mathrm{q}_{\alpha\beta\mathrm{Z}} = \exp\left(-(\alpha - \widehat{\alpha})^2 - \left(\beta - \widehat{\beta}\right)^2 - \left(\mathrm{Z} - \widehat{\mathrm{Z}}\right)^2\right) \qquad (5)$$

$$\widehat{\alpha} = \frac{(\alpha_{min} + \alpha_{max})}{2} \qquad (6)$$

$$\widehat{\beta} = \frac{(\beta_{min} + \beta_{max})}{2} \qquad (7)$$

where: "α" and "β" are respectively, the pans, tilts angles corresponding to the location of the target.

"Z", is the actual zoom setting of a given camera.

"$\widehat{\mathrm{Z}}$", is the desired zoom to record close-up videos.

B) Observational range "q_o"*:*

It reflects the observational constraints of a camera. It is set to 0 when the human target is outside the observational range of a camera; otherwise, it is set to 1.

$$\mathrm{q_o} = \begin{cases} 1 & \text{if } \alpha \in [\alpha_{\min}, \alpha_{\max}] \text{ and} \\ & \beta \in [\beta_{\min}, \beta_{\max}] \text{ and} \\ & \mathrm{Z} \in [\mathrm{Z}_{\min}, \mathrm{Z}_{\max}] \\ 0 & \end{cases} \qquad (8)$$

where: $[\alpha_{\min}, \alpha_{\max}]$, $[\beta_{\min}, \beta_{\max}]$ and $[\mathrm{Z}_{\min}, \mathrm{Z}_{\max}]$ are the external vertical, horizontal rotations, and zoom factor.

C) Target-Camera Distance "q_d"

Tracking becomes harder as camera-to-target distance grows. The quality of captured imageries generally degrades as the distance"$\mathrm{d_{tc}}$", between the target and the camera increases or decreases consequentially. Hence, this quality measure is only based on "$\mathrm{d_{tc}}$".

$$\mathrm{q_d} = \exp - \left(\left(\mathrm{d}_{tc} - \widehat{\mathrm{d}}\right)^{\wedge 2}\right) \qquad (9)$$

where $\widehat{\mathrm{d}}$ is the desired distance of a target to a camera that allows recording close-up videos.

D) View Angle "q_σ"

It is more desirable to select a camera that has frontal view of a target, in order to record high quality videos that will serve for identification purposes. This later is defined as it is the angle between the velocity vector of the target and the optical centre of a camera.

$$\mathrm{q}_\sigma = \exp(-\sigma) \qquad (10)$$

where σ is defined as the angle between the velocity vector of a target and the line that join the position of that target and a selected camera.

E) Handoff candidate"q_c"

Handoff candidate quality gives preference to handoff candidates in the vicinity of the camera currently observing the target. The idea is that nearby cameras have similar viewpoints, making appearance based target signature more relevant for the candidate camera.

$$\mathrm{q_c} = \exp(-\delta) \qquad (11)$$

where δ is the angle between the fixation vector of camera c_i^{PTZ} and the fixation vector of the camera currently observing the target.

V. Multi-Cameras Scheduling Algorithm

In this section, we describe the scheduling policy for scheduling a set of PTZ-AC's in order to achieve the above cited challenge. Finding an optimal event that fits with the movement horizon of the targets/ threats, while maximizing the objective function (1) is a combinatorial problem. Our policy is based on some modifications in the well known Round Robin algorithm which yields to a novel heuristic able to satisfy the three points of our challenge. Algorithm 1 outlines our policy strategy:

Data: set of PTZ-AC's currently assigned to the different targets"$\mathrm{V_{Busycam}}$" . Set of PTZ-AC's cameras that are currently available"$\mathrm{V_{Freecam}}$". Set of targets that are currently not assigned a PTZ-AC "$\mathrm{V_{Unsched-o}}$". Set of targets that are currently being followed by PTZ-AC's "$\mathrm{V_{sched\_o}}$". Set of threats that are currently not assigned a PTZ-AC "$\mathrm{V_{Unsched-ot}}$". Set of threats that are currently being followed by PTZ-AC's "$\mathrm{V_{sched\_ot}}$".

```
Begin:
  Set $V_{\text{Busycam}}$, $V_{\text{Unsched\_o}}$, $V_{sched\_o}$, $V_{\text{Unsched\_ot}}$ and $V_{sched\_ot}$ to $\{\emptyset\}$.
  $V_{\text{Freecam}} = \{c_i^{PTZ}/\ i = 1, \dots, N_a\}$
  For $t$= 1: $t_p$: ….. do
    Remove targets and threats that appears to have left the scene from $V_{\text{Unsched\_o}}$, $V_{sched\_o}$, $V_{\text{Unschedot}}$ and $V_{Lsched\_ot}$. Move the corresponding cameras from $V_{\text{Busycam}}$ to $V_{\text{Freecam}}$
    For all New arrivals "$o_i$" do
      Set "$t_{s\_o_i} = 0$" (the timestamp of"$o_i$"). Set "$t_{rc\_o_i} = 0$" (times-recorded count on"$o_i$"). Add "$o_i$" to $V_{Lsched\_o}$
    End for
    For all Cameras "$c_i^{PTZ}$" in the $V_{\text{Busycam}}$ do
      if $t_{s\_o_i}$ by a "$c_i^{PTZ}$" is equal to $t_{pr\acute{e}emption}$ then Set "$t_{s\_o_i} = 0$". Increment "$t_{rc\_o_i}$". Move "$c_i^{PTZ}$" from $V_{\text{Busycam}}$to$V_{\text{Freecam}}$. Move "$o_i$" from $V_{sched\_o}$ to $V_{\text{Unsched\_o}}$
      else if("$t_{s\_o_i}$" by a Camera "$c_i^{PTZ}$" is>= $t_{pr\acute{e}emption}$ and ($V_{\text{Unsched\_o}} \neq \{\emptyset\}$ or $V_{\text{Unsched\_ot}} \neq \{\emptyset\}$) and (Camera "$c_i^{PTZ}$" is relevant to at least one of the targets in $V_{\text{Unsched\_o}}$ or threats in $V_{\text{Unsched\_ot}}$) or ("$o_i$" times-recorded count ≥1 and a target "$o_j, j \neq i$" in $V_{\text{Unsched\_o}}$ has times-recorded count equal to 0 and Camera "$c_i^{PTZ}$" is relevant to "$o_j$" ) or threat "$ot_j, j \neq i$" in $V_{\text{Unsched\_ot}}$ has times-recorded count equal to 0 and Camera "$c_i^{PTZ}$" is relevant to "$ot_j$" )then
        (Set "$t_{s\_o_i} = 0$" and Move "$c_i^{PTZ}$" from$V_{\text{Busycam}}$ to $V_{\text{Freecam}}$ and Move "$o_i$" from $V_{\text{sched\_o}}$to $V_{\text{Unsched\_o}}$ ) or (Move "$c_i^{PTZ}$" from $V_{\text{Busycam}}$ to $V_{\text{Freecam}}$ and Move "$ot_i$" from $V_{\text{sched\_ot}}$to $V_{\text{Unsched\_ot}}$ )
    End if, End for
    For all Cameras "$c_i^{PTZ}$" in $V_{\text{Freecam}}$ do
      Compute $V_{\text{relevant\_o}}$ which consists of the targets in $V_{\text{Unsched\_o}}$ that are relevant to "$c_i^{PTZ}$".
      Compute $V_{\text{relevant}_{\text{ot}}}$ which consists of the targets in $V_{\text{Unsched}_{\text{ot}}}$ that are relevant to "$c_i^{PTZ}$"
      If $V_{\text{relevant\_ot}} = \{\emptyset\}$ then
        If $V_{\text{relevant\_o}} = \{\emptyset\}$ then Continue
        Elseif $V_{\text{relevant\_ot}} = \{\emptyset\}$ and $V_{\text{relevant\_o}} \neq \{\emptyset\}$ then Pick target "$o_i$" from $V_{\text{relevant\_o}}$ with the highest probability of threat. Assign "$c_i^{PTZ}$" to "$o_i$". Move "$c_i^{PTZ}$" from $V_{\text{Freecam}}$ to $V_{\text{Busycam}}$. Move "$o_i$" from $V_{\text{Unsched}}$ to $V_{sched}$
        Elseif ($V_{\text{relevant\_ot}} \neq \{\emptyset\}$ and $V_{\text{relevant\_o}} = \{\emptyset\}$) or ($V_{\text{relevant\_ot}} = \{\emptyset\}$ and $V_{\text{relevant\_o}} = \{\emptyset\}$ ) then Pick target "$ot_i$" from $V_{\text{relevant\_ot}}$ with the highest probability of threat. Assign "$c_i^{PTZ}$" to "$ot_i$". Move "$c_i^{PTZ}$" from $V_{\text{Freecam}}$ to $V_{\text{Busycam}}$.
End if, End for, End for
```

Algorithm 1. Sheduling policy

VI. Simulation Results

We simulate a monitoring area with up to 15 autonomous targets that enter, travel for free inside, and leave the monitoring area of their own volition. We tested the scheduling strategy in various scenarios using from 1 to 5 PTZ active cameras.

Each target spends anywhere from 40 to 150 seconds in the monitoring space. The processing time judged satisfactory to capture sufficient frame for identification and classification purpose is set to 30 seconds, while the preemption cut-off time is set to 5 seconds. The targets are assumed to enter the monitoring area randomly. The simulation time is set to 180 seconds.

As expected, we can see from Figure 2 that the probability of correct recording depends to the ratio between the number of targets and available cameras. However, we judge that the results are acceptable since the majority of obtained values are superior to 0.5. This is due in part to the use of the quality measure which exhibits high success rate for recording close up video and lower average lead time required by a PTZ-AC to fixate on a target and initiate video recording.

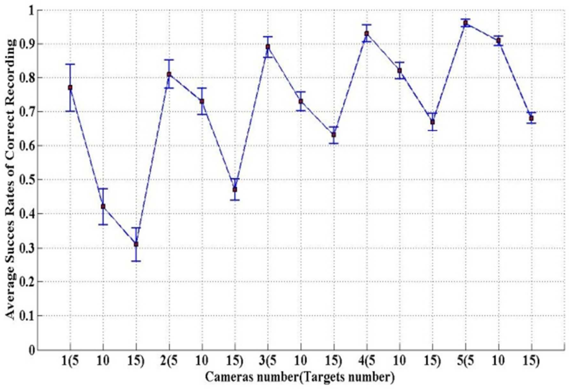


Figure 2. Average succes rate of correct recording

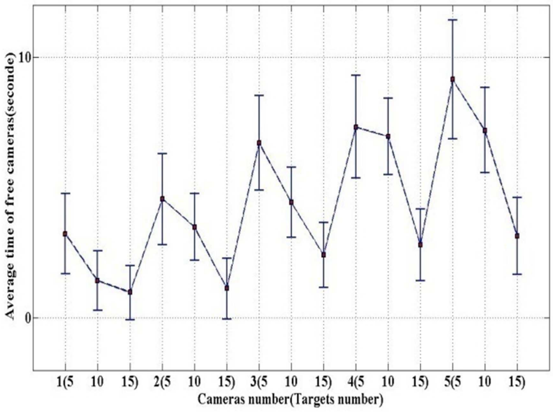


Figure 3. Average time of free cameras

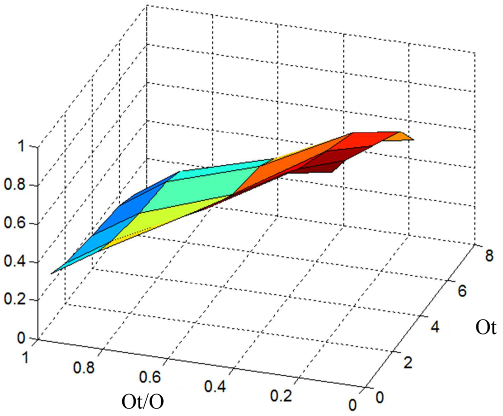


Figure 4. Average succes rate of total threats covrage versus number of targts

Figure 3 shows that average time of free cameras depends also to the above cited ratio. We can remark that the results are very encouraging since the highest obtained time which corresponds to the worst result; present an approximate rate of 0.04 in the total considered simulation time.

Figure 4 represents the success rate of assuring total coverage of a number of threats (up to 7) while using 5 cameras. We can remark that this rate decrease as the number of total target increase. Nevertheless, this result confirms the capacity of the elaborated policy to target several objectives at the same time and return an optimal scheduling solution at every time step.

VII. CONCLUSION AND FUTURE WORK

We have presented a novel approach to control multiple PTZ-AC's assisted by a set of WFOV-FC in order to satisfy three main objectives in real time. The main novelty of our approach lies in capturing high quality video for as many as possible of the targets in the scene, while observing each target for as long or as many times as possible and at the same time maximizing the coverage time of targets identified as a threats during their stay in the surveillance zone. The problem was solved by using a probabilistic objective function that encapsulates a set of quality measures. The reported simulations demonstrate the effectiveness of the proposed policy in satisfying the above objectives in the same time. We are interested in conducting more detailed quantitative performance evaluation in the future by validating imageries captured online with biometrics tasks such as face detection or recognition that can be conducted offline.

REFERENCES

[1] N. Takemura and H. Ishiguro, "Multi camera vision for surveillance," Handbook of Ambient Intelligence and Smart Environments, Part II, pp. 149-169, DOI: 10.1007/978-0-387-93808-0_6, Springer Science+Business Media, LLC 2010.

[2] C.-Y. Lin and al., "Multi-camera invariant appearance modeling for non-rigid object identification in a real-time environment," Journal of Vision Communication and Image Representation, In Press, Corrected Proof, DOI: 10.1016/ j.jvcir, 2012.

[3] O. Hamdoun and al., "Person Re-identification In Multi-camera System By Signature Based On Interest Point Descriptors Collected On Short Video Sequences," 2nd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC-08), Stanford, United States, pp. 162-168, DOI: 10.1109/ ICDS.2008.4635689, 2008.

[4] J. Hensler, K. Denker, M. Franz, and G. Umlauf, "Hybrid Face Recognition Based on Real-Time Multi-camera Stereo-Matching," ISVC, Part II, LNCS 6939, pp. 158–167, Springer-Verlag Berlin Heidelberg, 2011.

[5] L. Fiore and al., "Multi-Camera Human Activity Monitoring," Intel Robot Syst. Journal, vol. 52, pp. 5–43, DOI: 10.1007/s10846-007-9201-6, Springer Science + Business Media B.V., 2008.

[6] R. Collins, O. Amidi, and T. Kanade, "An active camera system for acquiring multiview video," in Proc. Int. Conf. Image Processing (ICIP), pp. 517–520, Rochester, N.Y., Sept. 2002.

[7] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. "Tracking across multiple cameras with disjoint views," in Proc. IEEE International Conference on Computer Vision, pp. 952-957, 2003.

[8] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in Proc. of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 00), vol. 2, pp. 142–151, 2000.

[9] S. Khan and M. Shah, "Consistent labeling of tracked objects in multiple cameras with overlapping fields of view," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, pp. 1355–1360, 2003.

[10] J. Kang, I. Cohen, and G. Medioni, "Multi-views tracking within and across uncalibrated camera streams," in Proc. First ACM SIGMM International Workshop on Video Surveillance, pp. 21–33, ACM Press, New York, NY, 2003.

[11] N. T. Siebel, "Designing and Implementing People Tracking Applications for Automated Visual Surveillance," PhD thesis, Dept. of Computer Science, university of Reading, UK, 2003.

[12] T. Gandhi and M. Trivedi, "Calibration of a reconfigurable array of omnidirectional cameras using a moving person," in Proc. 2nd

ACM International Workshop on Video Surveillance and Sensor Networks, pp. 12–19, ACM Press, New York, NY, 2004.

[13] D. Devarajan , R. J. Radke, and H. Chung, " Distributed metric calibration of ad-hoc camera networks," ACM Transactions on Sensor Networks Journal, vol. 2, pp. 380-403, 2006.

[14] J. Costello, C. Diehl, A. Banerjee, and H. Fisher, "Scheduling an active camera to observe people," in the proceeding of the ACM 2nd international workshop on Video surveillance & sensor networks, ISBN/ 1-58113-934-9, pp. 39-45, 2004.

[15] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Markl, S. Pankanti, A. Senior, C. Fe Shu, and Y.L. Tian, "Smart video surveillance," IEEE Signal Processing Magazine, vol. 22, pp. 38–41, 2005.

[16] T. Zhao and R. Nevatia, "Tracking multiple humans in complex situations," IEEE Transaction on Pattern Analysis and Machine Intelligence Journal, vol. 26, pp. 1208-1221, 2004.

[17] T. Zhao and R. Nevatia, "Tracking multiple humans in crowede environment" in Proc. of the IEEE CVPR, vol. 2, pp. 406-413, 2004.

[18] N. Krahnstoever, T. Yu, S. Lim, K. Patwardhan, and P. Tu, "Collaborative Real-Time Control of Active Cameras in Large Scale Surveillance Systems," Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications M2SFA2 2008, Marseille, France, 2008.

[19] F. Qureshi and D. Terzopoulos, "Surveillance camera scheduling. "A virtual vision approach," ACM Multimedia Systems Journal, Special Issue on Multimedia Surveillance Systems, vol. 12, pp. 269–283, 2006.

[20] S.N. Lim, L.S. Davis, and A. Mittal, "Constructing task visibility intervals for a surveillance system," ACM Multimedia Systems Journal, Special Issue on Multimedia Surveillance Systems 12, 2006.

Reducing Energy Consumption in a Sheep Tracking Network Using a Cluster-based Approach

Ragnar Stølsmark
Institute of Computer Science and Electrical Engineering
University of Stavanger
Stavanger, Norway
rstolsmark@live.no

Erlend Tøssebro
Institute of Computer Science and Electrical Engineering
University of Stavanger
Stavanger, Norway
erlend.tossebro@uis.no

***Abstract*—Sheep tracking ease the work of the farmer when retrieving the sheep. The current commercial sheep tracking solution is popular but not energy-efficient. It only uses GSM and GPS and has no interaction between the sheep. Sheep often walk in clusters. If a farmer knows the location of all clusters and also which sheep are in each cluster, he does not need to know the exact location of each sheep. By using a new cluster-based localization and data retrieval approach this paper show, through energy measurements and simulations, that it could be possible to reduce the average energy consumption by more than 50 % in flocks were the sheep walks in large clusters. The reduced energy consumption could be used to either increase the update frequency or to reduce the battery capacity. The cluster–based approach can also eliminate the need for GSM and GPS modules on part of the sheep nodes, thereby making the system more affordable for farmers. The reduced energy consumption and cost makes the solution described in this paper better than the currently available commercial one.**

Keywords-Wireless Sensor Networks; Animal Tracking; Energy Consumption.

I. INTRODUCTION

Sheep farmers in Norway send their sheep to graze on the fertile mountain grass during the summer. This is important since the grass on the farm then is allowed to grow freely and can be harvested and used as food for the sheep during the winter. One of the big problems with this custom is to locate the sheep when the summer is over. The common method for sheep localization has been that the farmer searches for the sheep manually. This search will typically start in the area where the farmer thinks the sheep will be located and expand outwards to the less likely areas. The search continues until all sheep are found, or the farmer grows tired of looking for them. This process can take a week or more of walking in often challenging terrain.

In the last few years there have come a solution to this inefficient retrieval method. Telespor [1] is a commercially available system based on the electronic shepherd research project [2]. It tracks the sheep using GPS and sends the position of the sheep back to the farmer using the GSM network. However, it is far from perfect. The biggest problem is cost. Each unit cost approximately € 200, which is too expensive for most farmers, at least if they want to equip their entire flock with these devices. Another limiting factor is the battery capacity. The batteries should not have to be replaced during the season which last around 100 days. Since there is a size and cost limitation on the batteries, it is important to reduce the energy consumption of each update to allow for more position updates and thereby increase the utility of the system.

This paper looks at the possibility of mitigating both the cost and energy consumption problem by taking advantage of the fact that sheep tend to move in clusters. In the current Telespor system every sheep find their own position using GPS. The farmer however, only needs to know which sheep are in each cluster and the position of one of the sheep in every cluster. Knowing this, it is possible to construct an algorithm where only the sheep with the highest battery level in a cluster use the GPS to find its location. The rest of the cluster only needs to report which cluster they belong to. This will reduce the energy consumption since GPS localization requires fairly high amounts of energy.

Using a cluster-based approach it is also achievable to have some of the sheep only carry radio transceivers, not GPS receivers and GSM transceivers,. It will only be possible to locate these cheaper equipped sheep when they are in a cluster with at least one sheep with a GPS receiver. It can be argued that this information is of limited value when retrieving the sheep. The farmer has to retrieve every cluster anyway and will therefore find these extra sheep. It can however be useful during the season for a farmer to know if a sheep is doing well. If for instance a sheep that has been following a cluster for some time suddenly disappears, it might be a signal that it is injured or dead.

Two systems have been developed and tested during this research project. The difference between the two systems lies in how the sheep transport data back to the server. In the system called Distributed GSM, every sheep reports their position. In the Centralized GSM approach, only the sheep responsible for finding the GPS position of the cluster, report its position along with the information of which sheep are in the cluster.

This paper is structured as follows: Section II contains related work. System design and equipment is covered in

Section III. Section IV contains the results found by analyzing the different systems, measuring the energy consumption and performing simulations. Section V concludes the paper and looks at future work.

II. Related Work

The Telespor [1] system described in [2] is used as a base line reference throughout this paper and is therefore a closely related work. Telespor is described in Section III as a benchmark version was made to work on the same equipment used in the rest of this paper. This benchmark version was built on the principles described in the electronic shepherd paper.

Stølsmark and Tøssebro looked at the possibility of localization via trilateration in a sheep tracking network [6]. Some of the sheep would be equipped with GPS and the rest of the flock would find their position by using GPS sheep as beacons for RSSI-based trilateration. They found that the position estimates were not accurate enough to be useful. This was mainly due to the RSSI being highly dependent on non-distance related factors, such as weather and topography.

Huircán et al [7] tried to locate cattle in a field using RSSI. They were able to achieve a high level of accuracy but only by having a dense beacon placement, with around 80 m between each beacon. This makes such a localization scheme unsuitable for a large area where sheep typically graze.

In the Zebranet project [8, 9] Juang et al. tried to track zebras in an area without GSM coverage. The zebras would store logs of their own positions and exchange these logs when they met other zebras. The logs could then be downloaded by the data collecting scientists driving around the savannah with radio transceivers.

The WildSensing project [10] tries to monitor badgers in the wild in an energy-efficient manner. Dyo et al. equipped the badgers with RFID tags and set up RFID readers at strategic locations where the badgers would likely be. Since the badgers sleep during the day, the readers would only be active at night. Markham et al. even tried monitoring the badgers in their underground burrows using magnetic fields [11].

Polastre et al. monitored seabirds on the remote Great Duck Island [12] using a tiered architecture to save energy and money. This meant that the wireless sensor network nodes would send their data to a base station on the island which would provide WAN connectivity to send data back to the scientists.

Surveys on different localization methods can be found as part of the extensive wireless sensor network survey performed Yick et al. [13] and in the survey by Akyildiz et al. [14].

Much research effort has been put into localization, but different applications require different solutions. A solution that works for badgers is not necessarily suitable for sheep. In the case of sheep, it is possible to design a system that benefits from the fact that lambs follow their mothers. The application-specific requirements will make it difficult to design one tracking system that can handle every situation. It could be possible to identify a few different scenarios that would fit most applications and design solutions to them. An example of such a scenario could be tracking a large animal in an area without GSM coverage. Identifying the suitable scenarios and creating solutions to them is still an open problem.

III. System Design and equipment

A sheep tracking system has to be able to provide the farmer with the position of his sheep, at regular time intervals, throughout an entire season, without any manual local intervention. Any method requiring manual synchronization or battery replacement is therefore not an option. It should also be possible to change the frequency of position updates during the season since it is more important to have frequent updates when collecting the sheep.

This paper looks closer at three possible sheep tracking solutions. They all use GSM for data transport between the sheep and the farmer. They also use GPS as part of the localization process. The difference between them lies in the amount of interaction between the sheep in a flock.

A. *Telespor*

Telespor is the system farmers are currently using to locate their sheep. It is a simple but working solution with no interaction between the different sheep in a flock. At regular intervals the sheep calculates their location using GPS and sends this location to a server via the GSM network. The benefit of this approach is that since it does not communicate between the sheep, it is not necessary to equip the sheep with an additional transceiver and antenna. One of the problems the farmers have reported with the Telespor system is the poor performance in areas with little GSM coverage [3]. Sometimes, it could take weeks between sheep position updates. The results presented in [4] show that adding sheep to sheep communication equipment is a good solution to the coverage problem. Therefore it is reasonable to argue that sheep to sheep transceiver equipment should be added anyway, and thereby making it possible to implement the two other systems studied in this paper.

Telespor is used as a baseline reference when studying the performance of the different systems. It is especially interesting to look at how much energy consumption can be reduced by using a cluster-based positioning approach, to see if it is worth the effort and added complexity. Algorithm 1 describes the Telespor solution.

```
On init(){
        SleepUntil(first update);
}
On update(){
        pos = GPS.getPosition();
        GSM.SendToServer(pos, id);
        SleepUntil(next update);
}
```

Algorithm 1. The Telespor algorithm.

B. *Distributed GSM*

The Distributed GSM solution is identical to the Telespor system in the way that each sheep send their position back to the server. The positioning differs due to the cluster-based approach. The sheep are synchronized and wakes up from sleep at regular intervals. Upon wake up, every sheep will calculate a delay based on the amount of energy left in their battery. In experiments 20 seconds was found to be a sufficient maximum value for this delay. If a leader message is received during this time the receiving sheep will become a follower of the leader sheep. If a sheep does not receive any leader message before the delay expire, it will become a leader sheep itself and send out a leader message. A leader sheep and all of its followers are considered to be one cluster. The leader will locate its position using GPS and broadcast this position to the followers. The followers, as well as the leader, will report the same position back to the server via the GSM network. Distributed GSM require every sheep to individually report its own position, no in-network aggregation or collection is performed in this solution. The Distributed GSM algorithm can be seen in Algorithm 2.

```
On init(){
        Clock.Synchronize();
        SleepUntil(first update);
}
On update(){
        delay = CalculateDelay(battery.status);
        Wait(delay);
        if(HasReceived(leaderMessage)){
           //become follower
           leader = leaderMessage.leader;
        }else{
           //become leader
           leader = this;
           leaderMessage.leader = this;
           Broadcast(leaderMessage)
           pos = GPS.getPosition();
           Broadcast(sheepid, pos);
        }
        GSM.SendToServer(leader.id, id);
        SleepUntil(next update);
}
```

Algorithm 2. The Distributed GSM algorithm.

C. *Centralized GSM*

Centralized GSM is identical to Distributed GSM when it comes to leader choice and sheep localization. However, instead of each sheep individually reporting its position to a server via the GSM network, the follower sheep send a notification to the leader. The leader sends its position along with the list of followers to the server. This has the potential of saving energy since the energy consumption of each follower is reduced at the expense of increased energy consumption for the leader. The increased consumption at the leader is due to more communication and an extra listening period when waiting for the messages from the followers. It is also possible to use the Centralized GSM solution to reduce costs by not equipping every sheep with GSM transceivers. Centralized GSM is described in Algorithm 3.

```
On init(){
        Clock.Synchronize();
        SleepUntil(first update);
}
On update(){
        delay = CalculateDelay(battery.status);
        Wait(delay);
        if(HasReceived(leaderMessage)){
           //become follower
           SendToLeader(id);
           SleepUntil(next update);
        }else{
          //become leader
          numFollowers = 0;
          followers = Ø;
          Wait(followerdelay);
          For(each receivedFollower){
             followers.Add(receivedFollower) ;
             numFollowers++;
          }
          pos = GPS.getPosition();
          GSM.SendToServer(pos, id,
          numFollowers, followers);
          SleepUntil(next update);
        }
}
```

Algorithm 3. The Centralized GSM algorithm.

IV. RESULTS

To find the best algorithm, three different investigation methods were used: Analysis, experiments and simulations.

A. *Analysis of the different solutions*

To be able to better understand the difference in energy consumption between the different solutions an analysis of them was performed. The analysis tries to calculate the expected number of updates each sheep can perform before running out of battery.

TABLE 1. LIBELIUM WASPMOTE ENERGY FIGURES.

Battery capacity (Bat)	6600 mAh
GPS consumption (GPS)	27.5 mA
GSM consumption (GSM)	100 mA
Processor consumption (Proc)	9 mA
XBee broadcast consumption, full power (XBc)	160 mA
XBee receive consumption (XRcv)	73 mA

The analysis use the energy consumption figures for the Libelium Waspmote [5]. This is the hardware that was used to measure the energy consumption of the different algorithms. The relevant hardware figures are listed in Table 1.

The analysis is based on a scenario where the sheep is always part of the same cluster and every sheep starts with a fully charged battery. Defining N as the number of sheep in a cluster, each sheep will be leader in 1/N of the execution cycles. U is defined as the average number of position updates a sheep can perform before running out of battery power. The following formulas were used for the different algorithms:

1) Telespor

$$U = \frac{\mathrm{Bat}}{\mathrm{GPS} + \mathrm{Proc} + \mathrm{GSM}} \qquad (1)$$

2) Distributed GSM

$$U = \frac{\mathrm{Bat}}{\frac{1}{N} * (\mathrm{GPS} + 2 * \mathrm{XBc}) + \frac{N-1}{N} * (2 * \mathrm{XRcv}) + \mathrm{GSM} + \mathrm{Proc}} \qquad (2)$$

3) Centralized GSM

$$U = \frac{\mathrm{Bat}}{(\frac{1}{N} * (\mathrm{GPS} + 2 * \mathrm{XBc} + (N-1) * \mathrm{Xrcv} + \mathrm{GSM}) + \frac{N-1}{N} * (2 * \mathrm{XRcv} + \mathrm{XBc}) + \mathrm{Proc})} \qquad (3)$$

When performed with a varying number of sheep in the cluster, the analysis was able to provide some insight into the properties of the different algorithms. Telespor is the preferred algorithm when there is only one sheep in a cluster, making it good for small flocks. For clusters with more than one sheep Distributed and Centralized GSM is better than Telespor. Distributed and Centralized GSM has almost the same energy consumption, however the gap between them increase with cluster size in favor of Centralized GSM. The analysis results can be viewed in Fig. 1.

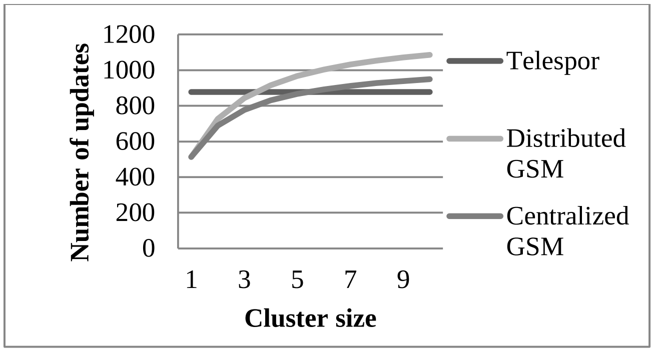


Figure 1. Analysis of average number of updates per sheep with different cluster sizes.

One algorithm was analyzed, in addition to the three mentioned earlier. This was a Centralized GSM algorithm were the nodes would broadcast every message they received to extend it into a multi-hop network, since this could increase flock size. It was decided that this algorithm would be dropped after the analysis showed that it performed worse than the other algorithms, especially in clusters with more than 10 sheep. Another point is that if the cluster covers a very large area, it gets harder for the farmer to find the sheep. With a multi-hop network it becomes more difficult to define the maximum area of a cluster. Limiting the number of hops could be a possible solution.

B. *Measurements of Energy Consumption*

The three different systems were implemented on the Libelium Waspmote [5] wireless sensor network platform. This is a platform were different modules can be attached when needed. The GPS and GSM modules were used during the energy consumption experiments. The GSM communication was performed by means of GPRS data packets with a maximum payload of 100 bytes. If a sheep cluster had more than 30 sheep, it would have to send additional packets and thereby slightly increase energy consumption. The communication between the nodes was carried out using an XBee 868 MHz transceiver attached to a 4.5 dBi antenna. This has five different power levels. On the highest power level it has an output of 315 mW and a mean range of 515 m [4]. The transceiver was set to transmit at the lowest power level during the energy measurements. This corresponds to an output of 1 mW, with an unmeasured range shorter than 515 m. The energy consumption analysis showed that the choice of power level was not a significant factor in the total energy consumption. There was only a 2% increase in battery life at the lowest power level compared to the highest. The battery used in the test had a voltage of 3.7 V and a capacity of 6600 mAh. All tests were performed with the nodes stationary, in close proximity and in an office environment.

TABLE 2. ENERGY CONSUMPTION MEASUREMENTS

Node type (number of nodes in test)	Average battery level percentage decrease per update	Standard deviation
Telespor (1)	0.5472	3.2
Distributed GSM leader (1)	0.1445	0.7
Centralized GSM leader (1)	0.4043	0.9
Distributed GSM leader (2)	0.2418	0.8
Distributed GSM follower (2)	0.0875	0.6
Centralized GSM leader (2)	0.7576	1.5
Centralized GSM follower (2)	0.0400	1.3
Distributed GSM leader (3)	0.3056	1.0
Distributed GSM follower (3)	0.0920	0.7

Six different setups were used for the experiments: Single node Telespor, Distributed GSM and Centralized GSM, two nodes running Distributed GSM and Centralized GSM and three nodes running the Distributed GSM algorithm. Every time one cycle of the algorithm had been completed the nodes would start the algorithm over again without sleeping, as the sleeping energy consumption should be equal among the three algorithms. All nodes reported their battery level back to the server in the same message used for reporting position. In the Centralized GSM case this information was sent via the current leader node. Each setup was tested with approximately 100 updates per node. The average decrease in battery level per update is displayed in Table 2 along with the standard deviation of the measurements. The most surprising result was the big difference between the energy consumption between the Centralized and Distributed GSM algorithms. It seems like the GSM use more energy than assumed in the analysis, however some of the difference might be caused by the battery level measurements not being 100 % accurate. Sometimes the battery level would actually increase between updates. This factor is probably the reason why Telespor has higher measured energy consumption than the Leader in the Distributed GSM algorithm. The Distributed GSM leader algorithm does everything the Telespor algorithm does, but also sends out two XBee messages and waits for synchronization with potential followers. This leads to the conclusion that some of the energy measurements must be wrong. Therefore we have chosen to conservatively set the Telespor energy consumption as 80% of the measured Distributed GSM consumption when comparing the algorithms in simulations.

C. *Simulation Setup*

The results from the energy consumption measurements was used as input to a Java simulator built for the purpose of evaluating the three different algorithms in a realistic sheep grazing scenario with between 50 and 250 sheep. The simulator placed the sheep randomly in a landscape measuring 5000 x 5000 meters. Every sheep was equipped with a transceiver that had a range which followed a gaussian distribution with a configurable average.

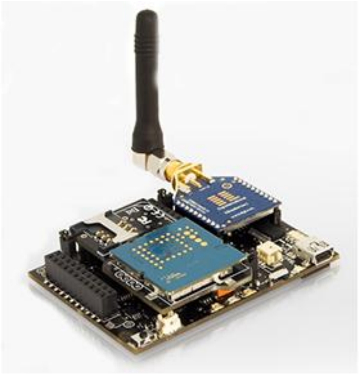


Figure 2. Libelium Waspmote [5].

Once placed, the sheep would discover the other sheep within their transceiver range and form clusters. The sheep would then start executing the leadership choice part of the two cluster-based algorithms. With a role as leader or follower the sheep would then deduct energy from the battery accordingly. This cycle of leader choice and energy deduction would continue until no sheep had any energy left in their battery. Each simulation scenario was repeated 1000 times and the figures presented in the results section are averages of these simulations. The simulations focus on number of updates instead of time until battery depletion. This is because all algorithms use the Waspmote hibernate mode between updates. In this mode the Waspmote consumes no energy from the main battery, since it only use the auxiliary clock battery to run the real-time clock. Therefore it is not necessary to consider the update interval when comparing the algorithms.

The Telespor solution was not simulated since it has no interaction between sheep. This means that it will have the same energy consumption no matter how the sheep are distributed. A simple calculation was done instead. If one Telespor update costs on average 0.244 % (80% of the measured consumption of the Distributed GSM leader) of the total energy in the battery, the battery would last approximately 410 updates.

D. *Simulation Results*

1) *Effect of Transceiver Range:* The transceiver range is adjustable by changing the transceiver power level. It is interesting to look at how this range affects the energy consumption. A shorter range will give smaller clusters and thereby increase the localization accuracy but it will also consume more energy. To test this, simulations with different transceiver ranges and a constant flock size of 250 sheep were performed. Table 3 shows the average number of updates per sheep in these simulations while Table 4 displays the average error.

The results indicate that it is possible to double the battery life by increasing the transceiver range from 100 to 500 m. An average error of approximately 300 m might be unacceptable, especially in areas with limited visibility such as forests.

TABLE 3. AVERAGE NUMBER OF UPDATES PER SHEEP WITH DIFFERENT TRANSCEIVER RANGES

Transceiver range/std. dev.	Distributed GSM	Centralized GSM
100 m/34 m	380	168
200 m/68 m	483	245
300 m/103 m	578	335
400 m/138 m	659	429
505 m/174 m	728	530

TABLE 4. AVERAGE ERROR [M] FOR EACH UPDATE WITH DIFFERENT TRANSCEIVER RANGES.

Transceiver range/std. dev.	Distributed GSM	Centralized GSM
100 m/34 m	17	20
200 m/68 m	77	90
300 m/103 m	149	165
400 m/138 m	223	234
505 m/174 m	299	318

Choosing a transceiver power level that gives a range of 300 m is a good trade-off between energy consumption and accuracy. When comparing the two algorithms it seems like the average error is quite similar.

The battery lasts longer using Distributed GSM than Centralized GSM, especially with a short transceiver range. There are simply not enough followers per leader to outweigh the added leader cost of the Centralized solution.

Clustering of Sheep: The two new algorithms presented in this paper use a cluster-based approach for localization and, in the Centralized version, also for data retrieval. Therefore, it is interesting to look at what effect the average cluster size has on the performance of the algorithms. To simulate sheep flocks with different tendencies to move in clusters, the sheep placement method was changed. First, a number of clusters were given a random position. Then, the sheep would be added to a random cluster and given the same position as that of the cluster. The simulations were performed with 250 sheep in the flock and the transceiver range set to 300 meters with a standard deviance of 103 meters. The average number of updates per sheep can be seen in Fig. 3. The centralized algorithm becomes better than the distributed when cluster size increases. The average energy consumption curves intersect at an average cluster size of approximately 8 sheep. This is not an unrealistic cluster size, especially if lambs are fitted with transceivers as well. The lambs always follow their sheep mother unless there has been an accident. Since each sheep typically have 2 lambs, there only needs to be 3 separate sheep in a cluster to get a total of 9 transceivers. If the lamb also carry transceivers the centralized algorithm is preferable. If not, the distributed algorithm will probably perform better, unless the flock is very large or in a small area. With a very small area there may not be much use for a sheep tracking system anyway.

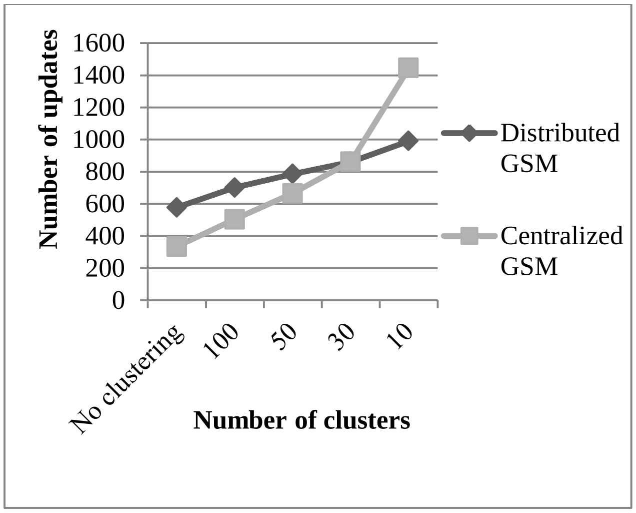


Figure 3. Average number of updates per sheep with different number of clusters.

2) GPS percentage: The sheep that are not leaders do not use GPS to find their position. To save money it could be possible to drop the GPS module from some of the sheep. These will then only have the possibility to be followers and can not find and report their position unless they are within range of another sheep with GPS. If this approach is to be implemented, it would be best to use the Centralized algorithm since it would then be possible to also abandon the GSM module on the nodes without GPS. The success of this method can be measured using the update failure ratio of the non-GPS nodes. The update failure ratio is the number of updates failed due to being out of range of a GPS-sheep, divided by the total number of updates performed by non-GPS nodes. Table 5 shows the update failure ratio for different flock sizes and number of clusters with a GPS percentage from 20-80 %. The transceiver range is kept at 300 m, with a standard deviance of 103 m and the centralized GSM algorithm is used. The GPS sheep are randomly chosen from the set of sheep. The results show that it is possible to drop the GPS and GSM modules on some of the sheep if the flock is big enough. 50 sheep is too few, but with more than 100 sheep and some clustering it is a good solution. Especially if lambs are included among those 100, since that increase the clustering effect.

TABLE 5. AVERAGE NON-GPS UPDATE FAILURE RATIO FOR DIFFERENT FLOCK SIZES AND NUMBER OF CLUSTERS.

Flock size and number of clusters	20 % GPS	50 % GPS	80 % GPS
50 sheep and no clustering	0.89	0.74	0.62
100 sheep and no clustering	0.79	0.55	0.39
250 sheep and no clustering	0.55	0.23	0.10
50 sheep and 30 clusters	0.65	0.33	0.18
100 sheep and 30 clusters	0.41	0.11	0.03
250 sheep and 30 clusters	0.12	0.01	0.00

TABLE 6. AVERAGE NUMBER OF UPDATES PER SHEEP WITH DIFFERENT FLOCK SIZES

Flock size	Distributed GSM	Centralized GSM
50 sheep	409	189
100 sheep	469	235
150 sheep	514	272
200 sheep	549	305
250 sheep	578	335

A good solution could be to equip the sheep mothers with GPS and the lambs with no GPS, since lambs would be able to report their position as long as they stay with their mother.

3) Flock size: The flock size is a factor in choice of algorithm, but not as important as the clustering effect. However, a bigger flock can lead to bigger clusters. The simulation results in Table 6 show the average number of updates with different flock sizes. The simulations were performed with no clustering effect and a 300 m transceiver range. The performance varies with flock size, but with no clustering effect the Distributed GSM outperforms the Centralized GSM for all common flock sizes. For flocks with less than 50 sheep and no clustering effect the most efficient algorithm is Telespor.

V. CONCLUSION AND FUTURE WORK

The energy measurements and simulations showed that it is possible to reduce the energy consumption of a sheep tracking network by more than 50% by using the Centralized GSM algorithm in a sheep flock with big clusters. In small flocks the Telespor solution is still the best since it has the lowest cost in clusters with only one sheep. In flocks with an average cluster size of approximately five sheep the Distributed GSM is the most energy efficient algorithm. The Centralized GSM is still preferable in these situations due to the possible cost reduction of not having to equip every sheep with GSM modules.

The best setup for a typical farmer with a flock consisting of more than 100 animals will be to equip all the sheep mothers with full functionality GSM and GPS nodes running the Centralized GSM algorithm. The lambs could then be equipped with a simpler version without GSM and GPS. This will reduce cost without increasing the update failure ratio since lambs follow their mothers.

The next step in this project is to simulate and check the efficiency of the Centralized algorithm using real-world sheep location data. Telespor has allowed us access to a data set with over 400 000 sheep positions. Using this data will give a clearer indication of how much performance gain can be expected. It will be interesting to look at how big the clustering effect is, as some sheep prefer walking alone while others are more social. With real-world data it is possible to simulate movement and thereby look at the possibility of enhancing the algorithm if some sheep stay together in a group over time.

The final step in this project should be a real-world deployment with as many sheep as possible. This is the only way to really prove the performance of the Centralized GSM algorithm.

REFERENCES

[1] Telespor. Company Website. Available: http://www.telespor.no. Last accessed: 24th Apr 2012.

[2] B. Thorstensen, T. Syversen, T-A. Bjørnvold, and T. Walseth, "Electronic shepherd – a low-cost, low bandwith, wireless network system," Proc. of the 2nd international conference on Mobile systems applications and services, ACM, 2004, pp. 245-255.

[3] A. S. Haugset and G. Nossum, "Erfaringer med bruk av elektronisk overvåkningsutstyr på sau i 2010," Notat 2010:17 Trøndelag Forsking og Utvikling AS, 2010.

[4] R. Stølsmark and E. Tøssebro, ”Can a multi-hop solution improve GSM coverage for tracking networks?,” Proc. of the european conference on the use of modern information and communication technologies (ECUMICT 2012), Nevelland v.z.w Landegem, Mar 2012, pp. 95-106.

[5] Libelium Waspmote: Product information. Available: http:// http://www.libelium.com/products/waspmote. Last accessed 24th Apr 2012.

[6] R. Stølsmark and E. Tøssebro, “Uncertainty in trilateration: Is RSSI-based range estimation accurate enough for animal tracking?,” Proc. 1st international conference on sensor networks (SENSORNETS) 2012, ScitePress, Feb 2012, pp. 237-241.

[7] Huircán, J. I., Muñoz, C., Young, H., Von Dossow, L.,Bustos, J., Vivallo, G. and Toneatti, M., “ZigBee-based wireless sensor network localization for cattle monitoring in grazing fields,” in Computers and Electronics in Agriculture vol. 74(1), Elsevier, 2010, pp. 258-264.

[8] Juang P., Oki H., Wang Y., Martonosi M., Peh L.S. and Rubenstein D., “Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet,” Proc. of the 10th international conference on Architectural support for programming languages and operating systems (ASPLOS-X), ACM, 2002, pp. 96-107.

[9] Zhang P., Sadler C. M., Lyon S. A. and Martonosi M., ”Hardware design experiences in ZebraNet,” Proc. of the 2nd international conference on Embedded networked sensor systems (SENSYS ‘04) , 2004, ACM, pp. 227-238.

[10] Dyo V., Ellwood S. A., Macdonald D. W., Markham A., Mascolo C., Pásztor B., Scellato S., Trigoni N., Wohlers R. and Yousef K., “Evolution and sustainability of a wildlife monitoring sensor network,” Proc. of the 8th ACM Conference on Embedded Networked Sensor Systems (SENSYS ’10), 2010, ACM, pp. 127-140.

[11] Markham A., Trigoni N., Ellwood S. A. and Macdonald D. W., “Magneto-inductive tracking of underground animals” Proc. of the 8th ACM Conference on Embedded Networked Sensor Systems (SENSYS ’10), 2010, ACM, pp. 365-366.

[12] Polastre J., Szewczyk R., Mainwaring A., Culler D. and Anderson J., “Analysis of wireless sensor networks for habitat monitoring” in Wireless Sensor Networks, 2004, Springer, pp 399-423.

[13] Yick J., Mukherjee B. and Ghosal D., “Wireless sensor network survey” in Computer Networks vol. 52(12), 2008, Elsevier, pp. 2292-2330.

[14] Akyildiz I. F., Su W., Sankarasubramaniam Y., Cayirci E., “Wireless sensor networks: a survey,” in Computer Networks vol 38(4), 2002, Elsevier, pp. 393-422.

On Estimating Mote Operation Times During Typical Cross-Layer Functions

Sankarkumar Thandapani and Aravind Kailas
Dept. of Electrical Engineering
The University of North Carolina at Charlotte (UNC Charlotte)
9201 University City Blvd,
Charlotte, NC 28223-0001
Email: sthandap@uncc.edu, aravindk@ieee.org

***Abstract*—When it comes to deploying large scale, low power wireless networks comprising battery-operated embedded systems or motes, frequent replacement of batteries is undesirable. This motivates estimating the energy consumption in wireless motes accurately prior to deployment, and would avoid "sudden" decreased network coverage owing to pre-mature mote deaths. Using our proposed model, the energy consumption in the PHY layer was found to be within 10–15 % of the actual value obtained using measurements, and corresponded to an accuracy of 2–10% in the mote life. Furthermore, the cross-layer energy profiling involving the MAC layer provided insights into the energy consumed during the key modes of a simple, practical MAC protocol. To summarize, the primary purpose of this paper is to profile the energy consumption in a mote using a novel "off-line" model to predict its operation life with high accuracy.**

***Index Terms*—Modeling and simulation of systems, practical medium access control protocols (MACs), energy efficiency**

I. Introduction

Most of the wireless motes are battery backed and are deployed in remote areas. Hence, replacing their batteries could be extremely difficult [1]. Accurate energy profiling of the motes still remains a key challenge in modeling the sensor networks [2]. The lifetime of sensor nodes may often be significantly shorter than expected. Szewczyk *et al.* found that their habitat-monitoring wireless network shrunk drastically due to mote failures within four days of deployment [3]. It has been observed that there were as many as 50 % of the wireless motes unexpectedly owing to inaccuracies in the energy consumption model. Thus, energy evaluation before deployment is extremely important. Hence, an accurate energy consumption model would help in avoiding expensive unexpected mote failures. The outcome of our research work is a simple, novel, system-level, "off-line" tool that accurately models the energy consumption exclusively in the PHY layer, (i.e., the transceiver (or the radio) and the micro controller (μC)) and a cross-layer energy consumption profiling involving the MAC layer. In areas involving commercial and scientific applications, low-power networks that operate in the industrial/scientific/medical (ISM) bands of $2.4\,\mathrm{GHz}$ are being widely adopted. Considering this, the effectiveness of the proposed model has been validated using commercial ZigBee-ready motes [4].

A wireless sensor network is composed of many sensor motes. A wireless mote comprises of transceiver, μC and sensor. In our paper, the PHY and the medium access control (MAC) layer are the focus areas for the energy consumption model. The classification of a transceiver would include radio frequency (RF) front-end (FE) and baseband processing back-end (BE). The front-end (FE) would in turn comprise of low noise amplifiers (LNAs), power amplifiers (PAs), mixers, filters, voltage controlled oscillators (VCOs), frequency synthesizers, intermediate frequency amplifiers (IFAs), automatic gain control (AGC) units, analog-to-digital converters (ADCs), and digital-to-analog converters (DACs) [5], [6]. The back-end (BE) comprises of blocks that perform functions as modulation, demodulation, error detection and correction, and pulse shaping.

The motes available in the industry are LOTUS [7], TelosB [8], MICAz [9], IRIS [10], CRICKET [11], etc. While deploying the mote, it is important to choose the right mote. It is remarked that measurements are necessary to validate the simulations, and there is the possibility to perform these measurements. However, it is very complex to measure the power directly and easier to simulate the energy consumption in a mote. It is necessary to treat the chip as one entity. In a simulation, the energy consumption of each block can be obtained and this information is useful when energy hogs in a mote needs to be studied. The proposed algorithm for modeling the energy consumption is "generic" in the sense that some of the more traditional receiver (e.g., low-power IF) and transmitter (e.g., direct-up

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-207-3

conversion) architectures that are very specific in their implementations [12].

The organization of the paper is as follows. Sections II and III describe the related state-of-the-art in predicting the energy consumption in wireless motes, and the novel energy consumption model. In Section IV, the experimental setup, methodology, and the results are presented. Finally, Section V has the concluding remarks resulting from this work.

II. Related Work

Although recent research provides many system-level energy consumption models, the accuracy of the model and completeness still remains an open topic to research. For instance, Cui *et al.* developed an energy model for low-power wireless motes to analyze the best modulation technique and transmission strategy to minimize energy consumption [6]. However, the energy costs associated with the μC and the modulation techniques were not considered; using our model, we show that when the modulation technique is considered, the architecture of the transceiver changes and the energy consumption increases by approximately 100%. The power consumption of the μC is non negligible too [14]. Our energy model considers the μC energy model which results in the higher accuracy of predicting the lifetime of the mote. The simulator presented in [15] considers the energy profiling of the μC but there a problems such as over counting and under counting of instructions which has been eliminated in our model. An energy model based on the transceiver battery life has been presented in [1]; however, the model did not account for the power consumption in the modulator, filters, the ADC, and the DAC. Using our model, we show that the powers consumed by the ADC and DAC are $1.4\,\mathrm{mW}$ and $19.01\,\mathrm{mW}$, respectively, and cannot be disregarded.

Another recent energy model took into consideration most of the RF FE blocks with the exception of the pulse shaping filter because it is usually very low relative to the other “energy-hogs” [5]. However, the model also did not consider other baseband functions such as modulation and coding.The energy consumption in the MAC layer has not been studied in any of the above mentioned works. Kohvakka *et al.* in [17] have shown that the energy consumption due to the MAC depends on the number of nodes and the beacon interval. Our analysis of the energy consumption due to the MAC layer shows the key energy consuming modes in a MAC protocol. To summarize, the energy profiling of the μC is important to accurately estimate the energy consumption in PHY layer. The model proposed in the paper takes into consideration the effects of the modulation technique along with the baseband processing blocks, the μC and the MAC layer resulting in a more accurate and complete estimation of the lifetime of a mote.

III. The Energy Consumption Model

The proposed energy consumption model has been adapted to the on-chip radio architecture [18], [19] to accommodate the preferred modulation technique, offset-quadrature phase shift keying (O-QPSK) and is shown in Fig. 1. The energy equation of each block proposed by many researchers has been tweaked to accommodate them as a part of the whole chip. Parameters such as voltage, device dimensions, frequency, data rate are common to all components in the chip and do not vary. Hence, they can be considered as constant and the expression becomes a function of the variables.

A. Physical Layer in a Wireless Mote

A PHY layer of the wireless mote mainly consists of the transceiver and the μC. The energy consumption in the PHY layer is given as: $E_{\text{Mote}} = E_{\text{Transceiver}} + E_{\mu\text{C}}$.

1) The Transceiver: We have used MICAz motes to validate our model because of their wide popularity and extensive use in the sensor network community. MICAz motes operate on $2.4\,\mathrm{GHz}$, and support data rates of up to $250\,\mathrm{kbps}$ [18], [19]. The operating voltage range is $2.1-3.6\,\mathrm{V}$ [19], and traditionally has four distinct operation modes depending upon the power requirements. The system-level breakdown of the energy consumption in the RF transceiver is given by $E_{\text{transceiver}} = \sum_i E_{\text{FE},i} + E_{\text{BE},i}$, where $i \in \{\text{Transmitter}, \text{Receiver}\}$, and E_{FE} and E_{BE} denote the energy consumptions in the front- and back-ends, respectively.

The RF FE of a transmitter comprises of a DAC, low-pass filter (LPF), mixer, PA, and the BE is the digital spreader. Similarly, the RF FE of the receiver is made up of a ADC, IFA, band pass filter (BPF), mixer, and a digital despreader as the BE block. The frequency synthesizer (FS) is common to both the transmitter and receiver architectures. Therefore, a simple unified expression for the energy consumption in the transceiver

ISBN: 978-1-61208-207-3

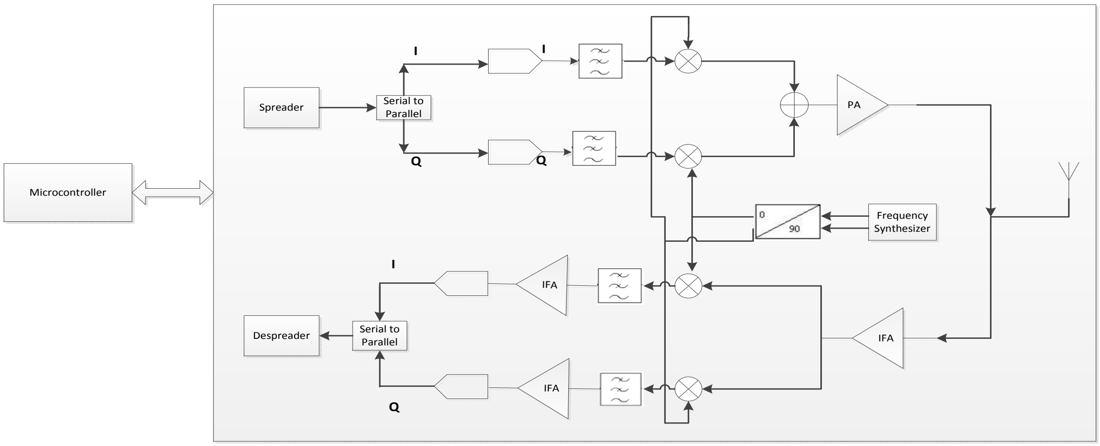


Fig. 1. Generic physical layer architecture of a wireless mote (a MICAz mote, in this paper).

is given by:

$$
\begin{aligned}
E_{\text{transceiver}} = t_{\text{tx}} \Bigg[& \underbrace{P_{\text{Spreader}}}_{\text{BE, Transmitter}} \\
& + \underbrace{2P_{\text{DAC}} + 2P_{\text{LPF}} + 2P_{\text{Mixer}} + P_{\text{FS}} + P_{\text{PA}}}_{\text{FE, Transmitter}} \Bigg] \\
+ t_{\text{rx}} \Bigg[& \underbrace{P_{\text{Despreader}}}_{\text{BE, Receiver}} \\
& + \underbrace{2P_{\text{ACD}} + 2P_{\text{IFA}} + 2P_{\text{Mixer}} + P_{\text{FS}} + P_{\text{LNA}}}_{\text{FE, Receiver}} \Bigg],
\end{aligned}
$$

where t_{tx} and t_{rx} are the time durations during which the mote is operating in the transmitting or the receiving mode, respectively. Next, the simplified analytical models for the principle "power hogs" are listed:

- **Power amplifier:** $P_{PA} = \alpha P_{\text{out}}$, where α is a constant that depends on efficiency of amplifier and peak to average ratio and P_{out} is the output power.

- **RF Filter:** $P_{Filter} = \beta \text{SNR}^2\text{BW}$, where SNR is the signal to noise power ratio, β is a constant that depends on Boltzmann's constant, the temperature, and BW is the bandwidth of operation.)
- **Low noise amplifier:** $P_{LNA} = \gamma \frac{\text{A}}{\text{NF}}$, where γ is the proportionality constant, A is the gain of the low noise amplifier, and NF is the noise figure.

- **Intermediate frequency amplifier:** $P_{IFA} = \delta(\text{BW} + \text{f}_0)\sqrt{\alpha_{BA}}$ [5], where δ is a coefficient which depends on the device dimensions and process parameters, BW is the bandwidth of the baseband amplifier, f_0 is the center frequency, and α_{BA} is the baseband amplifier gain.

- **Spreader (and Despreader):** $P_{\text{Spreader}} = P_{\text{XOR}} + \text{N}P_{\text{SR}}$, where P_{XOR} is the power consumption of the XOR gate, P_{SR} is the power consumption of the shift register, and N is the number of shift registers.

2) The μC: The μC on a MICAz mote is ATmega128L, a low-power CMOS 8-bit μC based on the enhanced reduced instruction set computer (RISC) architecture. The energy consumption of a μC can be given as: $E_{\mu\text{C}} = \frac{\text{I}*\text{V}*\text{N}}{\text{F}}$, where I is the current supply, V is the voltage supply, N is the number of cycles, F is the frequency of operation.

The operating system on the MICAz is TinyOs. TinyOS is a free and open source component-based operating system and platform targeting wireless sensor networks. TinyOS applications are written in nesC, an extension to the C programming language designed to embody the structuring concepts and execution model of TinyOS. The nesC code has been converted into assembly language code, a low-level programming language for μCs using XATDB, the debugger of a sensor network simulator [21] in order to find the energy consumed by the μC while running the executable code. The assembly language code is used to compute the cycle count using XATDB and is substituted in the above equation.Data such as the supply voltage, supply current, and frequency of operation has been taken from the ATmega 128L data sheet [22]. The predicted energy consumption of the μC to run the executable code is $15\,$mJ.

B. MAC Layer in a Wireless Mote

The MAC protocol on a MICAz follows the IEEE 802.15.4 protocol [9], and supports two kinds of modes: beacon and non-beacon enabled. In a beacon enabled mode, the motes synchronize with each other and transmit only during their specified beacon. In a non-beacon mode carrier sense multiple access with collision avoidance (CSMA/CA) is used in order to avoid collision of the packets.

1) CSMA/CA: In CSMA/CA protocol, as soon as a node receives a packet that needs to be transmitted, it checks if the channel is available, and transmits it. If the channel is busy, the node waits for a randomly chosen period of time, and then checks again to see if the channel is available. If the channel is clear when the back-off counter reaches zero, the node transmits the packet. If the channel is not clear when the back-off counter reaches zero, the back-off factor is set again, and the process is repeated.

2) Beacon mode: It is an energy efficient mode and, hence choice for our model. The beacon enabled mode is used by motes to synchronize with each other. The communications are performed in a super frame structure illustrated in Fig. 2. There are three main parts in a super frame viz. the beacon, contention access period (CAP) and contention-free period (CFP). The nodes enter into power saving mode at the end of the super frame. The coordinators listen to the channel during the whole CAP to detect and receive any data from their child nodes. The child nodes may only transmit data and receive an optional acknowledgement (ACK) when needed.

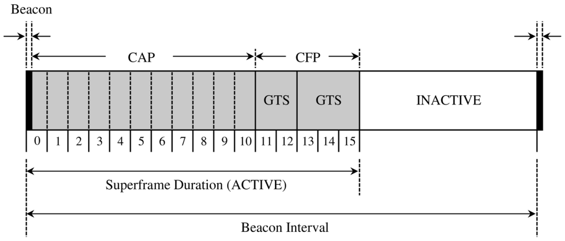


Fig. 2. Superframe structure (Beacon-enabled mode) implemented.

Contention access period (CAP) is the time interval during which the coordinators listens to the channel during the whole CAP to detect and receive any data from their child nodes. The child nodes may only transmit data and receive an optional acknowledgement (ACK) when needed. In star networks, a device may obtain better Quality- of-Service (QoS) by the use of guaranteed time slot (GTS), since contention and collisions are avoided. The superframe duration (SD) is the time interval between two super frames. Similarly the beacon interval (BI) is the time duration between two beacons.

3) Cross-Layer Energy Profiling: The energy consumption during a MAC mode is given by $E_i = E_{\text{i,Beacon}} + E_{\text{i,Direct}} + E_{\text{i,Indirect}} + E_{\text{i,Sleep}}$, i $\in$ $\{\text{mote, coordinator}\}$. In the **Beacon Mode** the transmitted beacon by the coordinator is received by the mote during this mode.In the **Direct Mode**, the mote exchanges data with the coordinator in its specified beacon slot.In the **Indirect Mode**, downlink data from a coordinator to its child node are sent indirectly requiring totally four transmissions. The availability of pending data is signaled in beacons. First, a child node requests the pending data by transmitting a data request message. The coordinator node responds to the request with ACK frame, and then transmits the requested data frame. Finally, the child node transmits ACK if the data frame was successfully received. A schematic of the start-topology is shown in Fig. 3.

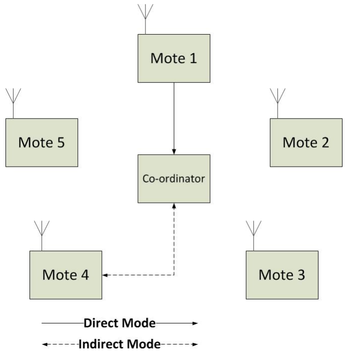


Fig. 3. Schematic of the star-network topology implemented using MICAz motes.

IV. Results and Discussions

A. Experimental Setup

The energy consumption due to the PHY layer of the mote was verified experimentally. A source-destination link was implemented using MICAz motes as shown in Fig. 4. A payload of 20 bytes was transmitted by the source every $250\,\text{ms}$, and the base station (i.e., the destination) received payload at intervals of $250\,\text{ms}$. A sampling rate of $400\,\text{Hz}$ was chosen to log the voltage and current consumption at the two motes to calculate the power consumption. The test consisted of 100 samples of current and voltage every second. The data was collected over a period of three days. The power consumed during a random hour was calculated and plotted with the simulation values. The samples obtained from the data logger were used to compute the power consumption per sample, which were then averaged to obtain $P_{\text{Transmitter}}$ and P_{Receiver}, the power consumed during the transmit and receive modes, respectively. The wireless motes were powered by two AA-sized batteries, each rated at $3000\,\mathrm{mAh}$ [25], which when multiplied with the operating voltage yielded the initial residual energy, which in turn was used in the estimation of the mote operation life.

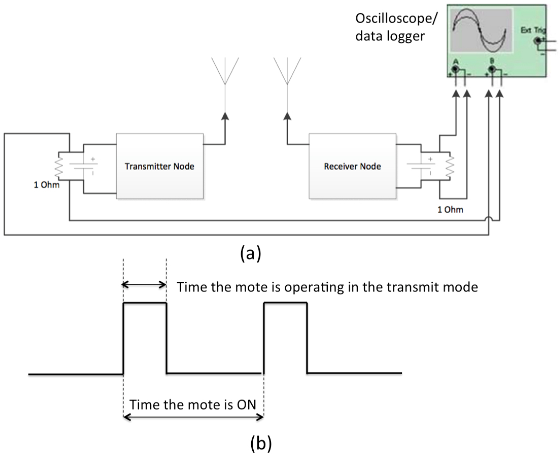


Fig. 4. (a) Illustration of the experimental setup and (b) timing diagram.

B. PHY Layer Energy Consumption Analysis

MATLAB [23] was used to implement the analytical model. The parameters for the transceiver model were selected from the CC2420 data sheet [19]. The energy consumption in the two modes of operation (transmit and receive) were estimated using the approach described in Section III. A, and the mote operation lives were compared to the experimentally measured values to validate the analytical model.

Figs. 5 and 6 illustrate the variation of residual energy (in mJ) with time of operation (in hours) at the transmitting and receiving motes, respectively. From Figs. 5 and 6, it can be observed that the energy consumptions of the transmitter and receiver obtained using our analytical model, are within 10.6% and 15% of the experimentally measured values. These differences can be accounted for by the light-emitting diodes (LEDs), battery leakage, on-board passive elements (i.e., resistors and capacitors) and voltage regulator that have not been considered in the analytical model.

The battery self discharge is negligible (typically 2–3% per month) in comparison to the discharge due to the load, and hence, neglected. Figure 7 illustrates the energy consumption (in mJ) for a certain payload, in the transmitting and receiving motes. The higher accuracy of the proposed model can be explained by the inclusion of the effects of digital modulation technique in the energy consumption computations along with the energy costs associated with the spreader (and de-spreader) and

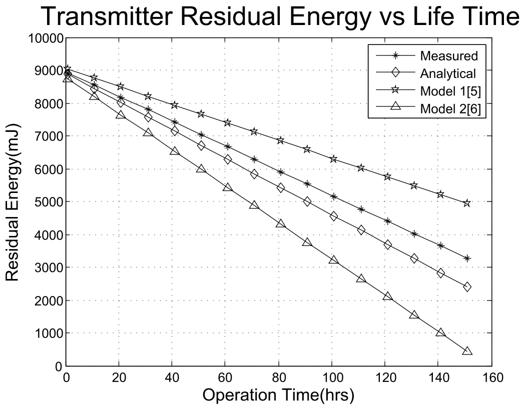


Fig. 5. Transmitter residual energy versus mote operation life (in hours).

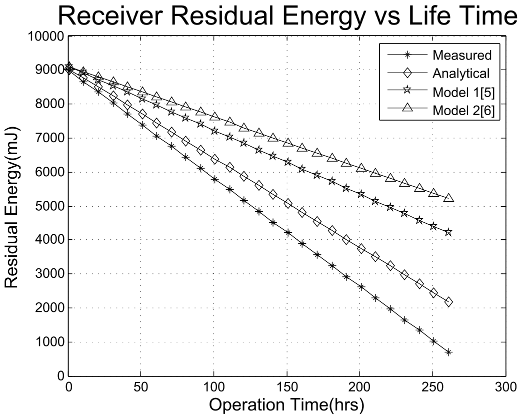


Fig. 6. Receiver residual energy versus mote operation life (in hours).

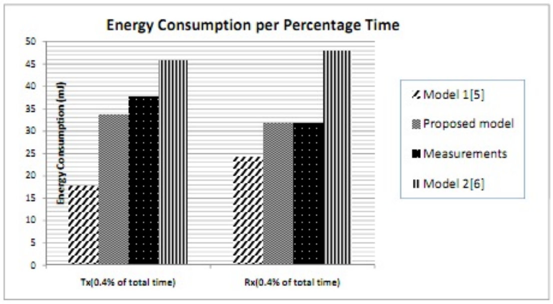


Fig. 7. Energy consumption of the transceiver for a 0.4% mode duration (i.e., transmit or receive).

μC. The linear equation for the transmitter analytical model is given by $\mathcal{R} = -33.678t + c$, where $\mathcal{R}$, t, and c denote the instantaneous residual energy on the battery, operation hours, and the total residual energy on the battery. The slope is the parameter that produces the change in the curve since the residual energy was assumed to be the same across the different models. Similarly, the equation of the receiver analytical model is given: $\mathcal{R} = -26.332t + c$.

C. Cross-Layer Energy Consumption Analysis

A star topology comprising five MICAz motes (referred to as the daughter motes) connected to a "coordinator" MICAz mote operating in the beacon-enabled mode. The energy consumption of each daughter mote is 1.25% of the coordinator mote, and is due to the higher duty cycle of the coordinator mote. Fig. 8 shows the time for the first mote and the coordinator to die. The burden of coordinating and being the "fusion" point for the cluster results in the coordinator node operating more, and hence dying earlier. While this should not be surprising, our analytical model helped predict this accurately and verifying this using our experiments. Next, the energy consumption of each mode in terms of the percentage of total energy consumption is given in Fig. 9. Again, not surprisingly, it can be seen that direct and indirect modes consumed the majority of the energy, again accurately predicted using our analytical model.

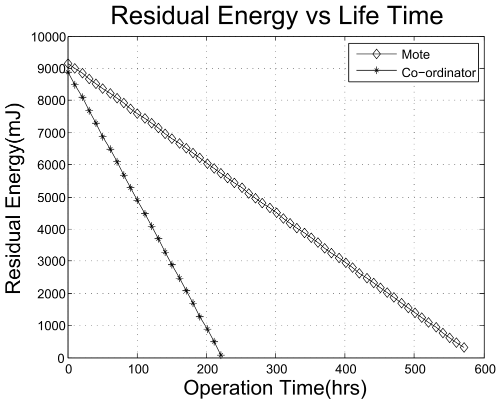


Fig. 8. Lifetime of the daughter and the coordinator motes

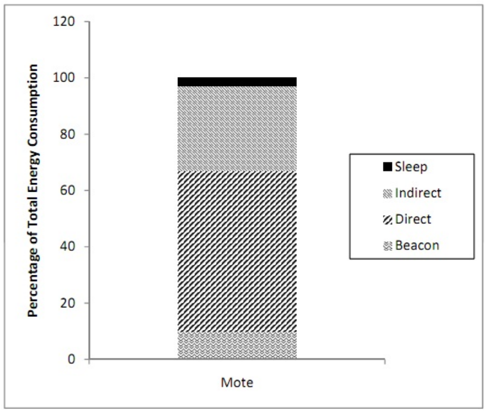


Fig. 9. Percentage of energy consumption per mode

V. Conclusion and Future Work

The wide-scale deployment of inexpensive wireless motes for networking hinges on accurately estimating the mote operation lives prior to deployment. Miscalculations in their estimations can prove costly, because of untimely, undesirable network partitioning. Using our unified, system-level energy consumption model, the predicted mote lives were found to be within 2–10% of the measured values. The high accuracy stems from the inclusion of the energy costs associated with the on-board functions such as control (i.e., the μC) and digital baseband processing (such as modulation, demodulation, spreading, and de-spreading).

By analyzing and expressing the results in terms of the three modes of operation of a mote (i.e., transmitting, receiving, and idle), our model gives better insights into the principle power hogs in the PHY layer of the mote during each mode. Furthermore, the cross-layer energy profiling involving the MAC layer provided insights into the energy consumed during the key modes of a simple, practical MAC protocol, and serves as motivation for future work in developing energy-efficient MAC protocols.

References

[1] A. Wang and C. Sodini, "A simple energy model for wireless microsensor transceivers," *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 5, Nov.–Dec. 2004, pp. 3205.

[2] O. Landsiedel, K. Wehrle, and S. Gotz, "Accurate prediction of power consumption in sensor networks," *The Second IEEE Workshop Embedded Networked Sensors (EmNetS-II)*, May 2005, pp. 37–44.

[3] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler, "Lessons from a sensor network expedition," 2004, pp. 307–322.

[4] Available: `http://www.zigbee.org`. Retrieved: Aug. 2012.

[5] Y. Li, B. Bakkaloglu, and C. Chakrabarti, "A system level energy model and energy-quality evaluation for integrated transceiver front-ends," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 1, Jan. 2007, pp. 90–103.

[6] S. Cui, A. Goldsmith, and A. Bahai, "Energy-efficiency of mimo and cooperative mimo techniques in sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 6, pp. 1089–1098, Aug. 2004.

[7] Available: `http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html`. Retrieved: Aug. 2012.

[8] Available: http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html. Retrieved: Aug. 2012.

[9] "MICAz Datasheet," 2004. [Online]. Available: http://www-db.ics.uci.edu/pages/research/quasar/MPR-MIB\%20Series\%20User\%20Manual\%207430-0021-06A.pdf. Retrieved: Aug. 2012.

[10] Available: http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html. Retrieved: Aug. 2012.

[11] Available: http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html. Retrieved: Aug. 2012.

[12] P. -I. Mak, S. -P. U, and R. Martins, "Transceiver architecture selection: Review, state-of-the-art survey and case study," *IEEE Circuits Syst. Mag.*, vol. 7, no. 2, pp. 6–25, 2007.

[13] X. Jiang, *et al.*, "Architecture for energy management in wireless sensor networks," *SIGBED Rev.*, vol. 4, pp. 31–36, July 2007. [Online]. Available: http://doi.acm.org/10.1145/1317103.1317109. Retrieved: Aug. 2012.

[14] M. Kramer and A. Geraldy, "Energy Measurements for MicaZ Node", Univ. of Kaiserslautern, Germany,2006

[15] V. Shnayder *et al.*, "Simulating the power consumption of large-scale sensor network applications," *Proc. 2nd Intl. Conf. on Embedded Networked Sensor Systems*, 2004, pp. 188–200.

[16] G. Merrett, *et al.*, "An empirical energy model for supercapacitor powered wireless sensor nodes," *Proc. 17th Intl. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2008, pp. 1–6.

[17] M. Kohvakka, *et al.*, "Performance analysis of IEEE 802.15.4 and ZigBee for large-scale wireless sensor network applications," *Proc. 3rd ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks*, 2006, pp. 48-57.

[18] N. -J. Oh and S. -G. Lee, "Building a 2.4-GHz radio transceiver using IEEE 802.15.4," *IEEE Circuits and Devices Mag.*, vol. 21, no. 6, pp. 43–51, Jan.–Feb. 2005.

[19] "CC2420 Datasheet," 2004. [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2420.pdf. Retrieved: Aug. 2012.

[20] G. Bertoni, L. Breveglieri, and M. Venturi, "Power aware design of an elliptic curve coprocessor for 8 bit platforms," *Fourth Annual IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerComWorkshops)* Mar. 2006.

[21] J. Polley, *et al.*, "ATEMU: a fine-grained sensor network simulator," *First Annual IEEE Com. Soc. Conf. on Sensor and Ad Hoc Communications*, pp. 145- 152, Oct. 2004.

[22] "ATmega 128L Data sheet," 2011.[Online]. Available: http://www.atmel.com/dyn/resources/proddocuments/doc2467.pdf

[23] Available: http://matlab.com. Retrieved: Aug. 2012.

[24] "TinyOS Application Code," 2010. [Online]. Available: http://code.google.com/p/tinyos-main/source/browse/\#svn\%2Ftrunk\%2Fapps\%2FRadioSenseToLeds. Retrieved: Aug. 2012.

[25] "Energizer AA Battery Data sheet" 2010. [Online]. Available: http://data.energizer.com/PDFs/191.pdf. Retrieved: Aug. 2012.

A Wireless Sensor Network to Study the Impacts of Climate Changes in Agriculture: The Coffee FACE in Brazil

André Torre-Neto
Embrapa Agricultural Instrumentation
São Carlos, SP Brazil
andre@cnpdia.embrapa.br

Raquel Ghini
Embrapa Environment
Jaguariúna, SP Brazil
raquel@cnpma.embrapa.br

***Abstract*—Climate change is considered one of humankind's greatest challenges in the near future. The climate change is expected to interfere in the scenario of worldwide agriculture. Its economic, social and environmental impacts can be positive, negative or neutral, since these changes can decrease, increase or have no impact on plant diseases, pests or weeds depending on each region or period of time considered. A type of experiment called FACE, Free Air Carbon-dioxide Enrichment, has been conducted in the USA, UK, Germany, Japan, Australia, Italy, Denmark, among other countries, to study particularly the impacts of the CO2 concentration increasing on crops. In Brazil, the first FACE experiment in South America has been installed by a group of scientists of Embrapa (Brazilian Agricultural Research Corporation). Compared to the existing FACE projects, the Brazilian implementation has been innovated with a wireless sensor network approach. In the present article, we describe the design and some operational aspects of that implementation.**

Keywords-Wireless Sensors; Environment Monitoring; Plant Diseases; Climate Change; FACE Facility.

I. INTRODUCTION

The global atmospheric CO_2 concentration is increasing rapidly in the last decades and despite the international efforts for the reduction of CO_2 emission. It will probably continue increasing and a long period will be necessary for it to return to the previous concentration [1]. The effects of high CO_2 atmospheric concentration on crops are often observed in the host plant, resulting in alterations in the host-pathogen relationship. CO_2 enrichment promotes changes in plant metabolism, growth and physiological processes. There is a significant increase in the photosynthetic rate and a decrease in the transpiration rate per unit of leaf area, while total plant transpiration sometimes increases, due to the larger leaf area. Despite the evidence of beneficial effects of CO_2 on the host plant, it is not well known whether these effects will still take place in the presence of pathogens, pests and weeds or other limiting factors, particularly in tropical countries [2]. Few studies have been conducted in controlled conditions. They might not reflect plant responses in the field, where there are variations and interactions among temperature, precipitation, and other factors. The search for more realistic conditions has led to the use of open-top chambers (OTCs) and Free Air Carbon-dioxide Enrichment (FACE) experiments [3].

In Brazil, the first FACE facility has been installed near Jaguariúna city - state of São Paulo, besides the installation of six OTCs experiments throughout the country (Belém, PA; Petrolina, PE; Sete Lagoas, MG; Londrina, PR; Jaguariúna, SP; and Vacaria, RS). The project named "Impacts of climate change on plant diseases, pests and weeds", with the nickname "Climapest", has been supported by Embrapa (Brazilian Agricultural Research Corporation). The severity of diseases and pests, weeds, plant development, interaction with microorganisms, plant nutrition, production and other possible impacts will be evaluated. The Climapest-FACE has been planned to discover the effects of high CO_2 concentration on coffee diseases, pests and weeds, as well as plant characteristics. The studies with forest species, apple, peach, soybean, grape, corn, cotton, castor beans, forage crops, coffee, cassava and banana will be conducted in the OTCs.

There are more than 30 FACE facilities around the world. They consist of a set of circles having pipes around them to perform the CO_2 fumigation. The fumigation can be achieved by direct injection or prediluted injection [4]. In either case the main operational issue is to maintain acceptable fluctuations and gradients of the CO_2 concentration inside the circles, which are affected mostly by the wind. In Brazil it was chosen the direct injection system and an octagonal arrangement of pipes, which is generally utilized in existing installations. Each octagon segment has individual gas valves to compensate the wind direction and a flow control device to compensate the wind speed changes. The OTCs have smaller circles, around 2m in diameter, and a plastic cover with an open top surrounds them. The basic instrumentation for a FACE and OTCs experiments usually consists of an Infra Red Gas Analyzer (IRGA) to measure the CO_2 concentration, an anemometer, a set of proportional or on-off valves and other environmental sensors like air temperature and humidity, solar radiation and precipitation. The improvement that has been accomplished for the Brazilian FACE and OTCs instrumentation is to operate all those devices based on the Wireless Sensor Network technology, already present in the rural area [5] and which is the expertise of the Brazilian FACE implementation group [6]. This approach has simplified the system installation and maintenance and has improved its electromagnetic compatibility, since lightning is a huge issue in Brazil.

II. MATERIALS AND METHODS

By the time this project started (January 2009), most necessary instruments were not commercially available as wireless devices. Therefore, the decision was to buy

ISBN: 978-1-61208-207-3

conventional sensors and actuators, as well as wireless ZigBee based modules, to develop a general-purpose interface circuit to integrate those parts to achieve the required wireless sensor network devices. In Table 1, it is shown a list of the chosen devices and the features considered for developing the interface circuit. The CO_2 sensor model GMP343 was selected for the FACE experiment and the CO_2 sensor model GMM222 was chosen for the OTCs experiment. The weather devices, i.e., the anemometer, the sensors of air temperature, air humidity, precipitation and barometric pressure are all part of the same instrument, the WXT520 weather station.

The wireless modules were purchased from Telegesis Inc., specifically the ETRX3 series. They incorporate the ZigBee protocol and operate in 2.4 GHz. They are IEEE 802.15.4 compliant and they are expected to operate in the planned range of 100 meters from each other with an on-board antenna. They also have all necessary digital and analog inputs/outputs, besides a serial interface and five counter/ timers. A set of proprietary AT commands facilitates their software development.

The general-purpose interface circuit was designed with the following premises:

• Should be powered by a 12Vdc external source or a 4.2Vdc internal lithium-ion battery;

• Should have serial communication with either EIA or TTL levels;

• Should provide four analog single ended inputs with fixed gain individually adjustable;

• Should have a switchable 12Vdc output capable to supply the power requirements of the selected sensors and actuators devices;

• Should provide I/O pins and power supply lines in a connector for a secondary interface board.

The block diagram of the achieved circuit can be seen in Figure 1. Two light emitter diodes (LEDs) inform the system operation mode. The 3.3Vdc regulator is a low dropout and low quiescent current circuit since the internal battery mode is supposed to be low power giving long battery life operation. This basic circuit was used to interface all devices listed in Table 1 but the latching Solenoid Valves. For those valves it was developed a secondary board with H bridge circuits to provide the direct and reverse pulses for up to four solenoids.

Figure 2 shows the final assembly for the GMM222 CO_2 probe adapted as a wireless sensor. A 12Vdc lead-acid battery associated to a photovoltaic panel was used as power supply due to the relatively high power requirements of the probe itself. The remaining devices had similar construction.

The network coordinator is an USB ZigBee interface, also purchased from Telegesis Inc. This USB stick has been used with the Windows 7 and Ubuntu Linux version 10.4 operating systems through, respectively, the Telegesis Terminal and the minicom terminal, to send the AT command lines directly to the devices. In this way, basic tests have been conducted to: check, list or modify the sensor network, switch the devices power; perform serial communication direct with the devices (through a data mode); acquire analog and digital signals; and switch the valves. Based on the set of the most useful AT commands, a program has been written in the LabView 8.2 graphical development environment to perform data collection and run the control algorithm for the CO_2 injection (Figure 3).

TABLE 1. LIST OF SENSORS AND ACTUATORS AND THEIR FEATURES

Device	**Operation method**	**Signal interface / Protocol**	**Power requirements**	**Response time**	**Supplier**	**Model / Comments**
CO_2 Sensor 1	IRGA	Serial RS-232/ASCII or analog (0-2.5V)	12 Vdc (11 to 36) / 1 W (max. 3.5 W)	2 s (no filter, no average)	Vaisala	GMP343 / Difusion probe
CO_2 Sensor 2	IRGA	Serial TTL/ASCII or analog (0-2.5V)	12 Vdc (11 to 20) / 2.5 W	20 s	Vaisala	GMM222 / OEM / Difusion probe
Anemometer	Ultrasound	Serial RS-232/ASCII	12 Vdc (5 to 32) / 36mW (with no heating)	0.25 s	Vaisala	WXT 520
Air Temperature	Capacitive			Immediate		
Air Humidity	Capacitive			Immediate		
Rain	Piezoelectric			Immediate		
Barometric Pressure	Capacitive			Immediate		
Solar radiation	Silicon photodiode	Analog (mV)	None	Immediate	Li-cor	LI-90 (Quantum) and LI-200 (Pyranometer)
Flow Controller	Differential precision temperature sensor windings	Serial RS-232/ASCII or analog (0-5V)	12 Vdc / 9.6W	2 s	Aalborg	GFC 17 with optional RS-232
Solenoid Valve	Latching	Direct and reverse pulses	12 Vdc / 24W (100ms pulses)	Immediate	Jefferson	BA222-70

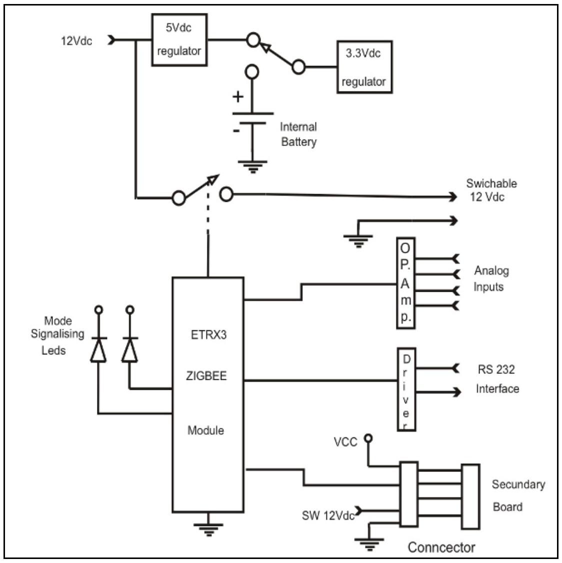


Figure 1. Block diagram for the circuit used to implement the wireless sensor network nodes.

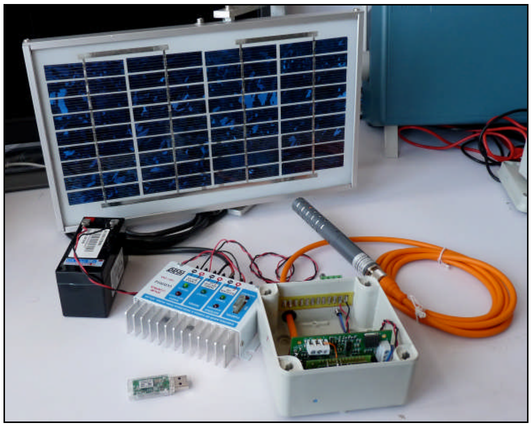


Figure 2. The OTCs CO_2 probe adapted as a ZigBee device powered by a 12Vdc lead-acid battery associated to a photovoltaic panel.

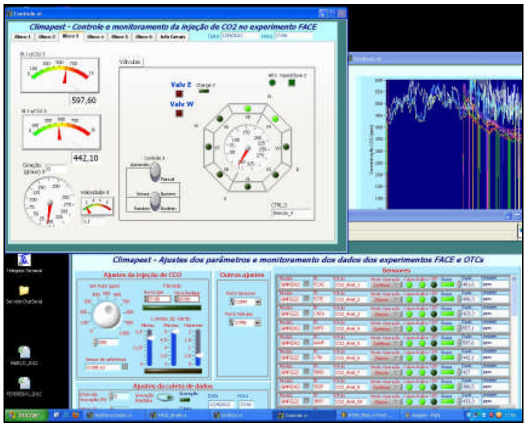

Climapest - Controle e monitoramento da injeção de CO2 no experimento FACE
Climapest - Ajustes dos parâmetros e monitoramento dos dados dos experimentos FACE e OTCs

Figure 3. Screen shot from the system program showing the control window, the configuration and data collecting window and the data-plotting window.

III. RESULTS AND DISCUSSION

Six OTCs experiments and one FACE facility have been established with the obtained Wireless Devices. Only the operational aspects restricted to the FACE facility are presented here. The OTCs have similar results from the operational point of view. Figure 4 shows one FACE plot of twelve, with the adapted sensors and valves. The experiment has been running with a constant gas flow of 60 l/min. The injection occurs only from two octagon sections, the ones positioned against to the wind direction. If the wind direction predominates towards just one section, mostly at a right angle to it, then the control algorithm alternates the injection from its left and right neighbors sections in a cycle of 5 seconds each, besides injecting all the time from the central section. In order to save money in the CO_2 consumption it was decided to run the injection only during the day, through the period from 7:00 to 17:00 hours. Therefore, during the night data have been collected only from the plots with injection to make sure there is no gas leak and to have data from the environment concentration. The system has been adjusted to allow injection only for wind speeds in the range from 0.2 to 4.0 m/s. There is yet no variation in the flow to compensate for wind speed. By the time the flow controllers were purchased, the specifications have been mistaken and they have not yet been replaced. Figure 5 shows the graphs of data collected during an arbitrary regular day of operation. It can be observed higher concentrations during the early morning period, due to calm wind conditions, and lower concentrations in the afternoon, due to heavy wind conditions during that period.

The most important practical observation is that the wireless instrumentation offers no significant time delay for the system control, allowing to follow the changes in wind direction in about a second. That is excellent considering that a lag time of up to 30 seconds has been reported for this application [7]. The experiment has been running for seven months now, what can be considered a middle-term evaluation. As expected, lightning has not been an issue. On the other hand, the network has often hung up. The problem has been identified as some ZigBee modules getting stuck in the data mode. This mode is used for direct serial data communication with most devices, and despite the correct sequence of the Telegesis AT commands supplied to open and close this mode, the transmission has eventually failed. A self-recovery solution has come up by a timeout function implemented in the ZigBee modules to leave itself the data mode. This function is also available among the Telegesis AT commands. This operational evaluation has included the system software, which has been improved a lot since the first day of operation in August 25th, 2011, with monthly updates.

V. CONCLUSION AND FUTURE WORK

In the present article, we have described the design and some operational aspects of the implementation of the FACE project in Brazil. The implementation was innovated with a wireless sensor network approach. It has been shown that the wireless instrumentation poses no significant time delay for the system control

Future works and challenges include: 1) to utilize the correct flow controller to compensate the effects of the wind speed; 2) to improve the injection control algorithm, reducing the operating costs; 3) to make the fumigation as uniform as possible inside the plot and avoiding the contamination of neighbor plots; all that making use of the advantages of wireless sensor network.

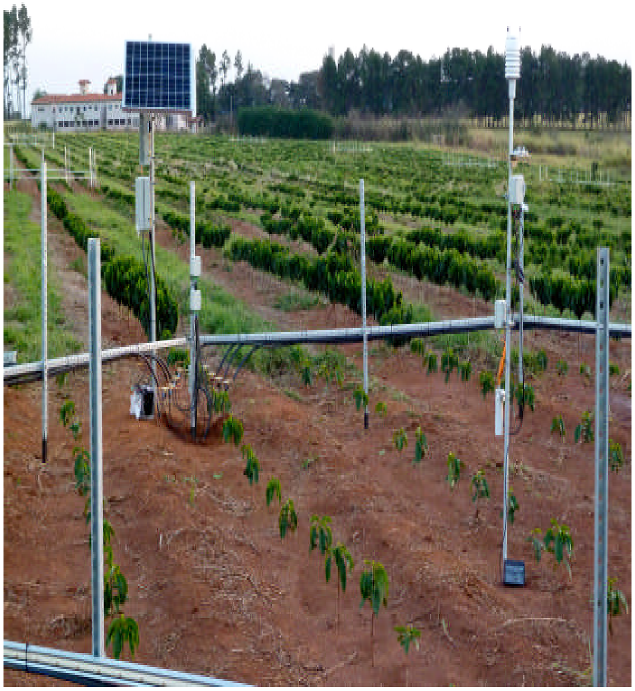


Figure 4. Partial view of one FACE plot showing the adapted devices: 1) one CO2 probe and one weather station at the plot center and 2) one flow controller and eight valves, one for each octagon section, positioned at the south border. The general- purpose circuit is inside the small plastic boxes. A 65 Watts photovoltaic panel has been used to supply power for each plot.

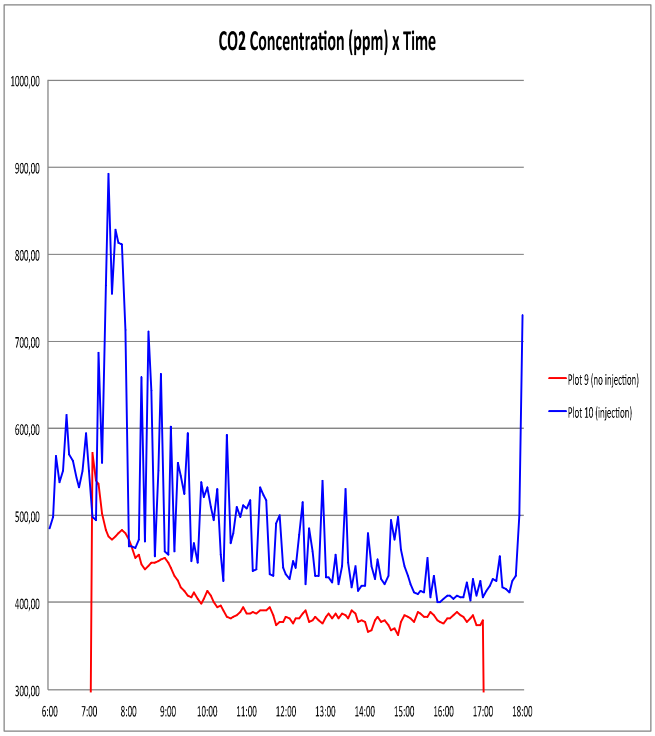


Figure 5. Data collected during one arbitrary day of operation at the center of two adjacent plots, one with and other without CO$_2$ injection

ACKNOWLEDGEMENTS

To the funding agencies:
- The Brazilian Agricultural Research Corporation, Embrapa, Macroprogram project “Impacts of climate change on plant diseases, pests and weeds”: http://www.macroprograma1.cnptia.embrapa.br/climapest
- Latin America Science & Technology Development Programme, CYTED project “Redes de sensores y microsistemas para el control del impacto de la producción agricola y la mineria en los acuiferos (REDSENSA)“
- National Council for Scientific and Technological Development, CNPq and The State of São Paulo Research Foundation,FAPESP in Brazil– project INCT NAMITEC: http://namitec.cti.gov.br/
To the people:
The technicians at Embrapa, especially Mr. Gilmar Victorino, Mr. Luis Aparecido de Godoy and the assistant Mrs. Anamaria Ferreira Mayer Dentzien

REFERENCES

[1] Intergovernmental Panel on Climate Change, IPCC Web Page: http://www.ipcc.ch/index.htm Access: oct. 2010.

[2] Raquel Ghini and Emilia Hamada, “Climate Changes: Impacts on Plants and Deseases in Brazil / Mudanças Climáticas: Impacto sobre Doenças e Plantas no Brasil”, Embrapa, first edition, Jaguariúna, SP, Brazil, 2008.

[3] G.R. Hendrey and F. Miglieta, “FACE Technology: past, present and Future” in: Nosberger J, Long SP, Norby RJ, Stitt M, Hendrey GR, Blum H (eds) “Managed ecosystem and CO2”, Springer, Germany, vol. 187, 2006.

[4] Keith F. Lewin, John Nagy, W. Robert Nettles, David M. Cooley and Alistair Rogers, “Comparison of gas use efficiency and treatment uniformity in a forest ecosystem exposed to elevated [CO_2] using pure and prediluted free-air CO2 enrichment technology” in Global Change Biology, n.15, 2009, pp. 388-395

[5] N. Wang, N. Zhang and M. Wang, “Wireless sensors in agriculture and food industry-Recent development and future perspective” in Computer and Electronics in Agriculture, n. 50, 2006, pp. 1-14

[6] Andre Torre-Neto, R. Ferrarezi, D. Razera, E. Speranza, W.C. Lopes, T.P. Lima, L. M. Rabello and C.M.P. Vaz, “Wireless sensor network for variable rate irrigation in Citrus.” Proc. 7th Information & Technology for Sustainable Fruit and Vegetable Production, Montpellier, France, v. 1, 2005., pp. 18-118

[7] Franco Miglieta, Alessandro Peressotti, Francesco Primo Vaccari, Alessandro Zaldi, Paolo deAngelis and Giuseppe Scarascia-Mugnozza “Free-air CO_2 erichment (FACE) of a poplar plantation: the POPFACE fumigation system” in New Phytologist, no. 150, 2001, pp. 465-476

ISBN: 978-1-61208-207-3

Generic Control Architecture for Heterogeneous Building Automation Applications

Armin Veichtlbauer, Thomas Pfeiffenberger, Ulrich Schrittesser
Advanced Networking Center
Salzburg Research Forschungsgesellschaft m.b.H.
Salzburg, Austria
{firstname.lastname}@salzburgresearch.at

***Abstract*—In home automation systems, and even more in building automation systems, the interoperability of installations from different vendors constitutes a significant problem for planners, construction companies and users. A generic communication infrastructure on All-IP basis, which can be used by several building automation applications like lighting, heating, ventilation, air conditioning (HVAC), access control, evacuation support, etc., can help to reduce costs during the whole building lifecycle. These applications perform control tasks with distributed sensors and actuators; many of these tasks are highly safety and security relevant. During the project Robust Facility Communication (ROFCO) we explored the requirements of a generic, but robust communication infrastructure in a building automation environment, designed and implemented a prototypical solution, and conducted a validation trial at the site of our project partner Techno-Z Salzburg.**

***Index Terms*—Control Systems; Building Automation; Supervisory Control and Data Acquisition Systems; Generic Infrastructures.**

I. Introduction

Home automation (HA) and building automation (BA) systems usually consist of a variety of different sensors and actuators (field level) as well as control devices (automation level), which are interconnected via several field bus technologies, like European Installationi Bus (EIB), Modbus, Local Operating Network (LON), Digital Addressable Lighting Interface (DALI), etc. Alternatively, radio or powerline communication may be used to reduce mounting costs, especially for older surroundings. The management level, if existing, supervises and controls the automation tasks; in many cases this is realised via web-based services in order to allow a remote control of the automation applications [1].

The market for HA and BA solutions has been rapidly growing in recent years; yet in most cases buildings are not equipped with an integrative solution from a system provider, but with individual solutions for different building automation applications [2]. The lack of interoperability of these heterogeneous solutions prevents the shared use of existing equipment, e.g., information from access control systems (like the number of persons in certain parts of a building) could be a valuable input for evacuation support systems in cases of danger, but is usually not accessible due to the proprietary nature of both solutions.

Our approach to overcome these drawbacks was to use open protocols and generic standards at every communication layer and at every level of the automation pyramid. Basically we intended to integrate different applications and different infrastructures via a convergence layer on All-IP basis, which we referred to as “X-Model”. Yet it was quite obvious that some additional functions have to be added to a working solution.

For instance, by ensuring interoperability in the way that applications should have access to the whole network and sensor/actuator infrastructure, the danger of potential misuse arises; this implicates the necessity to define appropriate security means in order to avoid damages. Thus the main goals of the generic architecture, which we have developed during the funded project “ROFCO” [3], is to ensure dependability, i.e., robustness, reliability, availability, safety and security [4].

Based on the requirements of a distributed heterogeneous BA system we defined three layers for for our generic ROFCO architecture, as shown in Fig. 1 [5]:

- An infrastructure layer, which embodies all the necessary networking functionalities for our control architecture
- A middleware layer, which provides appropriate robustness, reliability, availability, safety and security means on an end-to-end basis
- An application layer, which is responsible for the distributed control tasks of the applications using the ROFCO architecture

The application layer comprises several control logics (e.g., implemented with Programmable logic controller (PLC) or Direct Digital Control (DDCs)) at automation level as well as supervisory tools for end users (so called “SCADA” systems) at the management level.

In this paper, we concentrate on the infrastructure and middleware layers of this generic BA control architecture. We start with an overview of the scientific state-of-the-art and an assessment of existing market-ready solutions, in Section II. In Section III, we describe the communication infrastructure and the testbed we set up at the Techno-Z in Salzburg. This section is followed by Section IV, a description of the protocol functionalities we used in order to realise a robust and reliable communication system, and of the safety and security means we integrated in the prototype [6], constituting the ROFCO middleware. The subsequent section explicates the validation trials we conducted with our prototype, based on the use cases we defined for blinds and lighting control. We conclude

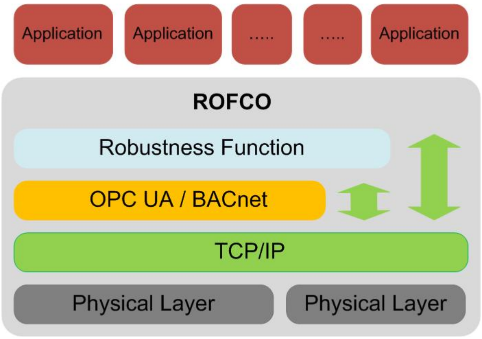


Fig. 1. ROFCO Architectural Concept [5]

with the findings derived from the conducted system trials in Section VI and an outlook in Section VII to potential follow-up research topics and exploitation activities.

II. Related Work

The heterogeneity of BA solutions has been identified as a potential barrier for BA technologies since about the turn of the millennium [7] [8]. Big vendors may offer integrative solutions, e.g., “Total Building Solutions” from Siemens [9] or “Raumtalk” from ABB [10], yet based on proprietary communication and control technologies.

Several research teams have tried to overcome this barrier by proposing interoperability features for BA systems, e.g., via gateways between field bus technologies [7], or by providing complete BA architectures for interoperable BA applications [1] [11]. For communication infrastructures, the idea of using the IP standard is not new [8].

A fully integrated approach however requires solutions for the whole automation pyramid, i.e., on every level of the control process: setting and getting values at field level, performing a control task at automation level, and supervising this at management level. A standardised middleware for that purpose needs to provide more than just IP communication; especially a generic modelling of BA objects and variables is inevitable.

For that purpose, the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) defined the BACnet standard [12]. With BACnet, complete BA environments could be built based on one generic technology [13]; yet in reality this approach has several drawbacks:

- The calculatory power required by the BACnet protocol suite is rather high, thus many field layer devices are not able to implement the BACnet stack, i.e., these devices have to be integrated via gateways.
- The support of the very common IP protocol is weak, as it is not part of the native BACnet stack. A work around named BACnet-IP is provided, i.e., basically a tunneling of BACnet messages through an IP network.
- State-of-the-art network management concepts like QoS (Quality of Service) are not supported with BACnet, which is especially critical with the use of safety or security relevant control applications (e.g., evacuation support) [14], as they require very high dependability standards, especially concerning availability of communication infrastructure.

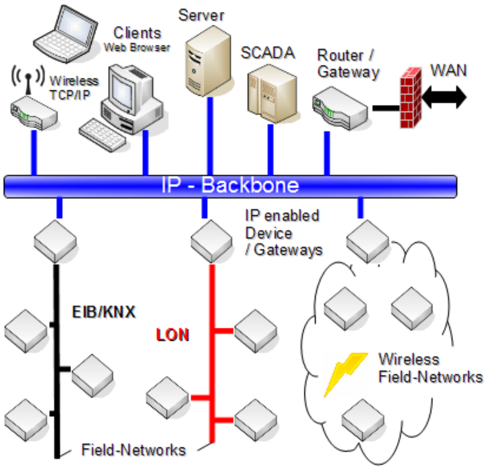


Fig. 2. ROFCO Network Infrastructure

The definition of the OPC-UA [15] standard, which is already commonly used for the control of industrial production [16], may help to overcome these shortages. By using OPC-UA in combination with TCP as transport protocol we can integrate IP networks and all the QoS mechanisms existing for the TCP/IP protocol stack. Some academic implementations of OPC-UA for BA systems are already existing, e.g., the solutions of the TU Vienna [17]. Yet the requirements for end systems still are rather high, resulting in the necessity to provide gateways to legacy systems containing older devices with not sufficient calculatory power.

There are some further research activities in the area of BA systems. These include topics as safety and security [18], control strategies and technologies [19], as well as performance issues [20]. Especially the safety and security topics are of notable interest in order to produce saleable solutions, as open systems always are always prone to outages [21] in consequence of improper use or even planned attacks.

III. Infrastructure

The goal of our work was to create a generic BA architecture which allows for easy integration of dependability, i.e., providing generic interfaces for different BA applications, including visualisation and supervisory control.

For that purpose we defined an IP based network backbone (Fig. 2), which connects all legacy components via gateways. SCADA systems however, e.g., “Zenon” from our project partner Copa-Data, can be integrated natively, i.e., as a part of the ROFCO robustness domain. This is realised by providing an open software interface containing IP sockets. Due to this openness several SCADA manufacturers may share different end devices and data servers; thus our solution provides a holistic concept to integrate global dependability means, opposed to currently available island solutions.

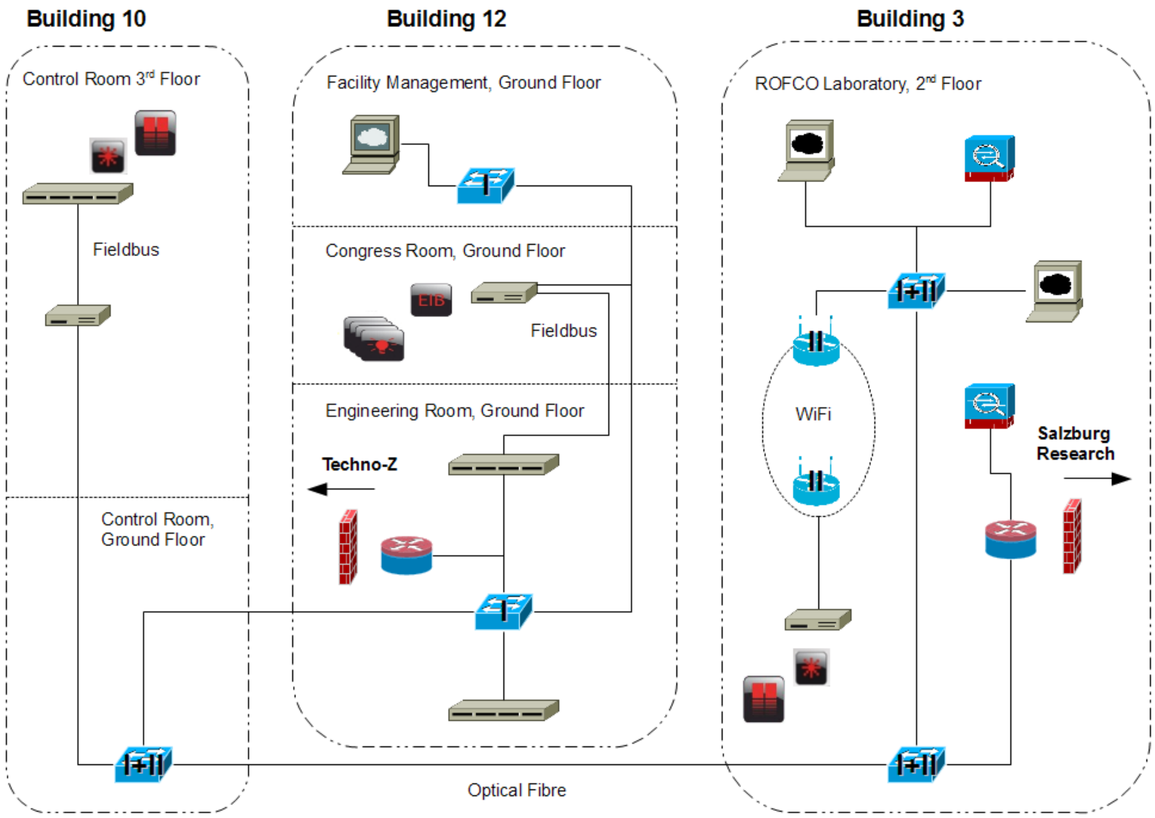

Building 10
Building 12
Building 3
Control Room 3rd Floor
Fieldbus
Control Room, Ground Floor
Facility Management, Ground Floor
Congress Room, Ground Floor
EIB
Fieldbus
Engineering Room, Ground Floor
Techno-Z
ROFCO Laboratory, 2nd Floor
WiFi
Salzburg Research
Optical Fibre

Fig. 3. ROFCO Testbed

A. Testbed Network

In order to test the feasibility of our concept, we built a prototypical solution, which we tested at the site of the project partner Techno-Z Salzburg.

Fig. 3 shows the network topology of the ROFCO testbed, which expanded over three buildings (3, 10, 12) at the Techno-Z. It was basically composed of two class C IP subnets:

- Subnet I is the management subnet of the Techno-Z used in Building 10 and 12
- Subnet II the control subnet from the ROFCO laboratory at Building 3

In both subnets we used switches with two redundant GBIC ports, thus connecting both subnets with redundant fiber connections between Building 3 and Building 10. A third switch in the ROFCO laboratory builds the interface to the various ROFCO servers. As part of the robustness concept these (manageable) switches are configured with the spanning tree (STP) mechanism. Due to the ROFCO security concept two Virtual Local Area Networks (VLAN) are configured on these three main switches, i.e., the devices connected to these switches can be run in both VLANs.

Both subnets are connected with respective company networks (Techno-Z and Salzburg Research) via a router/firewall combination. For further security issues an internal sniffer was installed to monitor the traffic inside the control and management subnets. Both functionalities, along with an intrusion detection system, can be performed by using the “MF-Security-Gateway” from the project partner Underground8.

B. Testbed Components

Each building at the Techno-Z Salzburg is equipped with different BA systems, e.g., a Somfy system to control blinds and a Sauter system to control the lighting and all HVAC components via EIB/KNX. In the following, we will describe those components which we have researched as part of the ROFCO testbed.

- *Somfy Control, Building 10*
  To control the blinds of the Buildings 10 to 15, the Somfy blind control is separated into three zones. In zone one, a single Somfy control system at the 3rd Floor regulates the whole blinds for Building 10. At this place a controller of our project partner cTrixs was installed, which serves as gateway between the blind circuit (over relay control and digital I/Os) and the Ethernet wiring.
- *Facility Management Room, Building 12*
  For managing the BA systems for the Techno-Z complex,

a control computer is situated in the facility management room in Building 12 on the ground floor. On this computer e.g., the Sauter BA system (which includes the HVAC capabilities) or the Designa access control system are visualized. Also the central fire indicating equipment is located in this room.

- *Engineering Room, Building 12*
  The Sauter BA system, the EIB lighting system and the central switch are located in the engineering room at the ground floor in Building 12. The entire building is wired from this switch. For the ROFCO network a port on the central switch was reserved and activated. There is also the possibility to configure VLANs on this Catalyst 2950 switch. A second cTrixs controller provides the interface to the EIB lighting in the congress room in Building 12; it is connected to the central switch and to the EIB bus to control the lights at the ground floor.
- *ROFCO Laboratory, Building 3*
  The laboratory is equipped with a cTrixs Application Server (CAPS) and a Zenon machine with master/backup function. On the Zenon Display the use cases we considered in ROFCO (lighting and blind control) can be visualised and controlled. The CAPS is used as a central server for the cTrixs controllers.

At the ground floor in Building 12, the lighting is not fully represented in the current building management. Thus the lighting data points and also the blinds functions in the ROFCO Showcase are implemented and visualised on the CAPS and Zenon surfaces. In Building 3, the blinds are handled by an IP-enabled cTrixs controller, but in opposition to the solution in Building 10, the connection is done directly via analog outputs and relays, and not via EIB. A Wireless Local Area Network (WLAN) bridge has been installed to transmit data to the controller.

IV. Middleware

The ROFCO Middleware layer (RML) is used to establish a dependable end-to-end communication between different entities (Fig. 1). It supports the independent distribution of control information between different end systems.

A. Dependability Requirements

A main requirement of the ROFCO system is to use commercial off-the-shelf (COTS) hardware. As the ROFCO hardware must support high reliability and calculable availability, the mean time between failure (MTBF) and the mean time to repair (MTTR) metrics of each hardware device must be known in order to derive the overall availability of the ROFCO system. For authentication and authorisation well established mechanisms have to be used, such as ITU-T X.501 [22] or IETF RADIUS/DIAMETER [23] [24]. Encryption is a further main requirement to establish a secure connection over a distributed heterogeneous communication system. For the underlying network functionalities classical network devices like CISCO switches and routers are used. Address management and routing are based on IP [25], routing metrics [26] must be supported.

B. ROFCO Entities and Roles

After determining the requirements for our prototype we compared potential technologies for our intended solution and decided for the use of OPC-UA as generic communication and management protocol. Using the free OPC-UA stack from the OPC foundation [27] we implemented the basic functionalities prototypically. OPC-UA can be used as a good base to create a generic control architecture, yet in order to integrate the intended dependability means, we had to define functionalities, which go beyond an ordinary OPC-UA implementation. The entities we defined for that purpose and their specific tasks in the ROFCO system are listed in the following:

- Client
- Server
- Registrar
- Mediator

The Client communicates and exchanges information with the Server. To be part of the installed ROFCO system the Client and all defined parts must register at the Registrar. To communicate with a non-ROFCO entity or device the Mediator maps the information between ROFCO entities and non-ROFCO entities. The Server supports the possibility to present the information in OPC-UA style. The Registrar provides interfaces for authentication and authorisation to the ROFCO system. To integrate QoS, service classes are defined for the different requirements of the supported applications.

C. Registration and Authentication Process

ROFCO devices, such as sensors, actuators, controllers, etc. must register to a ROFCO Registrar. This is necessary to exchange session keys and validate user certificates. Each ROFCO device sends its valid user identification to the corresponding ROFCO device. The corresponding ROFCO device can verify the received user certificate. If the ROFCO device does not trust the user it can check the certificate by sending it to the Registrar. Within the registration process the access levels of ROFCO devices are managed. In the ROFCO show case some ROFCO devices have limited access level to some resources. Fig. 4 shows a scheme of the registration process, which has to be performed by all devices taking part in the ROFCO system.

D. Quality Assurance

To identify potential failures in the design and the application life cycle in the whole ROFCO system a procedure called failure modes and effects analysis (FMEA) has been used. The FMEA gives an overview about which parts of the ROFCO system have the most important impacts on failure.

The ROFCO system supports the dependable operation of a communication infrastructure. To detect misbehaviour of the end systems, keep alive messages are sent during normal

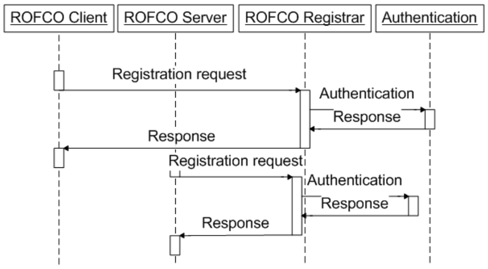


Fig. 4. ROFCO Registration Process

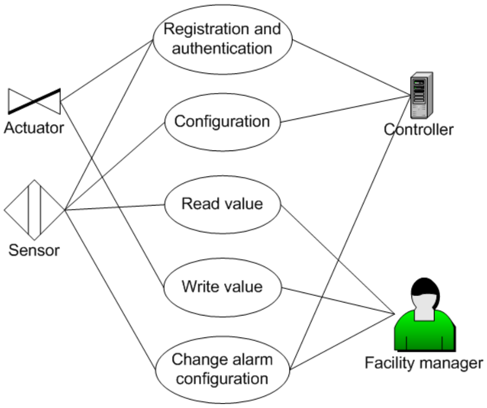


Fig. 5. ROFCO Use Case Blinds Maintenance

operation. Messages sent between different devices are signed and authenticated.

Anomaly detection is a further goal of the ROFCO architecture to find faulty messages and traffic in the system. With traffic monitoring this traffic can be detected and isolated from the system.

V. Validation

To develop a dependable system it is a basic precondition to use well established and standardised methods for verification and validation. These methods are based on several different standards, e.g., IEC 61508 [28]. In this paper we concentrate on the validation steps of the ROFCO project. The validation strategy is based on pre-defined use cases. During the course of the project these use cases were adapted to needs and requirements. Thus we have achieved an iterative product life cycle process during the project lifetime in order to enhance the quality of the ROFCO architecture. The requirement engineering process and the product life cycle process are based on standards [29].

As an example for the whole validation mechanism in ROFCO, Fig. 5 shows the use case of the maintenance sequence of sun-blinds. User stories have been used to describe the use case in such a way, that all stakeholders could understand the requirements and the interaction with the ROFCO system. For requirement gathering the verbal description of the use case and the discussion with the stakeholders improved the understanding for the developers.

For validating the ROFCO system different steps were defined. Like in an agile software development process, each single use case had to be validated. Based on the verbal description and the UML Use Case Diagram of each use case we defined the respective tests. Each test definition had some attributes, such as test description, pre-conditions and post-condition, as defined in [30]. The whole ROFCO system, as described above, was validated in the ROFCO validation trial. All involved stakeholders and project partners have prepared the defined use cases to validate the ROFCO system. During some pre-tests, some misconfigurations in the controller setup could be identified and fixed. The validation trial showed the interworking of a heterogeneous building automation system as expected.

VI. Conclusion

As a result of our validation trial we proved the feasibility of our approach, as we were able to access the control devices using different OPC-UA clients. We were able to implement getter and setter functions for the data points in different building units (lighting, blinds). Furthermore, we developed a robustness concept based on availability calculations according to IEC 61508 [28] functional safety standard and assessed the system relevant risks with an FMEA.

A possible barrier for a wide adoption of our approach in future commercial solutions are the relatively high requirements on the used devices. In order to be able to proceed all the session and rights management data as well as the OPC stack the devices need a certain minimum of calculatory power; for practical reasons this can not be guaranteed in all cases. Here this can be counteracted by the use of gateways to those legacy systems, which are not able to implement a native OPC connection, yet this limits the beneficiaries of our system to a more narrow system border. However, future developments have to be observed accurately, as the progress of calculatory power in embedded devices may make this drawback obsolete in a few years.

VII. Further Work

For safety and security relevant applications like evacuation support, not only principle concepts need to be shown, but solutions have to be provided which meet certification requirements. In parts such standards are existing (e.g., for certification of evacuation systems) but the whole process of installing and maintaining different applications in a building environment is not standardised as such, i.e., changing the set up of any other linked automation subsystem would enforce another accreditation for the security relevant application, as the environment of the security relevant subsystem has changed. This is a major obstacle to install interoperable systems and one of the reasons why integrated HA/BA solutions are still rare.

An interesting research field is emerging through the current developments in the smart grid sector. The establishment of communities which are sharing energy resources generates many questions regarding not only security, but also privacy, billing, optimisation of resource usage, and thus controlling not only single buildings but bigger unions. Especially the integration of a distributed energy control network with existing solutions in HA/BA is a considerable challenge.

Acknowledgment

The work described in this paper was conducted during the project "Robust Facility Communication" (ROFCO) funded by the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT).

References

[1] K. Charatsis, A. Kalogeras, M. Georgoudakis, J. Gialelis, and G. Papadopoulos, "Home / Building Automation Environment Architecture Enabling Interoperability, Flexibility and Reusability," in *Proceedings of the IEEE International Symposium on Industrial Electronics 2005 (ISIE 2005)*, vol. 4, Jun. 2005, pp. 1441–1446.

[2] F. Ferreira, A. Osorio, J. Calado, and C. Pedro, "Building Automation Interoperability – A Review," in *Proceedings of the 17th International Conference on Systems, Signals and Image Processing (IWSSIP 2010)*, 2010, pp. 158–161.

[3] Salzburg Research Forschungsgesellschaft. (2012) ROFCO – Robust Facility Communication. Accessed: 2012-06-19. [Online]. Available: http://www.salzburgresearch.at/en/projekt/rofco_en/

[4] G. Panholzer, A. Veichtlbauer, P. Dorfinger, and U. Schrittesser, "Simulation of a Robust Communication Protocol for Sensor Data Acquisition," in *Proceedings of the 6th International Conference on Wireless and Mobile Communications (ICWMC 2010)*, Valencia, Spain, Sep. 2010, pp. 145–150.

[5] A. Veichtlbauer and T. Pfeiffenberger, "Dynamic Evacuation Guidance as Safety Critical Application in Building Automation," in *Proceedings of the 6th International Conference on Critical Information Infrastructure Security (Critis 2011)*, Lucerne, Switzerland, Sep. 2011.

[6] C. Probst, "Konzeptionierung eines Benutzermanagements für den Zugriff auf vertrauliche Daten von IP fähigen Sensornetzen," May 2010, in German.

[7] J. P. Thomesse, "Fieldbuses and interoperability," *Control Engineering Practice*, vol. 7, iss. 1, pp. 81–94, Jan. 1999.

[8] E. Finch, "Is IP everywhere the way ahead for building automation?" *Facilities*, vol. 19, iss. 11/12, pp. 396–403, 2001.

[9] Siemens AG. (2011) Total Building Solutions für intelligente Gebäude – Siemens Building Technologies. Accessed: 2012-06-19. [Online]. Available: http://www.industry.siemens.de/buildingtechnologies/de/de/total-building-solutions/Seiten/total-building-solutions.aspx

[10] ABB Asea Brown Boveri Ltd. (2012) Raumtalk – Building Automation over IP. Accessed: 2012-04-11. [Online]. Available: http://www.abb.at/cawp/deabb201/24d156e58bc98443c125720b0025238d.aspx

[11] W. Granzer, W. Kastner, G. Neugschwandtner, and F. Praus, "A Modular Architecture for Building Automation Systems," in *Proceedings of the 6th IEEE International Workshop on Factory Communication Systems (WFCS '06)*, Jun. 2006, pp. 99–102.

[12] American Society of Heating, Refrigerating and Air-Conditioning Engineers Inc., "BACnet - A Data Communication Protocol for Building Automation and Control Networks," ANSI/ASHRAE Standard 135-2004, 2004.

[13] D. Snoonian, "Smart buildings," *Spectrum, IEEE*, vol. 40, pp. 18–23, Aug. 2003.

[14] U. Schrittesser, "Synthese von redundanten vermaschten WLAN," Jun. 2008, in German.

[15] CAS. (2010) OPC Unified Architecture. Accessed: 2012-06-19. [Online]. Available: http://www.commsvr.com/UAModelDesigner/Index.aspx

[16] W. Mahnke, S.-H. Leitner, and M. Damm, *OPC Unified Architecture*. Springer-Verlag Berlin Heidelberg, 2009.

[17] A. Fernbach, W. Granzer, and W. Kastner, "Interoperability at the Management Level of Building Automation Systems: A Case Study for BACnet and OPC UA," in *Proceedings of the 16th IEEE Conference on Emerging Technologies and Factory Automation (ETFA '11*, Sep. 2011.

[18] W. Granzer, W. Kastner, G. Neugschwandtner, and F. Praus, "Security in Networked Building Automation Systems," in *Proceedings of the 6th IEEE International Workshop on Factory Communication Systems (WFCS '06)*, Jun. 2006, pp. 283–292.

[19] T. I. Salsbury, "A Survey of Control Technologies in the Building Automation Industry," in *Proceedings of the 16th IFAC World Congress*, vol. 16, part 1, Prague, Czech Republic, Jul. 2005.

[20] S. Makarechi and R. Kangari, "Research Methodology for Building Automation Performance Index," *International Journal of Facility Management*, vol. 2, no. 1, 2011.

[21] C. Probst and A. Veichtlbauer, "Security Features of a Generic Sensor Data Acquisition System," in *Proceedings of the 6th International Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet 2010)*, Bodrum, Turkey, Oct. 2010.

[22] International Telecommunication Union. (2008) X.501. Accessed: 2012-06-19. [Online]. Available: http://www.itu.int/rec/T-REC-X.501

[23] C. Rigney, A. C. Rubens, W. A. Simpson, and S. Willens, "Remote Authentication Dial In User Service (RADIUS)," IETF RFC 2865, Jun. 2000.

[24] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "Diameter Base Protocol," IETF RFC 3588, Sep. 2003.

[25] J. Postel, "Internet Protocol," IETF RFC 791, Sep. 1981.

[26] R. Baumann, S. Heimlicher, M. Strasser, and A. Weibel. (2007, Feb.) A survey on routing metrics. TIK Report 262. Accessed: 2012-06-19. [Online]. Available: http://www.baumann.info/public/tik262.pdf

[27] OPC Foundation. (2012) OPC – The Interoperability Standard for Industrial Automation & Other. Accessed: 2012-06-19. [Online]. Available: http://www.opcfoundation.org

[28] International Electrotechnical Commission (IEC). (2012) Functional safety and IEC 61508. Accessed: 2012-06-19. [Online]. Available: http://www.iec.ch/functionalsafety

[29] "Systems and software engineering - software life cycle processes," Tech. Rep., 2008, Accessed: 2012-06-19. [Online]. Available: http://dx.doi.org/10.1109/IEEESTD.2008.4475826

[30] "IEEE standard for software test documentation," Tech. Rep., 1998, Accessed: 2012-06-19. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=741968

ISBN: 978-1-61208-207-3

A Survey of Deterministic Vs. Non-Deterministic Node Placement Schemes in WSNs

Luhutyit Peter Damuut
Computer Science and Electrical
Engineering Department,
University of Essex,
CO4 3SQ, England.
Email: pdamuu@essex.ac.uk

Dongbing Gu
Computer Science and
Electrical Engineering Department,
University of Essex,
CO4 3SQ, England.
Email: dgu@essex.ac.uk

Abstract—**In Wireless Sensor Networks (WSNs), the node's position determines the functionality, life span and the efficiency of the network. The choice of the deployment strategy is crucial in most mission critical application areas. This paper examines the issues surrounding the choice of sensor placement schemes with respect to the application areas, type of sensors and the operational environment. Simulation results, based on hierarchical data clustering algorithm , reveal the effect of both deterministic and non-deterministic sensor placement strategies on the lifespan of a network formed using homogeneous sensors. The results corroborate the widely held view that deterministic sensor placement schemes usually outperforms non-deterministic methods, due to the higher level of control available to the network designer in the former than in the later approaches.**

Keywords- sensor; deterministic; non-deterministic; deployment.

I. INTRODUCTION

Sensor nodes deployed with the intention of being operated autonomously in unattended environments like an oil pipeline running through hundreds or even thousands of miles presents a non-trivial challenge. In many configurations, it is normally envisioned that the wireless sensor networks (WSNs) should consist of hundreds or thousands of nodes, each operating on a small battery that stops working whenever it runs out of energy [1]. The WSN could fail to function should a significant number of those sensors exhaust their on-board energy supply. In certain applications and deployment schemes, failure of the critical node could result in the termination of the network's life [2]. Therefore, it is proper to carefully plan, design and manage WSNs in order to meet the application's requirements such as energy conservation which helps to prolong the overall lifespan of the network.

The choice of a sensor deployment scheme is often affected by the type of sensors, the application and the operational environment of the sensors [3]. The need to exercise control over node deployment is governed by the monetary and operational costs of the nodes and when their position in the network significantly affect their operation.

There are many perspectives under which the sensor placement problem could be viewed. The common ones include nodes function in the network, the optimization objective and the deployment methodology [4]. From the deployment point of view, we could classify the sensor placement problem into two namely, non-deterministic and deterministic placements.

Non-deterministic sensor placement is often referred to as random placement, while deterministic placement is often called controlled placement in some texts. In this paper, we would refer to random sensor placement as non-deterministic placement while controlled placement would be referred to as deterministic placement.

In Deterministic Sensor Placement Schemes (DSPS) [5], the nodes are placed in order to meet some desired performance objectives. For example, the coverage of the monitored region can be ensured through careful planning of node densities and fields of view and thus the network topology can be established at setup time. DSPS are common in certain applications like room temperature monitoring, medical applications, underwater acoustics, imaging and video sensors among others.

In many wireless sensor network applications however, the sensors are deployed randomly. In this placement scheme, there is little control over coverage and node density distribution to ensure strongly connected network topology [6]. Therefore, DSPS is often pursued for only a selected subset of the deployed nodes with the aim of structuring the network topology in a way that achieves the desired application requirements. Besides coverage, the node's positions in the WSN affect a number of network performance metrics such as energy consumption, delay and data throughput. For example, the signal strength gets attenuated with increase in distance from the transmitting node.

The remainder of the paper is organized as follows: Section II presents related work on sensor placement; Section III discusses some selected deterministic and non-deterministic sensor placement algorithms and Section IV highlights the factors that influence the choice of sensor node placement schemes. Finally, section V concludes the paper and points the way forward for our future work.

II. RELATED WORK

Sensor placement is an area that has been well researched over the years [7] [1] [6]. In indoor applications, the sensor placement problem closely resembles the art gallery problem (AGP) which is aimed at determining the minimum number of guards needed to cover the interior of an art gallery [8].

In most of the literature, the problem is often viewed as an optimization problem that aims to meet some specific sets of objective function(s) such as coverage of a well-defined physical topology; coverage of a specific target of interest; network connectivity or maximization of the network's life span as the case may be.

Sensor placement in WSN presents a serious challenge in most of the non-trivial applications. The researchers in [8], showed that sensor placement problem is NP-hard. As a result of the complexity involved, [7] [9] proposed several heuristics aimed at finding sub-optimal solutions to the problem.

The position of a sensor node in a WSN can be viewed from three different perspectives namely; the deployment methodology, the optimization objective and the role of the node in the WSN [4]. Fig. 1, summarizes the different perspectives under which a sensor node can be viewed. A node is either deployed deterministically (i.e., when the node is carefully and deliberately positioned to serve its purpose) or non-deterministically (i.e. placement with little control over the actual positioning compared to deterministic method). When viewed in terms of its optimization criteria, the node could be to maximize the network lifespan, to ensure maximum area or network coverage, maximize connectivity or it could be to minimize energy wastage and so on. Moreover, a node in the sensor field could be viewed in terms of its function in the network. Here, the node could be serving the role of a sensor, a relay node(viewed in some cases as cluster head) or a data sink (i.e., terminal point where decisions on the sensed information is taken).

These optimization strategies are however based on the assumption that these sensors maintain static positions throughout the life time of the network so that the quality of service metrics such as distance, network connectivity can be measured with relative ease.

Some researchers in this area however, advocate for dynamic adjustment of the node's location. Their argument is based on the fact that the optimality of the initial node's positions in the WSN may become void during the life time of the network ([10] [11]). This, in our opinion, is a valid point because external factors such as human activities (e.g., excavation) or environmental conditions (e.g., earth tremor), may change the initial locations of the deployed sensors. Moreover, network resources may result in changes as new nodes join the network, or as some existing nodes get exhausted and die out.

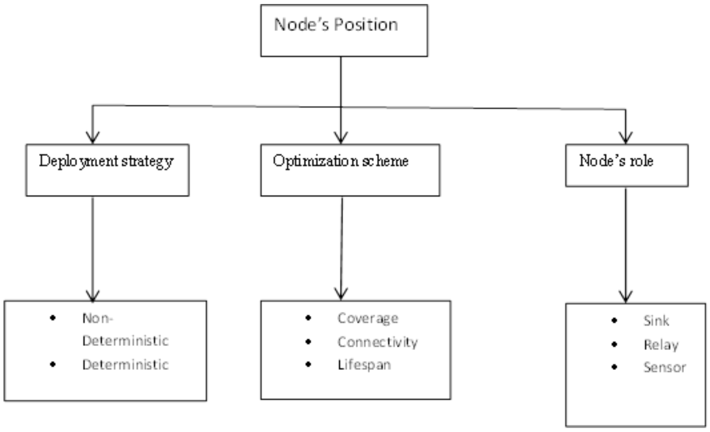


Fig. 1. Sensor Node Placement Strategies

Guo et al. [1] examined the impact of deterministic linear sensor placement on the life span of a wireless sensor network (WSN) deployed to monitor an oil pipeline using equal distance node placement scheme over different power configurations. They argued that by using only the right number of sensors, the life span of the WSN can be significantly enhanced. Any further addition to the minimal number of those sensors tends to worsen the lifespan of the network.

Shakkottai et al. [12] showed that it is possible to achieve optimal network connectivity without necessarily achieving area coverage. Their submission is based on the fact that disparity exists between the sensing and transmission ranges of a sensor . In practice these values are not always the same so, it is an important consideration that is equally related to the node deployment strategy. Examples of DSPS can be found in [13], where the sensors are used to monitor the health of buildings in order to detect corrosions and overstressed beams that can endanger the structure.

Similarly, varying the node density throughout the area of the sensor field can lead to unbalanced traffic load and hence bottlenecks. Likewise, a uniform node distribution may lead to depleting the energy of nodes that are closer to the base-station faster than those far from it, leading to shorter network lifetime [1].

III. SENSOR PLACEMENT ALGORITHMS

DSPS in WSN is concerned with careful and controlled placement of sensors around the area of interest while non-deterministic on the other hand there is little control over the deployment at the target locations. In those WSN applications that employ DSPS algorithms, the positions of the sensors can be optimized to achieve the best coverage, connectivity or to maximize network lifetime as the case may be. This sensor placement strategy is common in most indoor and industrial applications where there is reasonable level of control on the node positions. Table III below depicts some sample references to node placement

Reference	Type of coverage	Approach
[9]	Target	Objective is to minimize the number of deployed sensors in the network
[6]	Target	Algorithm is based on virtual forces (analogous to magnetic force of attraction on objects)
[14]	Area	Focus of maximizing area coverage; Algorithm is based on potential field; the number of neighbours of each sensor is required to be at least K. K, is number of communicating neighbours
[15]	Target/area	Focus on connectivity; Assumes equal communication and sensing ranges for all the sensors
[16]	Area	Considers connectivity; The algorithm works for arbitrary-shaped region and with any ratio of r_c/r_s, where r_c is communication range and r_s is sensing range

TABLE I
SOME SENSOR PLACEMENT ALGORITHMS

algorithms designed and implemented to address sensor area coverage, target coverage or both.

In [9], the algorithm uses path exposure (which is a metric), to estimate the likelihood that a target would be detected when it traverses through a sensor network. The metric is obtained by calculating the probability of detecting the target anywhere along the path. To obtain the probability, the total energy (E) that each sensor at position i (i.e., s_i) could expend when detecting a target at position u is formulated as:

$$E_i(u) = \frac{K}{||u - s_j||^k} + N_i \quad (1)$$

where K is the energy emitted at the target, k is decay constant $(2 < k < 5), ||u - sj||$ is the distance between the sensor s_i and the target and N_i is noise at s_i. The possibility of detecting a target event that occurred at position u is given by the probability that the total energy $\sum_{i=1}^{n} E(u)$ of all the n sensors, with reference to the target at u, is greater than a certain threshold η. This expression gives the probability that a target at location u can be detected by the network:

$$P_i(u) = Prob \sum_{i=1}^{n} E(u) > \eta \quad (2)$$

where $E_i(u)$ is as in equation 1 and η is the detection threshold.

The monitored area is divided into a fine grids, and every edge in the grid is assigned a weight equal to $P_i(u)$ for all the points within that segment. Dijkstra algorithm is then applied to find the path with minimum weight as the least exposed.

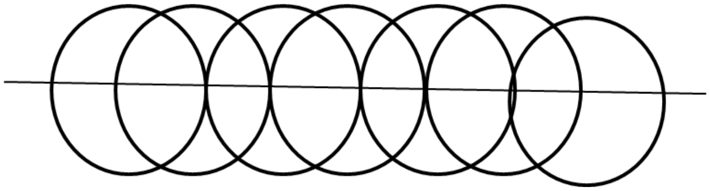


Fig. 2. Example of Sensor Placement Forming an r-strip

Zou and Chakrabarty [6], proposed the use of a virtual force and target location query. The idea is to use the virtual force to find the optimal location for a sensor after it is initially placed randomly in the sensor field. The operational principle of the virtual force is similar to that of the magnetic force which attracts opposite ends of the magnetic pole while it repels identical ends. The influence of this force on the sensors ensures coverage of the target location by moving them as far apart from their neighbor's as the force can possibly allow.

The same idea of the virtual force usage is further by extended by Poduri and Sukhatme [14], where they proposed two opposing forces F_{cover} and F_{degree}, the former being that which makes neighbouring sensors repel one another to increase area coverage, while the latter is the force of attraction among the neighbouring sensors to maintain a threshold of k connectivity (k being the minimum number of nodes required to maintain simultaneous connectivity). Consequently, the eventual position of any sensor in the network is determined by the net force (i.e., $F_{cover} + F_{degree}$) acting on that sensor. Intuitively, it is easy to see that under this scheme, a deployed sensor node would maintain its fixed position until at least one of its neighbours die out or if some external barrier or force tends to break this equilibrium. In our opinion, it is not energy efficient to use this node deployment strategy for unattended outdoor application areas for obvious reasons.

Different from the schemes in [6] and [14], the authors of [15] examine the classical case of deterministic placement of nodes in an r-strip (i.e. with equal sensing and communication radii for all the deployed sensors). Their work is aimed at properly placing the sensors to achieve connectivity, coverage and to minimize the overall number of sensor nodes. The r-strip is as shown in figure 2, where the sensors are placed side by side and the distance between any two adjacent sensors is given by r. The overlapping circular rings represent the sensing and communication radii for each sensor.

Non-deterministic sensor placement is common in such application areas as disaster recovery and forest fire detections and other mission critical applications where it is quite risky and/or infeasible to use deterministic deployment strategies.

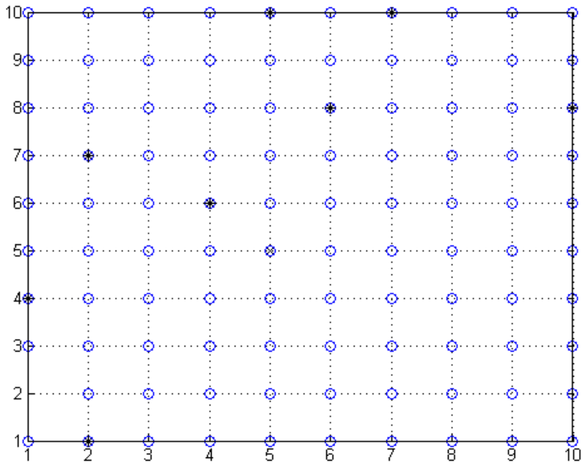


Fig. 3. DSPS Method

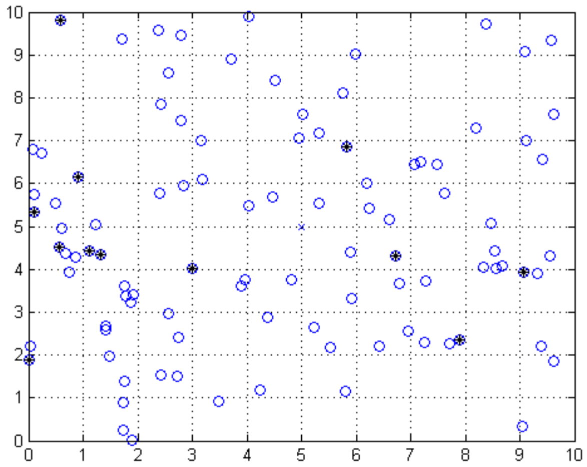


Fig. 4. Non-DSPS Method

IV. RESULTS AND DISCUSSIONS

In comparing the performance of DSPS against its non-DSPS counterpart, LEACH [17] algorithm was used with the following parameters: n = 100 (i.e., the number of nodes); the topology is given by a 10x10 2-dimensional grid and r=500 (i.e., the number of rounds for executing the algorithm). The data fusion center was situated at a fixed location within the sensor field while the nodes were deployed randomly (in the first case) and then deterministically (in the second case) respectively.

Recall that the LEACH algorithm proceeds in two phases namely, the network setup and the steady state phase. The network setup phase is comprised of the node deployment and initialization activities. Therefore, the sensor field for the DSPS method, was set up as shown in Fig. 3, where the nodes were carefully placed at evenly spaced grid points. This type of deployment is common in field surveillance applications (e.g., agriculture).

In a similar manner, the non-DSPS method was also set up as depicted in Fig. 4 with equal number of nodes on the same topology as in the case of the DSPS method.

The result in Fig. 5 shows the result of comparing the two deployment strategies in terms of the number of nodes whose energy are exhausted at each round of the algorithm. It is obvious that the DSPS method out performs against the non-DSPS method in terms of the number of dead nodes per round. This is indicative of the level of control that is inherently available to the network designer when DSPS method is used compared to non-DSPS methods.

In addition to the network performance indicators that correspondingly vary with the node deployment strategy in use, there are other important factors that govern the choice of any particular sensor node deployment scheme. Some of these factors are highlighted below:

(a.) Type of sensor:

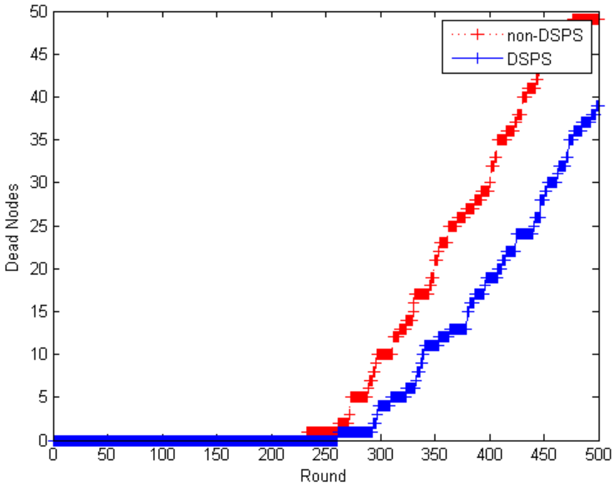


Fig. 5. Using LEACH to Compare DSPS With Non-DSPS

The type of sensor determines how it could be deployed. For example, heat sensors (thermal) cannot be deployed to measure wave amplitude (acoustic) because doing so would not yield the desired results. There are many dimensions to the type of sensor. Some of which include detection means, sensor material, size, weight, etc.

(b.) Application area:

Next to the type of sensor is the intended application area. In applications such as domestic appliances, personal health or scientific measurement, deterministic placement schemes are advisable while in military surveillance in enemy territories, forest fire detection, seismic sensing, etc., non-deterministic approaches are advisable.

(c.) Cost:

Cost is a general terms which is subject to interpretation based on the context. Here, we limit the word to the monetary cost of acquiring and maintaining the sensor node while in operation. For example, non-deterministic sensor node deployments are very common in application areas where the costs of the sensors are insignificant whereas, deterministic approaches are the norm in areas where the costs are high.

V. CONCLUSION

One substantive contribution of this paper is in corroborating the widely held view that the choice of sensor deployment strategy directly affects network performance. The paper also highlights the factors that influence the choice of node deployment strategies in wireless sensor networks. Notable among these factors include the optimization criteria of the network and the role of the node in the network. Other factors include the type of sensor, the application area and the acquisition and operational costs of the nodes.

Whenever the application requires massive number of sensors in potential target areas, and the cost of the nodes is insignificant, we opined that non-deterministic placement strategies are more practical. For example, when using a sensor network to monitor an oil pipeline running through thousands of miles or for security surveillance purposes, employing the non-deterministic deployment strategy would be highly recommended in order to meet certain acceptable performance objectives such as coverage and connectivity. This paper focuses its investigation on sensor node deployment strategies in WSNs only. It does not cover other types of nodes serving different roles (e.g., relay nodes or base station nodes) in the network.

We envision that a mix of both deterministic and nondeterministic sensor placement schemes would be the most effective and efficient placement strategy for large-scale and mission-critical WSN applications where inexpensive nodes could be deployed to serve the roles of sensors in the monitored region while few, more powerful nodes could serve the roles of data relay nodes in the network to save energy. In our future work, we plan to embark on the comparative analysis of sensor placement and data fidelity in WSN using some tested models.

ACKNOWLEDGEMENTS

This research was supported by the Petroleum Technology Development Fund (PTDF) under grant number 10PHD056 and in part by the University of Jos, Nigeria.

References

[1] Y. Guo, F. Kong, D. Zhu, A. Tosun, and Q. Deng, "Sensor placement for lifetime maximization in monitoring oil pipelines," in *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*. ACM, 2010, pp. 61–68.

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

[3] M. Abd-El-Barr, M. Youssef, and M. Al-Otaibi, "Wireless sensor networks-part i: topology and design issues," in *Electrical and Computer Engineering, 2005. Canadian Conference on*. IEEE, 2005, pp. 1165–1168.

[4] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.

[5] J. Li, L. Andrew, C. Foh, M. Zukerman, and H. Chen, "Connectivity, coverage and placement in wireless sensor networks," *Sensors*, vol. 9, no. 10, pp. 7664–7693, 2009.

[6] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2. Ieee, 2003, pp. 1293–1303.

[7] S. Dhillon and K. Chakrabarty, *Sensor placement for effective coverage and surveillance in distributed sensor networks*. IEEE, 2003, vol. 3.

[8] A. Efrat, S. Har-Peled, and J. Mitchell, "Approximation algorithms for two optimal location problems in sensor networks," in *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*. IEEE, 2005, pp. 714–723.

[9] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. Saluja, "Sensor deployment strategy for target detection," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM, 2002, pp. 42–48.

[10] K. Akkaya, M. Younis, and M. Bangad, "Sink repositioning for enhanced performance in wireless sensor networks," *Computer Networks*, vol. 49, no. 4, pp. 512–534, 2005.

[11] G. Wang, G. Cao, T. La Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4. IEEE, 2005, pp. 2302–2312.

[12] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2. IEEE, 2003, pp. 1073–1083.

[13] J. Paek, K. Chintalapudi, J. Caffrey, R. Govindan, and S. Masri, "A wireless sensor network for structural health monitoring: Performance and experience," 2005.

[14] S. Poduri and G. Sukhatme, "Constrained coverage for mobile sensor networks," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 1. IEEE, 2004, pp. 165–171.

[15] K. Kar, S. Banerjee *et al.*, "Node placement for connected coverage in sensor networks," 2003.

[16] Y. Wang, C. Hu, and Y. Tseng, "Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks," in *Wireless Internet, 2005. Proceedings. First International Conference on*. IEEE, 2005, pp. 114–121.

[17] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*. IEEE, 2000, pp. 10–pp.

Wireless Measurement Node for Dust Sensor Integration

Mokhloss I. Khadem, Grigore Stamatescu, Valentin Sgârciu
Faculty of Automatic Control and Computers
University Politehnica of Bucharest
sml_ka@yahoo.com, {*gstamatescu, vsgarciu*}*@aii.pub.ro*

***Abstract*—The presence of particulate matter of various sizes, generally classified as dust, is an important factor affecting air quality in outdoor and indoor scenarios. Many industrial processes require clean rooms where dust concentration has to be kept below certain thresholds in order to assure the success of the manufacturing process or the quality of the finished product. Applications which need precise dust quantity measurement presently use complex and expensive systems. In the mean time, wireless sensor networks (WSN) have emerged as a tool which promises higher resolution spatial measurements than conventional devices. Beside usual sensors, many WSN producers offer data acquisition expansion modules which supply general purpose analog and digital inputs allowing for the connection of external sensors. This paper proposes combining the two technologies by integration of an optical interferometer dust sensor with a conventional sensor networking platform through a data acquisition module. The system design and description as well as an experimental evaluation in a laboratory setting are presented. We evaluate the results obtained by comparison to a high precision laser air quality monitor for 0.5 and 2.5 micron particles. The results are encouraging and show that our approach is viable for testing outside the laboratory environment, in a real world deployment.**

***Keywords*-dust sensors; data acquisition interface; wireless sensor networks; embedded software;**

I. Introduction

Dust affects many sectors of human activity and it is widely present in the galaxy. Ambient radiation heats dust and re-emits radiation into the microwave band, which may distort the cosmic microwave background power spectrum. Dust in this regime has a complicated emission spectrum, and includes both thermal dust emission and spinning dust emission. Dust samples returned from outer space may provide information about conditions in the early solar system. Also dust has effects on aviation, in the past 30 years, more than 90 jet-powered commercial airplanes have encountered clouds of volcanic ash and suffered damage as a result. The increased availability of satellites and the technology to transform satellite data into useful information for operators have reduced the number of volcanic ash encounters. One of the more important and common effects of dust is the influence on environmental pollution and human health. Most industrial dusts contain particles of a wide range of sizes. The behavior, deposition and outcome of any particle after entry into the human respiratory system, and the response that it produces depends on the nature and size of the particle. Breathable dust approximates to the fraction of airborne material that penetrates to the gas exchange region of the lung. The breathable fraction varies for different individuals; however, it is possible to define a target specification for sampling instruments that approximates to the breathable fraction for the average person. Moreover there is a big effect of dust in industrial environments, especially in the case of high precision industries such as integrated circuits and nanoelectronic fabrication. In industrial plants, where combustible dusts or dust containing goods are produced, processed or stored, there is a risk of explosion. Mixtures of dust and air with concentrations above the Lower-Explosion-Limit (LEL) and below the Upper-Explosion-Limit (UEL) together with various modes of ignition (electric sparks, hot surfaces) can cause an explosion.

Networks of wireless sensor devices are being deployed to collectively monitor and disseminate information about a variety of phenomena of interest [1]. A wireless sensor device or mote is capable of sensing, limited amount of computation, signal processing and data storage and wireless communication. Advances in integrated circuits design have led to a reduction in the size, weight and cost of sensor devices, with an improvement in their resolution and accuracy. At the same time, modern wireless networking of large number of wireless capable sensor devices can work collaboratively to achieve a common objective. At its core, a WSN is meant to enable high temporal and spatial measurement of the surrounding environment. Some of the applications in which this technology has been deployed so far include: environmental monitoring, military surveillance, biosensors for health applications and smart sensors to monitor and control manufacturing facilities. Integration work with other systems is very important in order to provide a reliable tool for domain specialists interested in the provided data [2].

The rest of the paper is structured as follows. Section 2 elaborates on the theory of dust measurements, with focus on optical measurement devices and describes the context of our work through related work. Section 3 presents the system design along with the main components used to implement the wireless dust measurement node. Section 4 presents the monitoring results achieved with reference to a precision air quality monitoring device. In Section 5 we conclude the paper and highlight directions for future work.

II. Optical Dust Measurement and Related Work

The best known principles of dust measurement devices are: gravimetric, triboelectric and optical measurement. Each of these are best suited to specific application domains which differ in the intensity of dust pollution, water vapor proportion and the surface of the investigated area.

The optical dust measurement principle which we study in this paper is based on the attenuation of the intensity of a light beam by absorption and dispersion upon penetrating a cloud with solid particles. It is based on the attenuation of the intensity of a light beam by absorption and dispersion penetrating a cloud with solid particles. Figure 1 shows a diagram depicting the conversion from dust concentration to useful output using a fixed light emitter. Lambert-Beer's law [3] describes the relation between the light transmission and the dust concentration c according to the following equation:

$$I = I_0 \cdot e^{-\varepsilon \cdot c \cdot l} \tag{1}$$

where:

- I_0 = initial intensity
- I = resulting intensity of the light beam
- ε = coefficient of extinction (a specific constant accounting for dust type and application)
- l = distance
- c = dust concentration

For the light source with rectangular pulses, an efficient GaAs-Luminescent-diode with its maximum spectral sensitivity at 950 nm is used. Consequently, a photo diode of the same spectral sensitivity is used for receiving. The clock frequency should be chosen in a way that even rapid changes or momentary peak values of the dust concentration up to about 3 kHz are exactly reproduced [3].

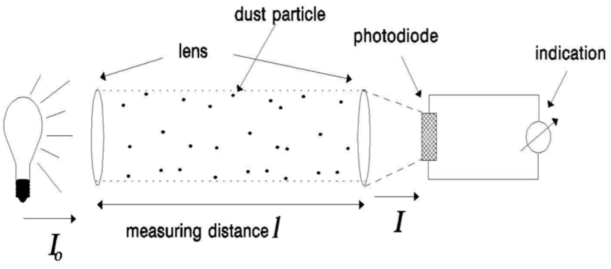


Figure 1. Optical Dust Measurement Principle Schematic

Zhang et al. [4] propose a forest fire monitoring system based on a designed ZigBee [5] wireless sensor node. The main goal is to measure smoke and humidity levels while benefiting from specific advantages of safety in data transmission, network establishement and low cost and energy requirements. The topological structure of the system is an adaptation of a cluster-tree. Compared with a reticular structure, a cluster-tree structure can be built more easily and the information path takes less memory space. At the same time, the chain structure needs to be stable and its scale is limited, which needs to be improved in future investigations. The proposed system is described as a first attempt and complement to existing forest fire monitoring and prevention methods. It provides a solid basis in terms of hardware for the application of advanced wireless sensor network technology. It is pointed out that, in order to extend the potential of the system and improve forest fire monitoring technology, the problems of energy consumption, nodes location and clock synchronization have to be addressed in the future. These are some of the remaining problem areas to be considered, before the level of forest fire monitoring can be improved. In comparison to our work, the focus is rather on networking aspects of the deployed system than on sensor integration. Also, smoke, as a type of dust, is inferred from temperature measurement and not from actual particulate matter detectors.

In [6], a comprehensive overview of smart home instrumentation systems (Figure 2) along with suitable hardware developments are presented. The authors state that air quality assessment and the thermal comfort sensation depend on numerous variables which are difficult to measure precisely at low cost. They present the core of the sensor system as a comprehensive monitoring system which continuously measures the indoor and outdoor air quality and through gas leakage detection and early fire warnings also address essential security matters. The conditions for indoor air quality control and the assessment of subjective thermal comfort are sometimes contradicting. For example, if the CO_2 concentration exceeds a certain level, commonly 1000 ppm, then the air quality control must have priority over the adjustment of thermal comfort. In our view, the wireless measurement node that we developed is suitable for hardware and software integration into such an environment. The advantages of low power wireless communication, small device size can be used to achieve significant savings in retrofitting existing homes and prepare them for smart house technology.

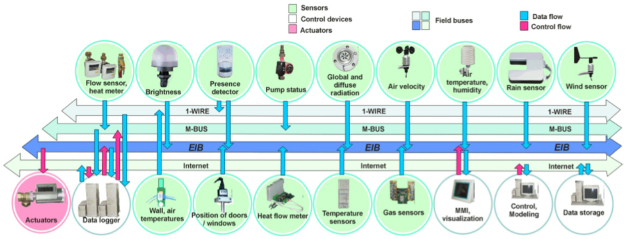


Figure 2. SmartHome Instrumentation Overview [6]

III. System Design

The high level architecture of our proposed system is shown in Figure 3. It consists of a mesh network of wireless measurement nodes relaying data towards the sink. At the gateway level, the data is collected, stored and presented for interpretation or further processing.

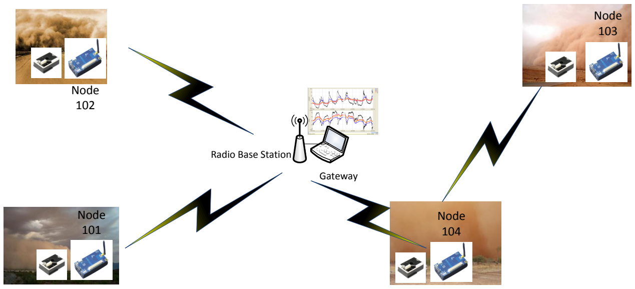


Figure 3. Wireless Dust Measurement System Architecture

A wireless measurement node consists of four parts: a processing/radio board, a mote data acquisition board, a optical dust sensor interfaced with a microcontroller development board. The most relevant characteristics of the hardware that we have chosen are described next.

The IRIS XM2110 is the main processing/radio board, which hosts an ATMega 1281 8 bit MCU and a IEEE 802.15.4 compliant RF230 radio transceiver operating in the 2.4GHz Industrial Scientific and Medical (ISM) band. This is the newest module in the line of the original Berkeley motes and is supported by the open- source community under TinyOS 1.x and 2.1 an event-based, low footprint operating system for resource constrained devices. Compared to the previous iteration MicaZ, the producer mentions better performance in terms of radio coverage and improved energy efficiency. The 51-pin connector provides stackable expansion possibilities to connect to the MCU peripherals.

The MDA300 [7] is a generic data acquisition expansion board for the IRIS platform. It offers analog input channels, digital input and output channels, relays and external sensor excitation. This opens up a whole range of new applications such as remote process control. The complete feature list is the following:

- 7 single-ended or 3 differential ADC channels;
- 4 precise differential ADC channels;
- 6 digital I/O channels with event detection interrupt;
- 2.5, 3.3, 5V sensor excitation and low-power mode;
- 64K EEPROM for onboard sensor calibration data;
- 2 relay channels, one normally open and one normally closed;
- 200 Hz counter channel for wind speed, pulse frequencies;
- external I2C interface.

The Sharp GP2Y1010AU0F [8] is a dust sensor with optical sensing system. An infrared emitting diode (IRED) and a phototransistor are diagonally arranged into this device. It detects the reflected light of dust in air. Especially, it is effective to detect very fine particle like cigarette smoke. In addition it can distinguish smoke from house dust by pulse pattern of output voltage. The features that have recomended it are the compact size envisioned for integration in air purifiers or air conditioning units which is also suitable for our design. Very important for wireless embedded application is the low current draw of 20mA. We employ a popular Arduino board to handle the interfacing of the sensor in this initial iteration. The microcontroller board is based on the ATmega328. It has 14 digital input/output pins, of which 6 can be used as PWM outputs, 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.The board has the task of enabling the duty cycling of the command for the optical sensor LED according to the pattern in Figure 4. Basically, a digital output controlling the led of the detector has to be set to low for 0.32ms in a 10ms time period, afterwards a analog voltage output can be read.

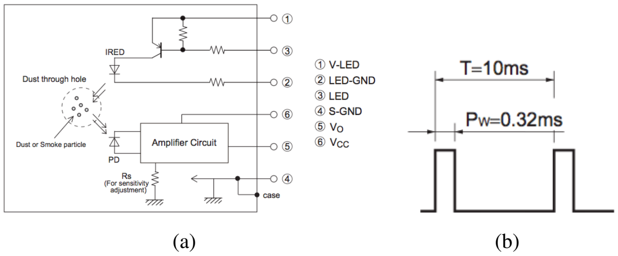


Figure 4. Optical Dust Sensor (a) Internal schematic (b) PWM Excitation

We place the dust sensor in our region of interest and air starts flowing naturally through the measurement space. The microcontroller board runs the embedded software loop to activate the light source and read the sensitive element processed output. Once a value has been read, it is output in the form of an analog signal captured by the ADC0 channel of the MDA300 board and interfaced to the appropiate input channel of the mote. The mote is tasked with broadcasting a radio message containing this voltage value. This happens either directly to the base station or, if the networking protocol decides that there is no direct or poor connectivity to the base station, via multi hop communication with neghboring nodes acting as relays for the source of the data. Our system communicates directly with the radio base station but multi hop is supported without any additional configuration. The overview of the components is shown in Figure 5. We have also employed auxiliary elements such as resistors and capacitors to build the sensor power supply circuit and the pulse width modulation to analog output for the microcontroller board.

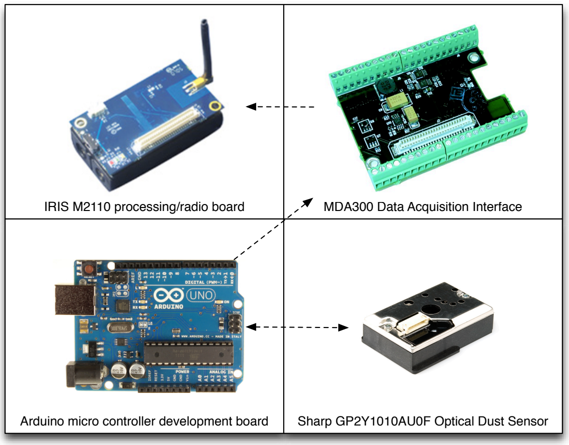

IRIS M2110 processing/radio board
MDA300 Data Acquisition Interface
Arduino micro controller development board
Sharp GP2Y1010AU0F Optical Dust Sensor

Figure 5. Components Overview

IV. RESULTS: VISUALIZATION AND EVALUATION

Figure 6 illustrates the experimental laboratory deployment. As a reference system, we use the Dylos DC1100 Air Quality Monitor which is a professional grade laser particle counter. It offers two measurement channels, one for small particles in the range $0.5\mu m - 2.5\mu m$ such as bacteria and mold and one for large particles, $2.5\mu m - 10\mu m$ like pollen or thick smoke. In comparison, the datasheet of the low cost optical dust sensor states more vaguely that it detects particles over $1\mu m$. Our version includes an RS-232 connector to communicate with a dedicated logging software on the host PC which records particle concentration every minute for both channels. The device is factory calibrated and we rely on this calibration to perform comparison to our measurement node. The default measurement unit is a count thousands of particles per cubic foot which has to be converted in milligrams per cubic meter. For the wireless node, the application handling the data is called MoteView [9]. Data logging and display is supported via MoteView user interface. The software is designed to be the primary interface between a user and a deployed network of wireless sensors. It provides an intuitive user interface to database management along with sensor data visualization and analysis tools. Sensor data can be logged to a database residing on a host PC, or to a database running autonomously. One important factor to consider is that the voltage values are scaled to adapt to the MDA300 input range of 0 to 2V as compared to the maximum saturated output of the dust sensor of 3.7V.

We follow the conversion procedure [10]. The assumption it follows are:

- particles are spherical, with a density of $1.65E12\mu g/m^3$;
- the radius of a small particle is $.44\mu m$;
- the radius of a large particle is $2.6\mu m$.

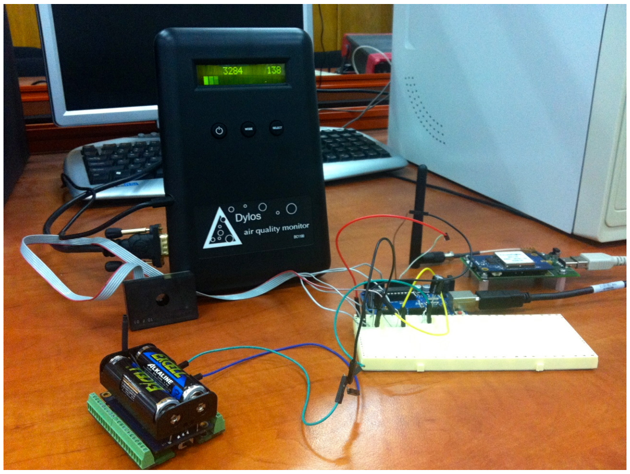


Figure 6. Laboratory Implementation of the Proposed System

Table I
DYLOS DC1100 PARTICLE COUNT VALUES

Channel	Small	Large
Average	666329	95512
Low	68300	8000
High	5157300	850900
Average mg/m^3	0.0139	0.4097
Low mg/m^3	0.0014	0.0343
High mg/m^3	0.107	3.64

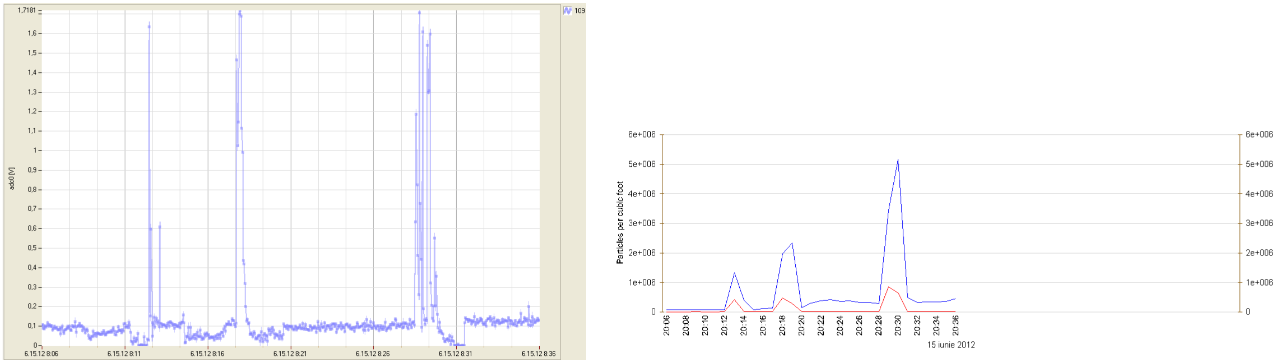
so that the formula for deriving mass density from particle count is:

$$c[mg/m^3] = \frac{n}{0.0283} \cdot \frac{4 \cdot \pi}{3} \cdot r^3 \cdot \rho \qquad (2)$$

where n is the particle count per cubic foot, r the radius of a particle and ρ the specific density of the particles.

Therefore, Table 1 shows particle counts per cubic foot for the small and large channels of the air quality monitor over the experiment period. The average, low and high values for both channels are also converted to mg/m^3 to enable comparison with the observed peaks of the Sharp dust sensor response.

The main experiment that we have performed consisted of a continous 30 minute monitoring period with the results illustrated in Figure 7. In this time span we have acquired 31 samples from the professional device and 632 wireless measurements from our node, with a 3 second sampling interval. We can see that the peaks are correctly identified by the measurement node which outputs a saturated value of around 1.72V. Comparing this to the characteristic output curve of the sensor in Figure 8, we assimilate the voltage output of our system for a maximum concentration of particles of $0.55mg/m^3$. An interesting fact is that, over the saturation limit we can still evaluate higher concentrations of dust due to the prolonged width of the high output signal.



(a)

(b)

Figure 7. 30 Minute Particle Monitoring Results (a) Wireless Measurement Node (b) Dylos DC1100 Air Quality Monitor

In practice, after the first smoke event the spike in the sensor answer seems somewhat isolated but for the next two experiments there are more similar high value readings in the corresponding time frame.

In order to establish a dust concetration baseline, we alternate steady periods with high particulate matter events in the form of smoke resulting from paper incineration. The spikes in the data analysis show the response of both systems to such events. The increasing baseline certifies that the dust sensor we use is sensitive to the steady increase in ambiental particulate concentrations resulting from paper burning events. The baseline voltage ranged from 0.05V to 0.15V in the later stages of the deployment.

Fine grained analyisis can be performed only in a given volume enclosure with a pre-measured quantity of particles. In our case we relate to the room volume and place the two measurement systems close together. It it to be noted that the Dylos device uses forced supply by means of a small fan while the measurement node relies on natural air flow.

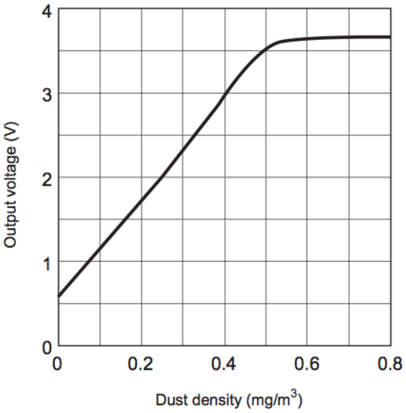


Figure 8. Dust Sensor Output Characteristic

After this experimental evaluation, we can conclude that initial testing has showed promising results regarding the response and sensitivity of the Sharp optical dust sensor and, implicitly, of our wireless measurement node. The professional grade air quality monitor provided a trustworthy calibration reference which has enable us to establish a baseline for dust concentration and effectively measure particle counts and equivalent mass densities.

V. Conclusion and Future Work

In this paper, we have reported how to create a wireless node to measure dust concentration at any industrial site or other areas, by combining smart data acquisition technology and wireless sensor networks. All the components in the laboratory have been connected in one system to create a smart node, and were adjusted according to their specific purpose. By using specialized software we collected the results from the nodes and we created the wireless link with the available network by self configuration with the network. In addition to the low cost, a smart node has many advantages: it runs on batteries which can operate for long periods of time, it is scalable, allowing the addition of nodes to additional features and to expand the network up to tens or even hundreds of nodes. We have used a professional grade air quality monitoring device in order to validate the experimental data coming from our system in low and high particle concentration environments.

Future work direction is two fold. First, we want to make the microcontroller development board redundant by implementing TinyOS components and modules directly on the mote processing board to handle sensor reading tasks. Its sole role as an intermediary in the current iteration of the system can be removed and thus enable the node to operate independently. Second, we propose to evaluate an alternative to the Sharp sensor in the form of the Shinyei PPD4NS particle sensor. Literature review has pointed out that in some situations, it is better suited to certain application due to enhanced sensitivity. Also, the pulse width modulation of this sensor works in a more reliable manner than the analog output voltage of the Sharp version.

ACKNOWLEDGMENT

The work has been partially funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/88/1.5/S/60203.

REFERENCES

[1] Y. Li, My T. Thai and W. Wu (Eds.), *Wireless Sensor Networks and Applications*, Springer, 2008.

[2] V. Sgârciu and G. Stamatescu, *Distance Process Monitoring using LabVIEW Environment*, LabVIEW - Modeling, Programming and Simulations, Ed. Riccardo de Asmundis, pp. 67-89, InTech, January 2011.

[3] F. Hauert and A. Vogl, *Measurement of Dust Cloud Characteristics in Industrial Plants*, Technical Report PL910695, Research Center for Applied System Safety and Industrial Medicine, January 1995.

[4] J. Zhang, W. Li, N. Han and J. Kan, *Forest Fire Detection System based on a ZigBee Wireless Sensor Network*, Front. For. China, vol.3(3), pp. 369-374, Higher Education Press and Springer Verlag, 2008.

[5] ZigBee Alliance, *ZigBee Specification*, ZigBee Document 053474R17, 2008.

[6] B. Ivanov, O. Zhelondz, L. Borodulkin and H. Ruser, *Distributed Smart Sensor System for Indoor Climate Monitoring*, Proc. of KONNEX Scientific Conference, Munich, 2002.

[7] Crossbow Inc., *MTS/MDA Sensor Board Users Manual*, Revision A, June 2007.

[8] Sharp Corp., *GP2Y1010AU0F Compact Optical Dust Sensor Datasheet*, E4-A01501EN, 2007.

[9] M. Turon, *MOTE-VIEW: A Sensor Network Monitoring and Management Tool*, 2nd IEEE workshop on Embedded Networked Sensors, EmNets '05, pp. 11-17, 2005.

[10] J. Lee et al., *Seasonal variations of particle size distributions of PAHs at Seoul, South Korea*, Air Quality Atmospheric Health, vol. 1, pp. 57-68, 2008.

Monitoring and Management of Power Consumption in Apartment using ZigBee

Kwang-Soo Kim, Hyunhak Kim, Tae-Wook Heo, Jong-Arm Jun
USN/IoT Research Department
Electronics & Telecommunications Research Institute
Daejeon, Republic of Korea
e-mail: {enoch, hh.kim, htw398, jajun}@etri.re.kr

***Abstract*—This paper describes a field demonstration system that monitors the power consumption in an apartment. To evaluate the effectiveness of the system, 20 houses selected from Mudeung Park Apartment have participated in this project. The system includes 20 smart meters, 20 in-home displays, 2 data collection units, 20 energy service interface servers, and one meter data management server. A wireless sensor network based on ZigBee is used to exchange metering data between smart meters and data collection units as well as between an in-home display device and an energy service interface server in a home; the meter data management server exchanges data with data collection units and energy service interface servers through an IP network. The meter data management server has operated to collect periodically the power consumption from all meters and to send the amount of power usage and charge as well as additional data such as weather and price policy to each in-home display device. This testbed provides valuable insights about the operation of the system in an apartment that has a lot of obstacles which interfere in the wireless communication.**

Keywords- smart grid; WSNs; apartment; power monitoring

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]-[3].

Korean Government has constructed a smart grid testbed on Jeju Island [4][5]. About 10 consortiums in five areas have participated in the testbed project to test technologies and develop business models. The areas are smart place, smart transportation, smart renewable, smart power grid, and smart electricity service. The smart place is related to customers directly and includes customers as a part of a smart grid to increase energy efficiency. As a utility provides power usages and charges that customers have consumed for the customers, the utility encourages consumers to save energy. The smart transportation constructs a charging infrastructure for electric vehicles. There are two types of electric vehicle chargers: high speed charger and low speed charger. The former can be used at electric vehicle charging stations and the latter can be used at a house. Therefore, a utility should consider the number of electric vehicles, the amount of battery installed in each electric vehicle, and the time when an electric vehicle starts the charging in order to balance the power generation and the power consumption. The smart renewable develops renewable power generation technologies and builds power generation complexes to reduce CO_2 emissions. Also, the smart renewable can construct a micro-grid that generates and consumes the power by itself. The smart power grid constructs open power grids that can allow various interconnections between power suppliers and consumers. By using the open power grids, many new business models will be developed. The smart electricity service is to improve consumer’s right-to-choose by introducing various price policies and demand response programs.

There are several competing technologies for capturing and transmitting the electricity usages of consumers in the smart grid, such as wired technology, PLC (power line communication) technology, and WSN (wireless sensor network) 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]-[14]. 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.

In Korea, the ZigBee and the PLC are the popular communication technologies to implement the monitoring and controlling of the energy consumption for the smart grid. We selected the Zigbee technology because it provides the high interoperability as well as is the defacto standard in the smart grid. Also, a ZigBee Device can be moved to another place easily. The ZigBee Alliance published the smart energy profile for interoperable products that monitor,

ISBN: 978-1-61208-207-3

control and automate the delivery and use of energy [15]. The profile includes several specifications related to the advanced metering, the demand response and load control, pricing, and text message.

We installed several devices at 20 houses whose residents expressed the participation in this project. They were selected from Mudeung Park Apartment to evaluate our system in a city environment with a lot of obstacles that interfere in the wireless communications. The system is shown in Figure 1. Every house has one smart meter and one IHD (in-home display), and one ESI (energy service interface) server. Two houses from those have additional devices: PCT (programmable communicating thermostat) and LCD (load control device). MDMS (meter data management server) installed in a utility collects the power usage of each customer through DCUs (data collection units), calculates the charge on the usage, and sends them to IHD periodically. A customer can check them on the screen of IHD.

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 2010, we implemented several devices based on the ZigBee standards [16]. They were installed in a two-story building in Jeju Island, South Korea. Our system focused on measuring and monitoring the amount of power consumption and generation, and controlling the energy consumption on the customer side. The system consists of 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. A central 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 charges.

However, the system test in Jeju Island has a shortcoming because the system was installed on a typical rural village. It is difficult to prove the efficiency of the system and the reliability of the wireless communication. As the people in the village go to work early in the morning, return to their home at sunset, and go to bed early in the evening, they almost do not use the power during the day and use a little at night. Therefore, it is very difficult to prove that the system saves the power consumption or increases the power efficiency. Furthermore, there are few structures that interfere with the wireless communication because most houses built in a single layer and are relatively far apart. Therefore, it also is hard to prove that the wireless communication used to exchange metering data among devices is stable. To evaluate the efficiency and the reliability of the system, we selected another testbed where a lot of barriers exist and the power usage is relatively high. We have worked with LS Industrial System, KEPCO (Korea Power Corporation), KEPCO Data and Network, ADTechnology, Wooam Corporation, KERI (Korea Electro Technology Research Institute), KETI (Korea Electronics Technology Institute).

III. Testbed Design

The system consists of six main components: (i) MDMS, (ii) DCU, (iii) ESI, (iv) IHD, (v) load control devices (LCD and PCT), (vi) smart meter, and (vii) an information network connecting all devices.

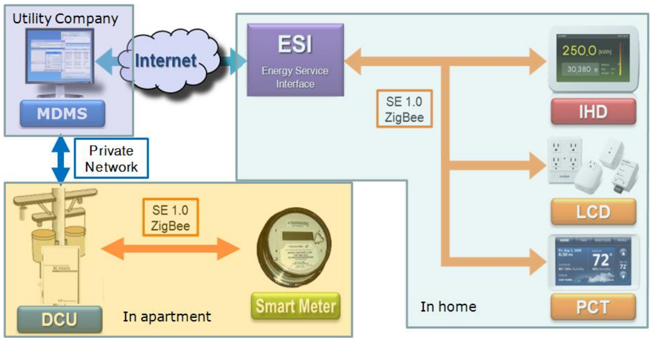


Figure 1. System configuration

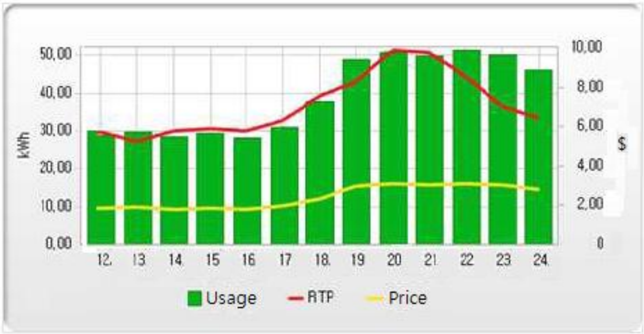


Figure 2. Power usage and charge

A. Meter Data Management Server

MDMS is the key component of our system. MDMS collects metering values from all smart meters through DCUs, and sends power usage and charge of each customer to IHD installed in the house of the customer. MDMS manages not only metering values and charges but also customer information, weather, CBL (Customer Baseline) for billing, power factor, price, etc. MDMS gathers the weather from Korea Meteorogical Administration and sends it to customers according to their residential area one a day, about 5 AM. MDMS calculates CBL that indicates a baseline load shape of the power usage of each customer. It is used to calculate the billing for each customer. Figure 2 shows the power usages and the charges from the noon to the midnight on 25 October. The bar chart indicates the usages, and the red and the yellow line curves indicate RTP (Real Time Price) and general price, respectively. Also, MDMS can send a load control message to a customer while the amount of power consumption is increasing dramatically. A load control device which receives the load control message reduces the power usage of an appliance by adjusting an operating condition of it or turns off an appliance, for example, increasing the set temperature of an air-conditioner.

MDMS uses CIM (common information model)-based XML messages to collect metering data from DCUs and to send billing information to each IHD. CIM is a standard defined as IEC (International Electrotechnical Commission) 61968 for exchanging information among electrical systems and devices [17]. Figure 3 is an example of customer billing information. The customer billing information is consisted of daily or monthly charge of a customer, billing date, billing cycle, meter identifier, type of billing, billing period, etc.

Customers can check their power usage and charge on a portal that MDMS provides. The type of users of the portal is divided into two groups roughly. One is customers and the other is administrators. For customers, the portal provides customer information including name, address, energy usage, charge, etc. For administrators, it provides the device information, network state information, statistics on the energy usage and charge. Authorized customers on the system can check their own information through the Internet. The Portal assigns different access rights to each user in order to limit the available information according to the user authority. For example, a customer cannot access the network state information. On the other hand, an administrator can access all information related to both customers and administrators. All information the Portal uses is stored and managed by MDMS.

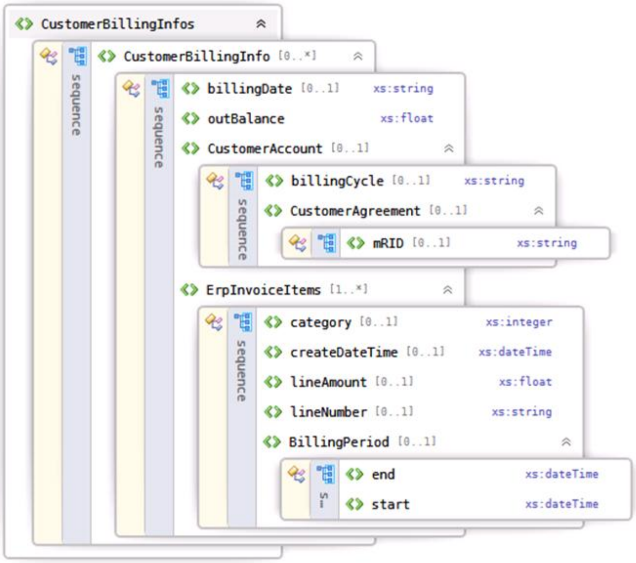


Figure 3. Customer billing information

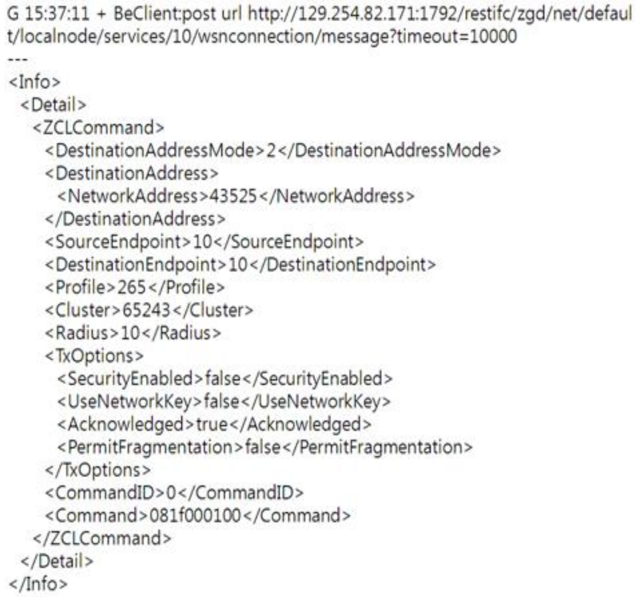
B. Data Collection Unit

DCU locates between a smart meter and MDMS and distributes the load of data collection of MDMS. If MDMS collects metering values from a thousand of meters directly, it gives much of a burden on MDMS. To collect metering values, a variety of communication methods such as ZigBee[18], PLC[19], and WiFi[20] are implemented in DCU.

In this project, DCU is installed on a utility pole outside the apartment and acts as a gateway in a wireless sensor network. The goal of DCU design is to collect metering values from 200 meters within 15 minutes. In this testbed, one DCU connects to ten smart meters and collects a metering value from each smart meter periodically. The data received from smart meters are transmitted to MDMS according to a transmission schedule. The size of data which DCU can sent at a time is designed less than 680 Bytes. During the operation of the system, there are many missing data due to collisions among data transmissions sent by smart meters. Therefore, the success rate of metering data collections is about 96 percent due to the collisions. To overcome these limitations, we adapt a query scheduling method which collects simultaneously two metering values. A query scheduling assigns a report time to each smart meter to avoid the collisions. Each smart meter sends its two values, the current metering value and the value which was measured 15 minutes ago, at the time assigned by the scheduler. However, this method increases the success rate to only 99 percent. The missing values are collected by a batch process in which MDMS executes one per day.

C. Energy Service Interface

ESI becomes a bridge between the wireless sensor network and MDMS through an IP-based network. ESI provides the gateway device interface and the smart energy profile 1.0 announced by ZigBee Alliance as well as provides the functions that connect and manage all devices within a home area network [15][18]. ESI includes PANC (personal area network coordinator) starting the network formation and routing messages between the gateway and each electric device. ESI is implemented on Linux 2.6 in the C programming language, and uses PHP (personal hypertext preprocessor) for the web application to exchange data formatted in XML with MDMS.



```
G 15:37:11 + BeClient:post url http://129.254.82.171:1792/restifc/zgd/net/defaul
t/localnode/services/10/wsnconnection/message?timeout=10000
---
<Info>
  <Detail>
    <ZCLCommand>
      <DestinationAddressMode>2</DestinationAddressMode>
      <DestinationAddress>
        <NetworkAddress>43525</NetworkAddress>
      </DestinationAddress>
      <SourceEndpoint>10</SourceEndpoint>
      <DestinationEndpoint>10</DestinationEndpoint>
      <Profile>265</Profile>
      <Cluster>65243</Cluster>
      <Radius>10</Radius>
      <TxOptions>
        <SecurityEnabled>false</SecurityEnabled>
        <UseNetworkKey>false</UseNetworkKey>
        <Acknowledged>true</Acknowledged>
        <PermitFragmentation>false</PermitFragmentation>
      </TxOptions>
      <CommandID>0</CommandID>
      <Command>081f000100</Command>
    </ZCLCommand>
  </Detail>
</Info>
```

Figure 4. Example of ZCL message

The data transmitted through the home area network is formatted in ZCL (ZigBee cluster library). We use different clusters according to the type of transmitted data. When we receive the metering data from a smart meter, we use the simple metering cluster. If we want to control LCD, we use the demand response and load control cluster. They are included in SEP (smart energy profile). ESI coverts a CIM-based XML message into a ZCL message and vice versa. An example of ZCL message is shown in Figure 4. The message is sent to LCD to collect the energy usage that an appliance connected to the LCD has consumed.

D. In-Home Display

IHD is a portable device installed in a home. Therefore, a customer can move it to another home. In this situation, we have to prevent this IHD from joining to another ESI installed in the home where the customer visits. If this IHD connects to ESI of another home, it might receive the power usage and charge of the home that the customer visits. To prevent the situation, we make ESI, IHD, PCT, and LCD within the same home share the same unique key.

IHD receives several data such as power usage and charge, weather, CBL, price, etc., from MDMS and displays them on the screen. Table 1 shows the attribute of CBL curve. IHD draws a CBL curve when it receives the data. The user interface of IHD is shown in Figure 5. A customer can see the amount of power consumption, hourly power price, the trend of them, and weather on the same window.

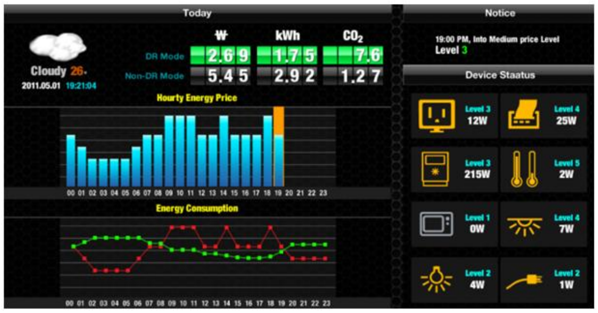


Figure 5. User interface of IHD

E. Load Control Device

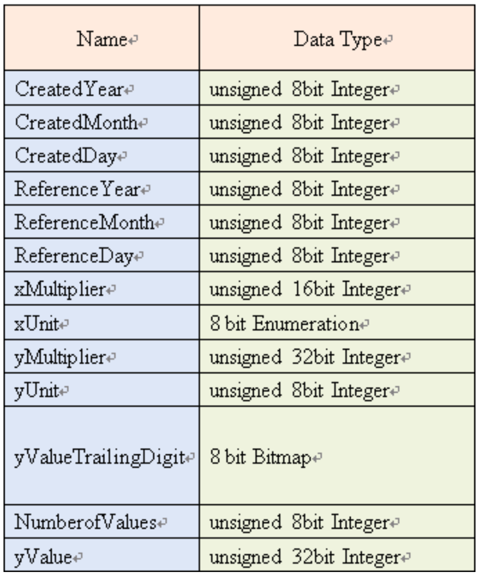
A load control device is divided into two groups: PCT and LCD. They receive a load control message and return the result to MDMS. LCD has been implemented in the form of a switch and an outlet. Their roles are a little different. PCT can control an air conditioner with ZigBee functions; however, PCT does not have a metering capability. On the other hand, LCD controls an appliance such as an electric pan; and it can measure the power usage of an appliance that connects to itself because it has a metering capability. Furthermore, LCD can send its metering value to MDMS through ESI. Therefore, our system can measure a fine-grained power usage that each appliance has consumed.

F. Smart Meter

A smart meter measures the electricity usage of a home in which it is installed. It has recorded the electricity usage and sent the data formatted in the smart energy profile [15] to DCU on every 15 minutes. The meter also sends LP_Data (load profile data) including forward active power, backward active power, reactive power, etc as well as power quality. 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 power quality consists of the set of electrical properties such as power quality, frequency quality, and voltage quality.

The meter provides several different tariffs including time-of-use, real-time-pricing, critical-peak- pricing, and progressive-pricing. In Korea, the progressive pricing policy is applied to most of electric users. However, the Korean government seeks to change the tariff from the progressive pricing to the real-time-pricing to reduce the electricity usage.

TABLE 1. ATTRIBUTE OF TODAY CBL



Name	Data Type
CreatedYear	unsigned 8bit Integer
CreatedMonth	unsigned 8bit Integer
CreatedDay	unsigned 8bit Integer
ReferenceYear	unsigned 8bit Integer
ReferenceMonth	unsigned 8bit Integer
ReferenceDay	unsigned 8bit Integer
xMultiplier	unsigned 16bit Integer
xUnit	8 bit Enumeration
yMultiplier	unsigned 32bit Integer
yUnit	unsigned 8bit Integer
yValueTrailingDigit	8 bit Bitmap
NumberofValues	unsigned 8bit Integer
yValue	unsigned 32bit Integer

G. Information Network

An information network called SUN (smart utility network) can be divided into three categories. One is HAN (home area network) that is a residential local network for communicating among ESI, LCD, and IHD deployed in a home; another is NAN (neighborhood area network) that communicates between DCU and either ESI or smart meter; the third is WAN (wide area network) that communicates between DCU and MDMS operated by a utility company. In this project, ZigBee is used in both the home area network and the neighborhood area network, and the Internet is used in the wide area network. HAN devices such as IHD and LCD cannot communicate with NAN devices such as DCU and a smart meter due to the wall of the house.

Our wireless sensor network is developed based on ZigBee specification and IEEE 802.15.4. The sensor network installed in each house consists of one ESI, one LCD, one IHD, and one PCT. In this project, the devices can exchange messages via one-hop communication with star topology because every device is within the radio range of ESI. Also, DCU exchanges messages via one-hop communication with a smart meter.

After PANC included in ESI starts the network formation, a device willing to associate with the network starts the association procedure by requesting for a beacon with channel scanning. A joined device permits the association by beaconing with setting permit-joining flag on. Once a device 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 at 20 houses in an apartment to evaluate the efficiency and the reliability of the system. To monitor and control the usage and charge of the electricity in each home, we install MDMS, DCUs, smart meters, and several devices including ESI, IHD, LCD and PCT in which locate a house. MDMS, DCU and ESI exchange their data and control messages through the Internet, 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, a customer could save the electric bill by checking the charge in real time. One of the most difficulties is to collect the metering values from smart meters through the wireless sensor network. When we have collected the values on every 15 minutes, the success rate is about 96 percent because the data collision in the air. Therefore, we collect simultaneously two metering values on every 15 minutes, the last two metering values including the current value, to increase the success rate of the data collection. MDMS also collects a missing value one per day if necessary.

In the future, we will operate the system more than one year, accumulate the metering data to analyze the efficiency of the system, and perform the economic analysis on the smart grid testbed to apply the devices to other households and buildings. In addition, we have to solve several security issues, because the power consumption measured in a sensor network is directly related to the amount due on the electricity bill of the user.

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 LS Industrial System, KEPCO, KEPCO-KDN, ADTechnology, Wooam, KERI, KETI members 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] http://smartgrid.jeju.go.kr/eng

[5] http://en.wikipedia.org/wiki/Jeju_Smart_Grid_Demonstration_Project_in_Korea

[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] 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.

[10] 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.

[11] 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.

[12] 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 2006), vol. 1, Jun. 2006, pp. 99–103.

[13] 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, Aug. 2008, pp. 531-536.

[14] J. A. Stankovic, “When Sensor and Actuator Networks Cover the World,” ETRI Journal, Oct. 2008, pp. 627-633.

[15] ZigBee Alliance, “Smart Energy Profile Specification,”(075356r15ZB), Dec. 2008.

[16] K.S. Kim, H. Kim, T.W. Heo, Y. Doh, and J.A. Jun, “A Smart Grid Testbed using Wireless Sensor Networks in a Building,” Proc. International Conference on Sensor Technologies and Applications, Aug. 2011, pp. 371-374

[17] IEC, “Application integration at electric utilities – system interfaces for distirbution management”, IEC 61968, 2003

[18] ZigBee Alliance, “ZigBee Gateway Device Specification,” (075468r30ZB), Jul. 2010.

[19] H. Dai and H. V. Poor, “Advanced Signal Processing for Power-Line Communications,” IEEE Comm. Mag., May, 2003, pp. 100-107.

[20] IEEE , “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (IEEE 802.11)”, 2010

U-Park : Parking Management System Based on Wireless Sensor Network Technology

Nikos Larisis[1], Leonidas Perlepes[1], Panayiotis Kikiras[2], George Stamoulis[1]
[1]Department of Computer and Communications
University of Thessaly, Volos, Greece
[2]AGT Germany, Darmstadt, Germany
nilarisi, leperlep, georges@inf.uth.gr, pkikiras@agtgermany.com

***Abstract*—Wireless Sensor Networks can be entitled as one of the most challenging and emerging technologies empowering the provision of enhanced services to miscellaneous application domains. The objective of this paper is to examine and present an implementation of a real-time parking management system. This system adapts efficiently into a contemporary urban environment, and eventually provides to users the ability to find and navigate easily to a free curb space. The system is comprised of a deployed sensor network based on two collaborative vehicle detection schemes supported by an event-driven processing algorithm and a web based application. The evaluation of the system is performed by conducting a number of experiments in order to select the optimal sensing modality and confirm the feasibility of the proposed application system in actual running conditions. Results demonstrate that the proposed system is capable of effectively detecting overlying automobiles and robustly distinguishing false positive indications.**

Keywords-wireless sensor network; vehicle detection; magnetic sensor; passive infrared sensor; web based interface

I. INTRODUCTION

The very viable and sustainable development of urban environment is inextricably connected with contemporary approaches concerning traffic congestion issues. A basic aspect of this problem is undeniably the parking procedure, a costly and time-consuming stage within urban transports and an everyday headache for millions of drivers and urban planning departments. Modeling this procedure is not an easy task due to problems in measuring the overall time since it being dependent upon inherently random factors. Looking for parking duration it has been proven that is not negligible and indeed according to [1], “cruising in congested downtowns takes between 3.5 and 14 minutes to find a curb space, and between 8% and 74% of the traffic can be appointed to cruising for parking”. In addition to that, a variety of studies, as [2] states, infer that automobiles spend approximately over 95% of their time parked and for trucks this percentage is over 85%.

In this paper, an innovative approach in solving the aforementioned problem is proposed utilizing WSNs (Wireless Sensor Networks). WSN technology has a wide spectrum of applications which are deeply embedded into the everyday world ranging from health monitoring and disaster early-warning systems up to vehicle detection and military purpose application scenarios. Such a sensor network is comprised of a large number of microelectronic devices capable of monitoring and collecting raw data in different environments and conditions. Typically, each device –often characterized as a node– is consisted of a microcontroller, a power source, a radio frequency transceiver, an external memory, and various sensors. These resource constrained sensor devices communicate wirelessly and untethered, forming complex mesh networks through which data is being disseminated.

Our objective is to investigate, simulate and experimentally verify the prospect of implementing an intelligent parking management system that provides to drivers the possibility to conveniently locate and be guided towards a free parking lot near their destination. From a system design perspective, WSN solution offers a number of unrivaled advantages. Firstly, we can hugely benefit from its low-cost deployment. Secondly, there is flexibility with regards to vehicle detection approaches since a wide spectrum of applicable sensing techniques is being offered. Finally, autonomous and long-term operation of the system compounds is another important feature. The latter can be achieved thanks to the enhanced administrative capability since extra information can be retrieved in real-time concerning features like illegal parking activity, time duration of each occupied lot, and other metrics that are indicative of a robust system operability.

Although there is an abundance of existing solutions focusing on parking assistance applications that are employing WSN techniques, our approach differentiates in many aspects. We propose a hybrid detection scheme utilizing magnetic and passive infrared sensors. A pair of sensors, located at a distinct parking spot within the urban fabric, collects the raw data. Then this information is received and processed at the sinks, and ultimately evaluated whether if the correspondent spot is occupied or not. The result is appropriately stored in databases in the base station and then it is uploaded to a web interface and finally, depicted onto a user friendly map.

The ensuing content of this paper is organized as follows. Section 2 briefly describes the related work and how it is correlated with our approach. Section 3 overviews the design stages of our system, the operating principle of the utilized devices and the system architecture. Section 4 describes system's implementation. Section 5 reports the system's testing results with respect to the various simulation scenarios and requirements. Finally, Section 6 concludes with general remarks and discussion about the potential for future expansion.

II. RELATED WORK

Confronting the problem of park cruising within the context of urban transportation planning is not something new to this research field. Besides theoretical approaches in analyzing the problem [1][2][3], there are numerous research projects [4] utilizing intelligent transportation systems (ITS) techniques outside the WSN research area. Indicatively, we can mention mobiPARK [5], a free parking lot indicating system based on a text messaging service platform for a driver that departs from a previously occupied lot. NYU-Poly and Rutgers University [6] collaborating research teams attempt to provide a multimedia stream application that could display free parking spots within the congested Brooklyn. As drivers cruise streets, sensors feed real-time visual information regarding parking availability uploaded via a 4G network and depicted onto special graphic user interface (GUI) platforms.

From another perspective that focuses mainly on WSN technologies there can be found a number of similar approaches. Parking Finder [7] is a Mica2Dot/MIB510-based WSN platform implemented for detecting the presence of a car over a designated parking lot within a metropolitan city infrastructure. It employs magnetic sensors (HMC1001) for vehicle detection with each mote running on a TinyOS [8] environment. After data aggregation and storage in the basestation the processed data can be accessed via a Java-based GUI. Although there are many commonalities with our presented approach, it lacks specific characteristics that pose serious threats to its overall credibility. The absence of experimental verification is more than evident. The GUI does not fully utilize the advantages of a web-based implementation that our version benefits from and does not offer enough interoperability. Above all, the unilateral usage of magnetic sensors does not take into consideration the inherent abnormalities of the application environment. Thorough experimentation within our research proved as necessary the complementary usage of infrared sensors which Parking Finder excluded as inappropriate at the initial stages of its implementation.

In Intelligent parking lot application using WSN [9], it was proposed and proven as optimal the combinatory usage of magnetic and ultrasonic sensors for an accurate and reliable detection of vehicles within a multi-storied parking space. Although there are similarities in the deployment scenery since Tmote Sky sensors are also utilized, their proposal undeniably cannot withstand a boisterous and hectic city center environment. Moreover, their data processing and detection algorithm followed a different approach employing a modified version of the min-max algorithm suitable for analyzing the waveformed raw magnetic and acoustic data. Their extensive experimental results imply and strengthen the necessity for assiduous system verification, an important attribute for every kind of such an application area systems.

Design considerations for a WSN for locating parking spaces [10] presents a scheme for reliable detection of automobiles employing the same type of sensor nodes, Mica2/MTS310/MIB510 equipped with the same 2-axis magnetometer HMC1002. Their proposed platform refers to public parking garages and is consisted of a network of nodes that are distributed in strategic locations throughout the targeted public parking area instead of at every parking spot. Although the prominent differences in the objective, the deployment environment and the algorithmic notion of the detection process, the important commonality lies in the crucial analysis of the peculiarity in the operating nature of the magnetic sensor. The obtained through trial and error accurate positioning and orientation of the node was also observed in this paper. Despite that, their proposed algorithmic approach in order to surpass this obstacle differentiates from our more practically orientated perspective.

III. DESIGN CONSIDERATIONS

In this section, the design process is presented detailing the procedure under which the specific combination of magnetic and passive infrared sensors was promoted as the optimal choice. Moreover, the operating principle of these sensing devices is briefly discussed and finally, the system architecture is unfolded.

A. Design Process

Solving a complicated issue such as cruising for curb-space parking, it is relevant upon multilateral research derived from distinct launching perspectives. Figure 1 indicates the initial attempt to address a solution via the usage of a mechanical arm with a sensor node attached at its base, placed under the road surface or enclosed in a protective sealing. A vehicle while in parking process would drag down the joint and an actuator would recognize the corresponding slope if a predefined critical angle ($\Phi critical$) was breached. This in turn would enable the record of a vehicle's presence for the designated spot and this information would be propagated throughout the network for further processing. After consideration, this scheme was aborted from being a plausible implementation due to the inappropriateness of joint's material while colliding with vehicle parts and the inherent difficulties in implementing such a mechanical arm based on the laboratory's available apparatus.

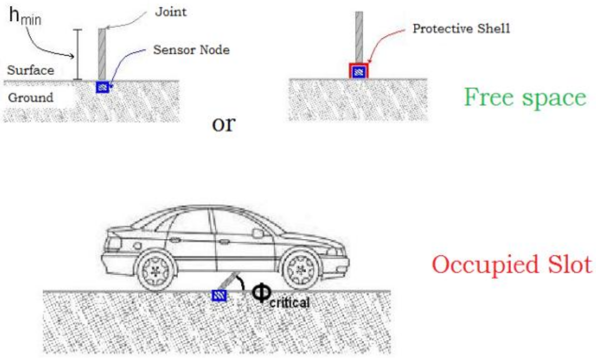


Figure 1. Initial system implementation.

As a next step passive infrared sensors (PIR) were selected as a plausible detection scheme within a deployed sensor network. Their advantage lies within their passive nature, that is, independence from external conditions since receiving and not emitting signal. Experimental verification proved that the PIR sensors were extremely sensitive in response to environment stimuli and their detection range was exceedingly wide. This results to the sensor being constantly triggered after the initial successful detection of the vehicle, in case later on another car or pedestrian was mobile within the detection zone. The unilateral usage of PIR was thus rejected since neither the detection zone nor the detection sensitivity can be recalibrated to custom needs according to its manual [11].

As a next designing stage, the viability of acoustic sensors' usage was investigated. Various experiments were conducted but without deducing sufficient and credible conclusions regarding neither a unilateral nor a collaborative usage of acoustic sensors. A detection pattern for overlying and passing vehicles could not be concluded and thus uniformly applied since noise in an urban fabric, and especially near road surface, is an unpredicted factor.

In order to select the optimal sensing device, and consequently scheme, we ought to consider the very nature of the object to be detected. Since a vehicle is in fact a sizable metal mass, the subsequent distortion it enforces in the earth's magnetic field is easily tractable by a magnetic sensor, thus resulting to its imperative usage. This distortion is unique for every ferromagnetic object and under ideal conditions can be perceived as its magnetic footprint, a useful characteristic for a wide spectrum of ITS applications relevant to tracking and detecting automobiles [12].

In addition to that, an auxiliary operation of passive infrared sensors would benefit the most to the system. In order to narrow their detection area it is obligatory to place them under road surface and inside a protective enclosure. The road penetrating costs can be compensated if considering the advantages that can be obtained through this enclosure approach such as, extra protection from theft and harsh weather conditions [13].

At the initial experiments, there were observed minor discrepancies in the magnetic sensor's output waveform. Strangely enough, it remained idle at a maximum value of approximately 780 (based on the range the utilized Oscilloscope/TinyOS application produces as an output, see Figure 2). Moreover, it did not respond to the presence of any ferrous objects unless a material with high magnetic permeability was presented in a distance less than 2cm. According to [14], "the NiFe core of the magnetic sensor is extremely sensitive. However, it is also subject to saturation. Saturation occurs when the sensor is exposed to a large magnetic field. Unfortunately the MTS310 circuit does not have an automatic saturation recovery circuit". After experimentation, and having excluded the prospect of implementing such a recovery circuit, it was elicited that the sensor can be calibrated via appropriate nesC [15] coding of the relevant interfaces and components. Its sensitivity level can be set manually within a range between 0 and 255, and accordingly to the ambient earth's magnetic field. Thus, as the sensor is moved around in the field its output waveform will ultimately exhibit transitions. It was experimentally proven that the exact sensitivity level for each deployment site can be calculated and eventually a waveform in between the desired range can be received. The sensor is thus ready for use.

B. Device Specification

The magnetic sensor module is based on Crossbow's Mica2 [16] motes equipped with the Atmega128L MCU (128KB flash, 521KB storage, 4KB RAM) that are attended with the MIB510CA programming board [17] both depicted in Figure 3. Sensing operation is performed via the MTS310 sensor board [18] equipped with the HMC1002 2-axis anisotropic magneto-resistive (AMR) sensor that functions under the following principle. Since the earth's magnetic field is uniformly distributed along an area of several kilometers a ferrous object (e.g., automobile) causes local disturbances in this field while idle and/or moving [19][20]. The AMR sensor, implemented as Wheatstone bridge, detects such disturbances in the X or Y axis and interprets them as a differential input voltage that can be transformed into exploitable information regarding the presence or not of an overlying ferrous object.

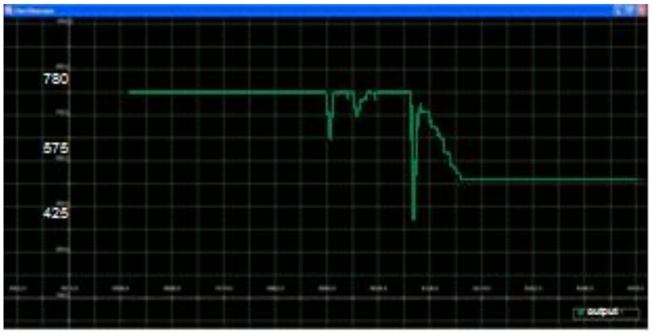


Figure 2. Transitions of the magnetic sensor's output while being moved.

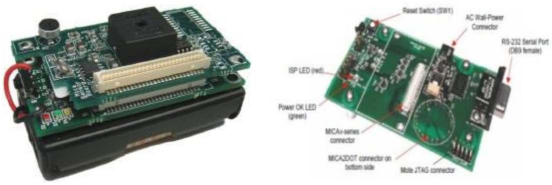


Figure 3. Mica2 mote equipped with the MTS310 sensing board and the MIB510 programming board, [16][17].

Figure 4. Tmote Sky equipped with WiEye sensor board, [22].

The other subsystem is constituted of a number of deployed Sentilla's Tmote Sky [21] nodes based on TI-MSP430F1611 MCU (48KB flash, 1MB storage, 10KBRAM). Sensing operation is performed via the WiEye sensor board [22] equipped with the AMN44121 PIR motion sensor, NaPiOn [23], all shown in Figure 4. According to the pyroelecrtic phenomenon [24], crystalline compounds can produce electric charge while being exposed to thermal energy. A variable thermal flux within the infrared spectrum will deduct a variable load onto the quad-type pyroelectric element of the Napion which's output amplitude into the measuring circuitry will be analogous to the amplitude of the changing flux. Thus, when an object enters the detection zone, NaPiOn identifies the temperature difference between the target and its surroundings and transforms this difference into a processable value-signal.

C. System Architecture

The proposed real-time parking management system is a hierarchical module comprised of collaborating subsystems and based on a three tier architectural design (see Figure 5).

At the lowest level the data collection subsystem developed in TinyOS-2.x samples the environmental stimuli which are then aggregated to the appropriate sinks. Current system implementation directs a one hop routing scheme containing four motes, two sensor motes (one magnetic and one PIR) that comprise a deployed system node, and the two correspondent base stations. Scalability though can be easily achieved via the usage of proper disseminating nesC interfaces, since the other hierarchical levels are already set for a larger number of deployed nodes.

The second level is comprised of the data processing applications residing in the two base stations. Specially implemented detection algorithms indicate the occurrence of a magnetic or a PIR triggering event. This information is then stored with a timestamp at a database named sensor_state. Each record refers to the status of the pair of magnetic and PIR sensors that are located at each parking lot. A second database named nodes is used for correlating these data with the third level. Analogously, each record corresponds to a deployed node and contains information regarding the status of the node and thus the availability of the parking lot, as well as timestamps and various spatial information. Moreover, there are Java-based algorithms responsible for the initialization of the system, the execution of the surveying algorithm and the update of all the databases' records correspondent to each node deployed.

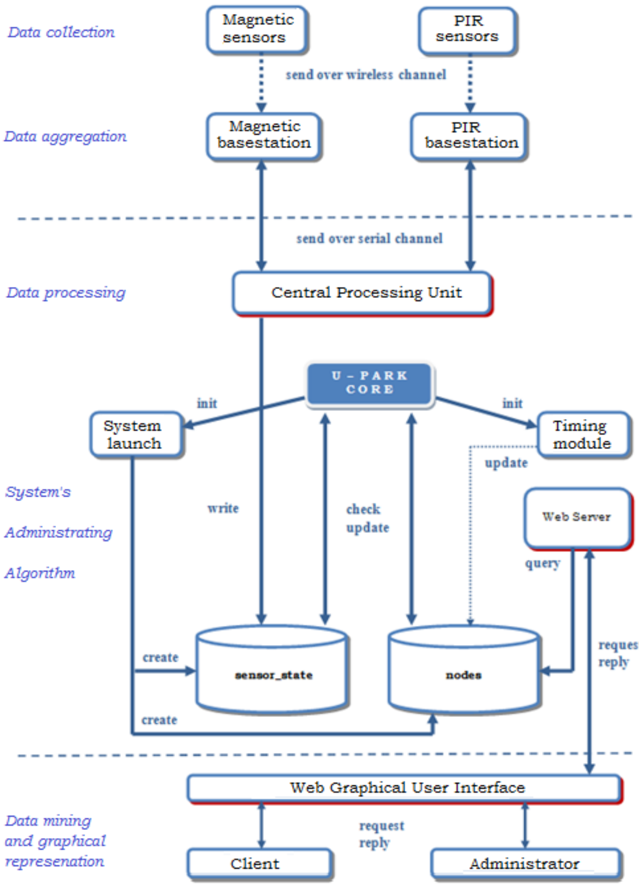


Figure 5. System's information flow chart.

Finally, a web server program is implemented with the task of uploading the information of the nodes database to a web GUI. There are two interfaces implemented with distinct layout, one for the administrator (providing extra functionality) and one for the potential clients that both offer a customized query processing operability. The end user can select an area of interest within the urban fabric and he is given a real-time graphical representation of the available parking lots with green dots appearing on a google map. The user is also provided if requested with routing directions. The administrator on the contrary has access to extra information such as lot's duration of occupation, overall system view, or error node's indication. An overall illustration of the system's information flow is presented in Figure 5.

IV. System Implementation

In the current section, the algorithmic perspectives of raw data processing as well as the administrative principles of our system are explained.

A. Raw Data Analysis

The designing process for an algorithm appointed with the magnetic raw data analysis is crucial for the overall credibility of the system. The initial experiments regarding HMC1002's response to ferromagnetic stimuli proved that a rather simple detection algorithm can be utilized based on two fundamental notions deriving from the Statistics field, the arithmetic mean and the population variance. It was observed and verified that in idling state the collected data is characterized by a mean value within a predefined equilibrium level and by a variance tending to zero. Upon the approximation of a ferrous object the output waveform would effect a drastic transition also indicative via an increase in the variance of the samples. Moreover, a constant presence of the object would be accompanied with a progressive stabilization of the samples' mean and with a variance again tending to zero. To this extend, the subsystem's states can be concluded (Table I), provided that appropriate operating levels are predefined with regards to the output waveform (see Figure 6).

As mentioned before, the processing algorithm of the infrared data has an auxiliary role due to the fact that after experimentation it was proven that PIR sensor was constantly triggered and thus, implementing an accurate pattern detection scheme was unfeasible. Its final contribution was confined in confirming an overlying ferrous object after consulting for a potential magnetic triggering recorded event. Nevertheless, its usage cannot be excluded otherwise the overall system's credibility can be dramatically hindered. Through its utilization all false positive magnetic triggering events can be neglected. On a similar basis we can define the correspondent operating states, as well as a state diagram (see Figure 7).

B. Core Algorithm

The core of the administrative algorithm consists of a infinite loop in which the node statuses are retrieved from the databases and according to various combinations we can define updating functions calculating the next system state.

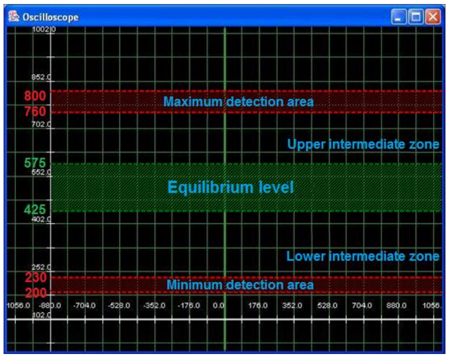


Figure 6. Magnetic sensing subsystem's operational levels.

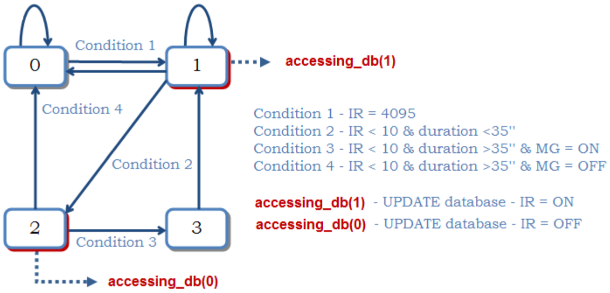

0
1
2
3
Condition 1
Condition 4
Condition 2
Condition 3
accessing_db(1)
accessing_db(0)
Condition 1 - IR = 4095
Condition 2 - IR < 10 & duration <35''
Condition 3 - IR < 10 & duration >35'' & MG = ON
Condition 4 - IR < 10 & duration >35'' & MG = OFF
accessing_db(1) - UPDATE database - IR = ON
accessing_db(0) - UPDATE database - IR = OFF

Figure 7. Infrared sensing subsystem's finite state machine.

These combinations are explained in Table II (OFF stands for a sensor not detecting any stimuli, and ON otherwise). In case A, both sensors are activated and thus we deduce that the corresponding parking lot is occupied. Cases B and C are referring to undesirable circumstances whenever either an object with low magnetic permeability (animal, pedestrian, other) is revolving near and/or above the protective enclosure or an automobile has not parked within the designated lines of the parking slot. In both cases the specific parking slot is consider to be in error state**.** After experimentation it was derived that the PIR sensor detection range (placed within the enclosure) is under 20cm, and the magnetic sensor less than 2m. The last combination (case D,E) corresponds to both sensors being deactivated. That is either when the system has recuperated from an error state (due to hardware failure, false positive magnetic triggering or cease of constant PIR triggering) or the parking lot is free after a vehicle completed its withdrawal. In both cases, the space is free and the nodes are reinstantiated.

TABLE I. Magnetic Sensing Subsystem States

STATE	System State	Interpretation
0	System Idle	Absence of ambient magnetic stimuli
1	Detecting	Variation detected
2	Parking slot - Busy	Overlying ferromagnetic object
3	Car unparking	Withdrawal initialization
4	Car withdrawing	Disengage. Final stage of withdrawal
5	Parking slot - Clear	Initial idling stage
6	B U S Y@MID - checking	Check for potential malicious parking behavior
7	B U S Y@MID - PICKED	Potential malicious parking behavior confirmed

TABLE II. System Status

Magnetic (MG)	Passive Infrared (PIR)	Interpretation
ON	ON	Case A - Car detected
OFF	ON	Case B - Error : MG = OFF ^PIR = ON
ON	OFF	Case C - Error : MG = ON ^PIR = Off
OFF	OFF	Case D - Node reinstated
		Case E - Car withdrawing

V. System Evaluation

In this section, the experimental setup is presented including a detailed description of the experiments that were conducted and the evaluation results that reinforce the system's credibility.

A. *Simulation Scenarios*

System verification and evaluation process included different simulation scenarios that reassured the correctness and the preciseness of the detection procedure. These scenarios involved among others,

- detecting an overlying and/or passing bicycle, motorbike, and distinct manufacturer car models (and thus distinct ferromagnetic footprints),
- conducting measurements for sensor's response while a vehicle could be approaching from different directions and approximating distances,
- simulating a parking procedure with different kinds of maneuvers and total moves,
- executing an experiment emulating actual parking conditions in a free spot between two parked cars,
- verifying system's response while using two magnetic sensors in order to simulate a malicious driver behavior,
- evaluating system's response in full-scale and real-time operating conditions.

Especially for the latter two scenarios, they were crucial in verifying the system's credibility. As previously stated, NaPiOn can benefit the most for the system. Its usage is essential in detecting a malicious driver parking between two nodes. As depicted in Figure 8, although magnetic sensors are initially triggered (t1 and t2 timestamps) the car stops in between them and outside their detection area. If NaPiOn sensor is used then the magnetic initial triggering event could be verified and act accordingly, resulting in successful vehicle error detection (Case B in Table II). The final scenarios included experiments being conducted in actual urban conditions. A picture of the developed WSN within a real parking area is given in Figure 9 with green arrow indicating the deployed pair of sensor nodes.

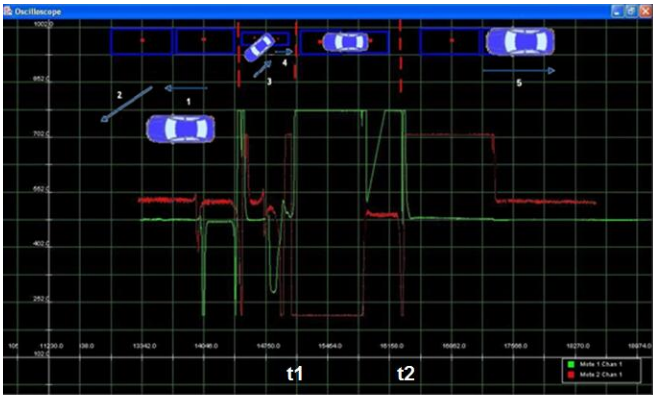


Figure 8. Setup justifing the necessity of NaPiOn's usage.

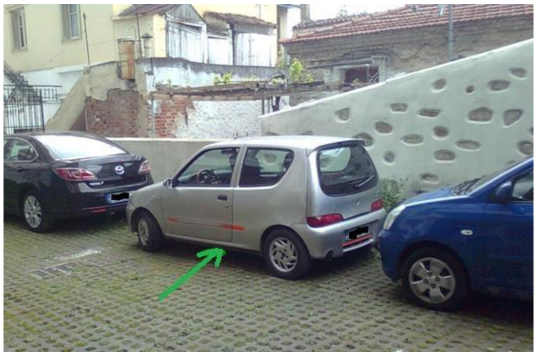


Figure 9. Deployed WSN in actual parking conditions.

B. *Experimental Results*

The experimental procedure evolved according to the following stages. Initially, the response of each sensor was measured individually and in laboratory environment. This was a compulsory step in order to become familiarized with the sensors' operating principle and in order to assure that the design of the detection algorithm would follow the correct path. As a next stage, experiments involving sensors' response within outdoor conditions were conducted.

The NaPiOn's outdoor experiments verified the initial studies and assumptions regarding its response to ambient stimuli. NaPiOn reacts independently from the object's nature to be detected and the weather conditions. On the contrary, magnetic sensor's experiments demanded a much more strenuous treatment since an assiduous comprehension of HMC1002's sensing philosophy was imperative before embarking on fully designing its detection algorithm.

In order to conclude to as safe experimental results as possible, it was necessary that the experiments were to be repeated under different environmental conditions and simulating scenarios. In Table III the characteristic environmental conditions of such distinct experiment are presented. As implied from the data, experiments on different dates and times of day were conducted in order to benefit from variable weather conditions. Regarding the magnetic sensor, its sensitivity function was invoked with different parameters for the scenarios were one and two Mica2 motes (n1, n2 in table) were deployed respectively.

TABLE III. Experiments Environmental Conditions

Date	09/04	13/04	16/04	17/04	18/04
Sensitivity	50,100 for n1	50 for n1	50 for n1 - 45 for n2	50 for n1 - 45 for n2	50 for n1
Time	17:00 – 19:00	18:23 – 20:10	17:30 – 21:00	20:30 – 21:30	17:30 – 19:30
Temperature	17ºC	23 ºC	16 ºC - 13 ºC	9 ºC - 8 ºC	10 ºC
Humidity	~60%	~50%	~55%	~92%	~77%

An important characteristic is the fact that night and rain conditions were present on 17.04.2011 day adding therefore more credibility to our experimental set. Apart from the above, more experiments were conducted on laboratory conditions regarding the evaluation and the correctness of the detection algorithm while it was being implemented throughout its various design stages.

Moreover, real-time system verification was performed including a final full-scale emulation of an actual parking procedure. The conducted experiments included active participation of all the operational components of the system that ranged from the initialization of the sensor nodes, verifying the response of the administrative algorithm and until projecting the information of the system onto the map of our web GUI. These experiments were performed both in safe and hostile environments. The former was performed in the scenery depicted in Figure 9 and the latter in front of the University building in one of the most traffic congested avenues and during evening rush hour. Truly, in both cases the response of the system was exceptional after carefully selecting such a sensitivity level that would avoid passing vehicles (even buses) hinders its functionality. Performing experiments with such a wide range of conditions verified the anticipated behavior of both the utilized sensors and verified the correctness of the utilized combinatory sensing modality. In Figure 10 the hand-crafted protective enclosure containing the pair of sensor nodes used during experimentation process is presented. The enclosure is made of a material with zero magnetic permeability (wood) and on the top it is protected from overlying leakage by a PVC surface enabling radiation penetration and thus not hindering NaPiOn's operability. More information on the verification procedure including photo shoots and video raw footage can be found on the application's support website [25].

VI. Conclusion

In this section, the overall knowledge obtained throughout the research procedure is presented and future work concludes the implementation presented.

A. *Lessons Learnt*

This research idea was developed in an attempt to seek an applicable solution for a major problem of contemporary urban environments. Cruising for parking contributes significantly in exaggerating air pollution due to traffic congestion issues, phenomena that ought to be reduced or even evicted from our future envisioned megacities. Our objective was to assiduously examine the viability of such a proposed application system by evaluating it on a small-scale deployed WSN in real-time and actual conditions. After examination the optimal WSN sensing technology was utilized, comprised of a collaborative usage of magnetic and passive infrared sensors. The data sampling procedure was implemented in Java via customizing the default TinyOS-1.x and 2.x functionalities, and the administrative algorithm included various other features responsible for processing, storing, uploading and retrieving the collected information. Ultimately, a designated parking curb space can be monitored and accordingly stamped onto a web GUI map if it is occupied or not. The experimental and simulation procedure involved testing a great range of detection scenarios and under variable ambient conditions. Concrete evidence was derived in terms of the specific utilized detection scheme and in comparison with the related work proposals. The magnetic sensor can be manually calibrated in order to adjust to custom needs and thus can efficiently detect any overlying vehicle with merely zero error rate. The auxiliary operation of the passive infrared sensor was proven obligatory since it enables the system to be aware of any kind of malicious parking behavior or false positive magnetic indications. Thus, the overall system error rate is substantially low, a fact proven through the strenuous full-scale experimental results.

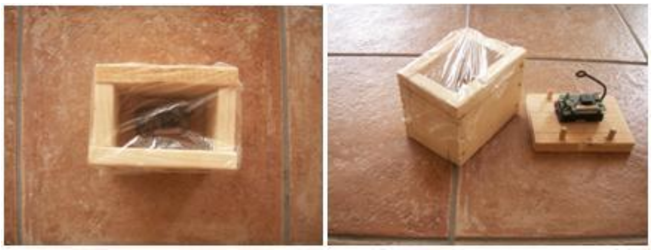


Figure 10. Nodes' hand-crafted protective enclosure.

B. *Prespectives for Future work*

The current implementation scale can be considered as a rather limited one if we contemplate upon the actual capabilities this system can offer at a full-scale deployment. To that extend, its further expansion could enable further optimization of the administrative algorithm in terms of reducing its response time from the moment that ambient stimuli is recorded until when the information is depicted onto the google map. Another important feature that should be ensured is expanding into a multihop networking architecture. On the same notion, when a multihop scheme will be adopted charging machines (present in every urban fabric) can be equipped with a pair of motes so as to operate with the form of relay nodes with extra powering capabilities that would reduce in this sense the overall networking strain. Some optimistic scenarios would even promote this parking assistance system as a platform responsible for traffic management system based on the knowledge that each driver declares his destination and thus he could receive distinct routing directions towards each free parking space. To that extend, routing algorithms can be adopted from relevant research areas in order to facilitate such kind of expansion.

Acknowledgment

This research was conducted at the Wireless Sensor Systems Laboratory of Computer and Communications Engineering Department at University of Thessaly during a period from December 2010 until July 2011.

REFERENCES

[1] D. C. Shoup, “Cruising for parking,” Transport Policy, Elsevier, vol.13, no. 6, pp. 479–486, November 2006.

[2] K. Button, “The political economy of parking charges in ‘‘first’’ and ‘‘second-best’’ worlds,” Transport Policy, Elsevier, vol.13, no. 6, pp.470–478, November 2006.

[3] P. G. Höglund, “Parking, energy consumption and air pollution,” Science of Total Environment, Elsevier, vol.(334-335), pp. 39–45. December 2004.

[4] L. E. Y. Mimbela and L. A. Klein, “A summary of vehicle detection and surveillance technologies used in intelligent transportation systems,”New Mexico University, The Vehicle Detector Clearinghouse, November 2000, unpublished.

[5] http://www.mobipark.gr 18.04.2012

[6] http://www.poly.edu/press-release/2010/11/05/free-4g-and-freed-parking-spot-nyc 18.04.2012

[7] M.Eissa and M.Elhag, “Park finder,” University of Illinois, December 2004, unpublished.

[8] P. Levis et al., “TinyOS: An operating system for sensor networks,”Ambient Intelligence, Springer, pp. 115–148, 2005.

[9] S. Lee, D. Yoon, and A. Ghosh, “Intelligent parking lot application using wireless sensor networks,” IEEE, International Symposium on Collaborative Technologies and Systems, pp.48–57, May 2008.

[10] V. K. Boda, A. Nasipuri, and I. Howitt, “Design considerations for a WSN for locating parking spaces,” IEEE, Proceedings of SoutheastCon, March 2007.

[11] Panasonic Electric Works Co., “NaPiOn (AMN) catalog.”

[12] L. E. Y. Mimbela and L. A. Klein, “Surveillance technologies used in intelligent transportation systems,” New Mexico University, The Vehicle Detector Clearinghouse, August 2007, unpublished.

[13] B. Song, H. Choi, and H. S. Lee, “Surveillance tracking system using passive infrared motion sensors in WSN,” International Conference on Information Networking, pp. 1–8, January 2008.

[14] Honeywell International Inc., “HMC1001/1002/1021/1022 1- and 2-axis magnetic sensors datasheet.”

[15] D. Gay, P. Levis, D. Culler, and E. Brewer. 2005. “nesC 1.2 Language Reference Manual,” ACM SIGPLAN Notices, vol. 38, no. 5, pp. 1–11, 2003.

[16] Crossbow Technology Inc., “MPR-MIB users manual,” Revision B, June 2006.

[17] Memsic Inc., “MIB510CA datasheet.”

[18] Crossbow Technology Inc., “MTS/MDA sensor board users manual,”Revision A, June 2007.

[19] http://www.metal.ntua.gr/uploads/3190/225/5th_ex.pdf 18.04.2012

[20] M. J. Caruso and L. S. Withanawasam, “Vehicle detection and compass applications using AMR magnetic sensors,” Honeywell International Inc., 1999, unpublished.

[21] Moteiv Corporation, “Tmote Sky datasheet,” 2006.

[22] EasySen LLC, “WiEye datasheet,” 2008.

[23] Panasonic Electric Works Co., “AMN motion sensors design manual datasheet.”

[24] http://users.auth.gr/~vanidhis/pdf/paragogi/7.pdf 18.04.2012

[25] http://inf-server.inf.uth.gr/~nilarisi/en/main.html 18.04.2012

A Study on the Applicability of Energy Harvesting Technology for the Sensor Network of Railroad System by Thermal Deviation

Jaehoon Kim
High-Speed Railroad Research Center
Korea Railroad Research Institute
Uiwang-City, Republic of Korea
lapin95@krri.re.kr

Jae-yun Lee
Graduate School of Mechanical Engineering
Sogang University
Seoul, Republic of Korea
jaeyun.76@sogang.ac.kr

***Abstract*— In this study, we verified the applicability of the energy harvesting technology to the railroad system for real-time wireless sensor networks. The origin of the power for this technology is found in the concept of energy harvesting, which resources thermal deviation occurring when a train runs. For this, we measured the temperature generated on the axle box bearing from the train's running environment by using a high-speed train, analyzed the thermal deviation and calculated the estimated energy harvesting power for the sensor network of the railroad system. The thermal deviation occurred in the axle box bearing when a train runs; the deviation was approximately 12.97 ~ 14.68 ℃. We calculated in the assumption of the basic thermo element of the axle box bearing area, which is the area of the temperature sensor attached on the axle box bearing. Theoretically, the energy harvesting power of 63.8mW can be obtained and thus by using the energy harvesting power it will be possible to operate the wireless sensor network for railroad monitoring.**

Keywords-Railroad; Sensor; Energy harvesting; Thermal deviation.

I. Introduction

With regards to railroads, accelerating the speed of the system always demands absolute development in reliability and safety of passengers and trains, and the increase in system maintenance cost has become another problem. Thus, it is necessary to develop a new technology of maintenance for fulfilling both reliability and safety, and maintenance cost in system speed acceleration. In order to accomplish this, a real-time sensor network system must be established. Usually, a sensor network system is needed for prevention and response before a dangerous situation occurs, by detecting abnormal conditions in the corresponding system during the operation. It is also used for constant monitoring to analyze the information for possible understanding of the situation.

However, present monitoring systems in railroad systems almost exclusively use wired sensor networks, and recently, the demand for monitoring systems with wireless sensor networks has expanded in accordance with their easy installation in places that had previously been considered difficult to access and install in. Specifically, when a wireless sensor network is applied to the railroad monitoring system, it is possible for the real time condition based-maintenance, differentiated from the existing method used for scheduled maintenance, and by this, the reliability and safety of maintenance is increased [1]. Though, in the case of a wireless sensor network with easy installation, it is necessary to solve the problem of power supply for using it in a monitoring system. This is because the regular changing of batteries is necessary for additional maintenance. Therefore, for the wireless sensor network in a railroad system, it is necessary to develop an 'Energy Harvesting' technology in order to be maintenance-free from the changing of batteries. It is also environment-friendly and semi-permanent [2]-[4].

Therefore, in this study, we verified the applicability of the energy harvesting technology to the railroad system for real-time wireless sensor networks. The origin of the power for this technology is found in the concept of energy harvesting, which resources thermal deviation occurring when a train runs. For this, we measured the temperature generated on the axle box bearing from the train's running environment by using a high-speed train, analyzed the thermal deviation and calculated the estimated energy harvesting power for the sensor network of the railroad system.

II. The temperature change of axle box bearing during the operation

A. *The expriments*

In this study, surrounding energy generated at the normal train operation environment was measured using high speed rail train in operation and to identify the applicability of the energy harvesting, the energy source was measured with the train running 586.2km (return way from Seoul to East-Daegu) at maximum operating speed 300km/h. And to measure the temperature variation of the heat sources generated during train operation at 300km/h, temperature sensor (Model: TC1047AVNB, 2.64mm x 3.05mm x 1.02mm, temperature-ranges of -40℃ ~125℃) was installed on axle bearing which is one of the part on which the temperature varies significantly with Bluetooth module as shown in Fig 1 and the temperature variation of axle bearing was measured on a real-time basis using wireless sensor.

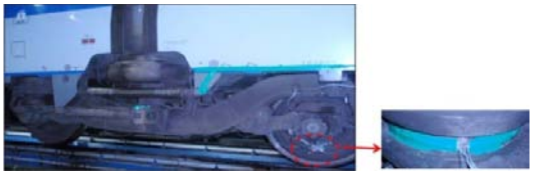


Figure 1. The temperature measuring on the axle box bearing

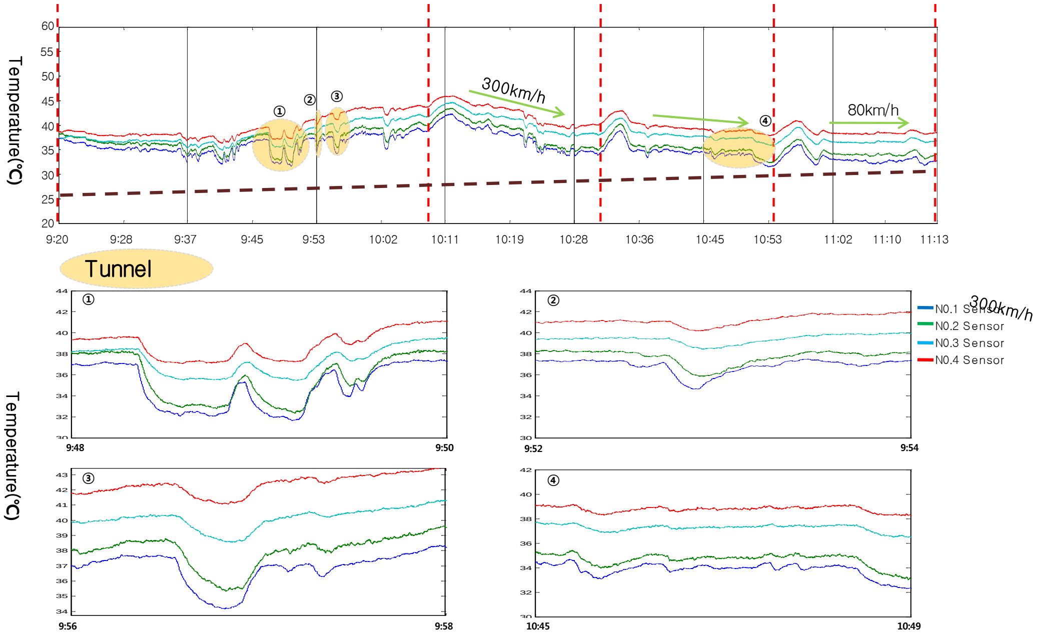


Figure 2. The temperature change of axle box bearing during the operation and zoom-in of tunnel section

B. *The result of temperature change of axle box bearing*

We analyzed the temperature of the axle box bearing in a 300km/h maximum speed operation train. As a result, as shown in Fig. 3, the temperature of the axle box bearing during the operation repeats up and down in accordance with the surroundings; tunnel, bridge, speed and stations, etc. If we look closely at Fig. 2, the temperature of the axle box bearing starts with a temperature similar with the outside air temperature, but increases as the train runs. The deviation between outer air and axle box bearing is 15～20℃ in the summer. Especially, the temperature tends to decrease when the train passes tunnels and each sensor's temperature change occurs by the location of the sensor on the same axle box bearing, as in Fig. 3, which depends on the direction of operation. The reason is the air-cooling effect taking place during the train operation, so the temperature of No. 1 and No. 2 sensors is lower than the temperature of No. 3 and No. 4. Also, regarding the temperature between the axle box bearing and outside air, we can use this thermal deviation to make energy harvesting possible if a thermo element is applied as the power.

In addition, the tunnel section of Fig. 2, the temperature changed very rapidly; the train enters the tunnel, the temperature of the axle box bearing descends, and after the train passed the tunnel, it ascends again. This is because of the air-cooling effect from the outside air temperature of going in and out of the tunnel. The outside air temperature in the tunnel is lower than the outside air temperature outside of the tunnel in the summer. Also, the average temperature data from the axle box bearing of high-speed trains is stated in Table 1. The outside air temperature was measured approximately 28℃～30℃ and the average thermal deviation (ΔT) was measured approximately 12.97℃～14.68℃.

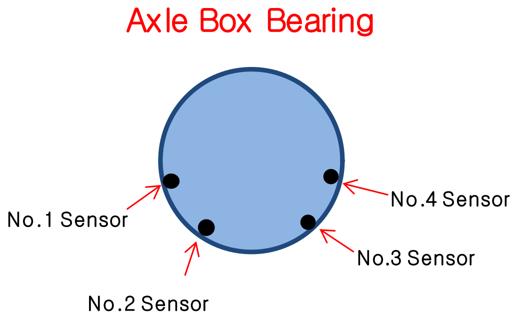


Figure 3. The temperature sensors location on the axle box bearing

TABLE I. THE AVERAGE TEMPERATURE OF AXLE BOX BEARING

Sensor No.	**1**	**2**	**3**	**4**
Temperature(℃)	43.39	42.97	44.68	44.31

III. APPLICABILITY OF ENERGY HARVESTING TECHNOLOGY FOR THE SENSOR NETWORK OF RAILROAD SYSTEM BY THERMAL DEVIATION

When looking at the results of the average temperature on the axle box bearing, we notice that the highest temperature was 44.68℃. Thus, if we suppose the outside air temperature is fixed at 30℃, the biggest thermal deviation obtainable from the axle box bearing is 14.68℃; generally, the thermo element is known to properly gain efficiency when the temperature deviation between high and low temperature parts is over 300℃ [5]. However, if the thermo element form is made as thin as a form of Thin-Film,

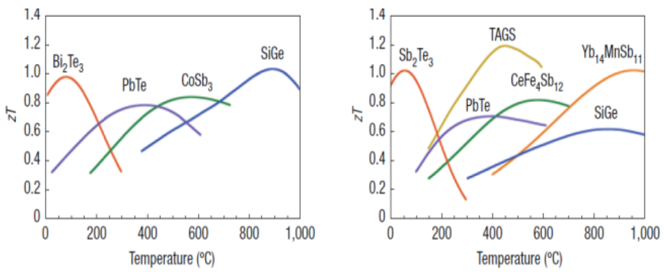


Figure 4. Figure-of-Merit vs. Temperature [6]

TABLE II. THE ESTIMATED ENERGY HARVESTING POWER BY THERMAL DEVIATION (SINGLE SIZE THERMO ELEMENT : A=4 , D= 1 ㎝, LARGE SIZE THERMO ELEMENT : Ø= 25.4 ㎝, W= 2 ㎝, D=1 ㎝)

sensor	Δ T	Power (mW) : Single size element	Power (mW) : large size element
1	12,97 K	1.2	47.9
2	13.93 K	1.4	55.8
3	14.31 K	1.5	59.8
4	14.68 K	1.6	63.8

it is not easy to obtain electricity and efficiency in small temperature deviation [6]-[8]. Of course, in this case, the Thin-Film thermo element has the most optimum quality index in corresponding temperature and should be chosen. As you can see in Fig. 4, when the thermal deviation is 14.68 ℃, n-Type should be selected as Bi_2Te_3 (Bismuth Telluride) and p-Type as Sb_2Te_3 (Antimony Telluride) for Figure-of-merit theoretically [6]. For the normalization of obtainable power by thermal deviation occurring in the axle bearing, we calculated in the assumption of the most basic thermo element, a single rectangular Thin-Film thermo element of 4 ㎠ area, 1 ㎝ thickness. Also, the axle box bearing symmetry and the thermal deviation in the axle box bearing is relatively even in all locations. Therefore, the maximum energy harvesting power obtainable by using average temperature measured in the 4th sensor was 1.6mW as shown in Fig. 5 and the power estimated through ΔT is shown in below Table 2.

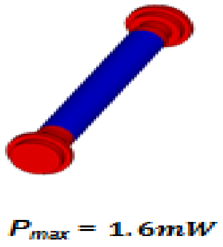
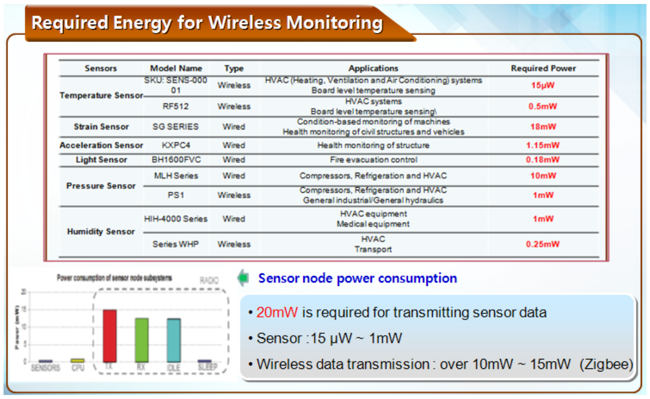


Figure 5. Estimated energy harvesting energy on the axle box bearing

We estimate the energy harvester need to generate more than 20mW for self powered (= energy harvesting) wireless on-board condition monitoring system. As shown in Fig. 6, sensor consumes very small energy, 15uW ~1mW, but wireless data transmit requires 10mW~15mW energy, even Zigbee protocol [9].



Required Energy for Wireless Monitoring

Sensors	Model Name	Type	Applications	Required Power
Temperature Sensor	SKU: SENS-000 01	Wireless	HVAC (Heating, Ventilation and Air Conditioning) systems Board level temperature sensing	15μW
	RF512	Wireless	HVAC systems Board level temperature sensing\	0.5mW
Strain Sensor	SG SERIES	Wired	Condition-based monitoring of machines Health monitoring of civil structures and vehicles	18mW
Acceleration Sensor	KXPC4	Wired	Health monitoring of structure	1.15mW
Light Sensor	BH1600FVC	Wired	Fire evacuation control	0.18mW
Pressure Sensor	MLH Series	Wired	Compressors, Refrigeration and HVAC	10mW
	PS1	Wireless	Compressors, Refrigeration and HVAC General industrial/General hydraulics	1mW
Humidity Sensor	HIH-4000 Series	Wired	HVAC equipment Medical equipment	1mW
	Series WHP	Wireless	HVAC Transport	0.25mW

Sensor node power consumption

- 20mW is required for transmitting sensor data
- Sensor :15 μW ~ 1mW
- Wireless data transmission : over 10mW ~ 15mW (Zigbee)

Figure 6. Required energy for wireless monioring

Regarding the larger sized thermo element of the axle box bearing area, the size is Ø= 25.4 ㎝, width= 2 ㎝, thickness=1 ㎝, which is the area of the temperature sensor attached on the axle box bearing. Theoretically, the energy harvesting power of 63.8mW can be obtained; so if more thermo elements are attached on the axle box bearings of high-speed trains though optimum design, hundreds mW of electric power can be obtained, and thus by using the energy harvesting power it will be possible to operate the wireless sensor network for railroad monitoring.

IV. CONCLUSION

This study was to confirm the applicability of energy harvesting technology, with new energy resources from the environment, for wireless sensor networks in railroad systems. We measured the temperature and thermal deviation during high-speed train operation, and estimated the energy harvesting power by thermal deviation on the axle box bearing.

1) The thermal deviation occurred in the axle box bearing when a train runs; the deviation was approximately 12.97～14.68° C

2) As a result of calculating the energy harvesting power by the thermal deviation, about 1.6mW power was obtainable in the rectangular thin film thermo element on the axle box bearing during the operation: 4 ㎠ area, 1 ㎝ height.

3) Concerning the larger sized thermo element of the axle box bearing area, the size is Ø= 25.4 ㎝, width= 2 ㎝, thickness=1 ㎝, which is the area of the temperature sensor attached on the axle box bearing. Theoretically, the energy harvesting power of 63.8mW can be obtained; so if more thermo elements are attached on the axle box bearings of high-speed trains though optimum design, hundreds of mW of electric power can be obtained, and thus by using the energy harvesting power it will be possible to operate the wireless sensor network for railroad monitoring.

ACKNOWLEDGMENT

Some experiments were supported by the Korea Railroad Corporation. All of that support is highly appreciated.

REFERENCES

[1] Gottfried Kure, “Condition monitoring: the apotheosis of maintenance,” International Railway Journal, pp. 42-43, 2009

[2] Oriane Gatinl, “Wireless Sensor Networks Opportunities,” Energy Harvesting & Storage Europe Conference, 2010

[3] Jaehoon Kim and Jae-Youn Lee, “A Feasibility Study on the Energy Harvesting Technology for the Real-Time Monitoring System of Intelligent Railroad Vehicle,” Journal of KSME, B, v35, pp. 955-960, 2011

[4] Yen Kheng Tan and Sanjib Kumar Panda, “Review of Energy Harvesting Technologies for sustainable wireless sensor network,” Sustainable wireless sensor networks, pp. 15-43, 2010

[5] James W. Stevens, “Optimal design of small ΔT thermoelectric generation systems,” Energy Conversion and Management, v42, pp. 709-720, 2001

[6] Poudeu PF, D'Angelo J, Downey AD, Short JL, Hogan TP, Kanatzidis MG, “High thermoelectric figure of merit and nanostructuring in bulk p-type Na1-xPbmSbyTem+2,” Angew. Chem. Int. Edn 45, pp. 3835-3839, 2006

[7] G. Jeffrey Snyder and Tristan S. Ursell, “Thermoelectric Efficiency and Compatibility,” Physical Review Letters, v91, n14, 148301, 2003

[8] Hwee-Pink Tan, Pius W. Q. Lee, Winston K. G. Seah, and Zhi Ang Eu, “Impact of Power Control in Wireless Sensor Networks Powered by Ambient Energy Harvesting for Railroad Health Monitoring,” International Conference on Advanced Information Networking and Applications Workshop, pp. 804-809, 2009

[9] Mathúna CO, O'Donnell T, Martinez-Catala RV, Rohan J, O'Flynn B, “Energy scavenging for long-term deployable wireless sensor networks,” Talanta, v75, pp. 613-623, 2008

ISBN: 978-1-61208-207-3

An Effective Coverage Enhancing Algorithm in Directional Sensor Networks

Chiu-Kuo Liang, Yin-Chung Hsu, and Chih-Hung Tsai
Department of Computer Science and Information Engineering
Chung Hua University
Hsinchu, Taiwan, R.O.C.
e-mail: {ckliang, m09402045, e09902009}@chu.edu.tw

***Abstract*—Directional sensor network is composed of many directional sensor nodes. Unlike conventional sensors that always have an omni-angle of sensing range, directional sensors may have a limited angle of sensing range due to technical constraints or cost considerations. Area coverage problem is still an essential issue in directional sensor networks. In this paper, we study the area coverage problem in directional sensor networks. The problem is to maximize the area coverage of a randomly deployed directional sensor network. Each directional sensor can through rotating orientation to get better coverage in an interested region. We, therefore, propose a greedy algorithm to enhance the area coverage. Simulation results show that our proposed algorithm outperforms the previous proposed method in term of the coverage area.**

Keywords-directional sensors; coverage; greedy algorithms.

I. Introduction

In recent years, wireless sensor networks (WSNs) have received a lot of attention due to their wide applications in military and civilian operations, such as environmental monitoring, battlefield surveillance, and habitat monitoring [1], [2]. Therefore, many research topics such as area coverage, routing, and network security [3] about WSNs gain widespread attention. However, area coverage is a fundamental problem in WSNs since it reflects how well the environment is monitored, and serves as a basis for applications such as habitat monitoring and target detection [3], [4]. Most of the past work is always based on the assumption of omni-directional sensors that has an omni-angle of sensing range. However, there are many kinds of directional sensors, such as video sensors [5], ultrasonic sensors [6] and infrared sensors [2]. The omni-directional sensor node has a circular disk of sensing range. The directional sensor node has smaller sensing area (sector-like area) and sensing angle than the omni-directional one.

For obtaining the better performance in directional sensor networks, directional sensors (e.g. cameras) may be able to rotate around a fixed axis to enhance its coverage in sensing radius [7], [8]. Therefore, the coverage region of a directional sensor is determined by both its location and its direction of sensing radius. Those sensors that can rotate their sensing directions are called the rotatable sensors. We define the working direction of a sensor as the direction in which it is currently pointing at. We also call the sensing range of a sensor's working direction as its coverage region. The coverage region of different sensors may be overlapped with other sensors after they are randomly deployed. Thus, we need to schedule sensors to face to certain directions to maximize the covered area of the whole network.

In this paper, our goal is to maximize the area coverage of a randomly deployed directional sensor network. The problem of working direction scheduling to cover maximal regions, called Maximum Directional Area Coverage (MDAC) problem, has been proved to be *NP*-complete [9]. A greedy solution has also been provided through scheduling working directions of sensors. We propose two different algorithms for MDAC problem with rotatable sensors. Simulation results show that both of our proposed algorithms outperform than the previous proposed *Face-away (FA)* algorithm [7].

The remainder of this paper is organized as follows: Related work is discussed in Section II. In Section III, the problem statement and sensing model are proposed. In Section IV, we propose two greedy algorithms for solving the problem. Section V describes the setting of our experiments and the performance metrics. In Section VI, we present experimental results obtained from different perspectives on the number of sensors, the sensing radius and the sensing angle, respectively. Section VII summarizes our findings.

II. Related work

Recently, in directional sensor networks, the coverage problem has been received a lot of attention from many researchers, not only in area coverage but in target coverage as well. The difference between area coverage [10], [11] and target coverage [12], [13] is in the measurement of the coverage performance. In the area coverage problem, we are focused on the coverage performance on the covered region while in the target coverage problem; the coverage performance on the number of covered targets is discussed. In this paper, we pay our attention to the area coverage problem. Therefore, in the following, we only discuss the recent works related to area coverage problem.

Ma and Liu [14] discuss that the number of directional sensors can be deployed to achieve coverage rate p in a distributed directional sensor network (equation (1)). Directional sensors are randomly and uniformly scattered within a given area. Here, R is the sensing radius, S is the given area, and α is the offset angle of the field of view. To be clear, $\alpha R^2 / S$ indicates that a directional sensor can monitor given area that is within its sensing region. Therefore, after N directional sensors are deployed, the probability that covers a given area is represented in

$$p = 1 - (1 - \frac{\alpha R^2}{S})^N. \qquad (1)$$

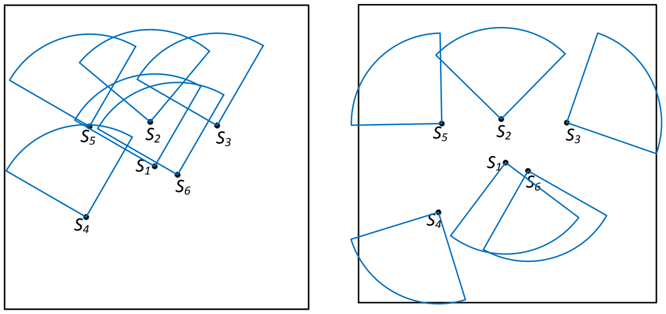
In other words, if the coverage rate of a given area is at least p, the number of deployed directional sensors should be represented in

$$N = \frac{\ln(1-p)}{\ln(S - \alpha R^2) - \ln S}. \qquad (2)$$

Kandoth and Chellappan [7] proposed a greedy solution called the *Face-Away (FA)* algorithm to achieve the maximal area coverage rate in the interested region. The *FA* algorithm works in a very simple manner. Each sensor calculates a new working direction that only needs the positions of neighboring sensors. The neighboring sensors of a directional sensor, say *s*, are those sensors located within the circular area centered at *s* with sensing radius *R*. In fact, every sensor should be recognizable from its surroundings when being viewed by its neighbors. Once a sensor is recognized, each sensor must center it in view and record the current working direction.

In Fig. 1(a), there are six sensors randomly deployed in this network, namely s_1, s_2, s_3, s_4, s_5 and s_6, each has an initial working directions. According to the *Face-away* algorithm, each sensor computes the position of its neighbors (the distance of *R*) by communicating with its neighbors. Each sensor will decide their working direction after rotating its angle to candidate point. The final result is shown in Fig. 1(b), where it can easily be seen that sensor s_1 are overlapped with s_6, and sensor s_2 is overlapped with s_3.

According to the above example, we can see that the *Face-away* (*FA*) algorithm cannot obtain better performance in term of the area coverage since there are still many overlapped area after scheduling by the *Face-away* algorithm, as shown in Fig. 1(b). In this paper, we propose a greedy algorithm to improve the performance of *Face-away* (*FA*) algorithm. The detailed procedure of our algorithms will be discussed in section IV.



(a) An initial deployment (b) Final result of *Face-away*

Figure 1. An example of *FA* algorithm.

III. Directional Sensing Model

In this section, we describe the directional sensing model and notations for the Maximum Directional Area Coverage (MDAC) problem. In a directional sensor network, each directional sensor cannot sense the whole circular area. Therefore, from the concept of field of view in cameras, we can employ a 2-D model where the sensing area of a sensor is a sector denoted by 4-tuple (P, R, $\vec{W}$, α). Here P is the location of the sensor node, R is the sensing radius, $\vec{W}$ is the working direction and α is the angle of view. The common directional sensing capability for 2D spaces is illustrated in Fig. 2. The special case of this model, where $\alpha = 2\pi$ can be described as omni-sensing model.

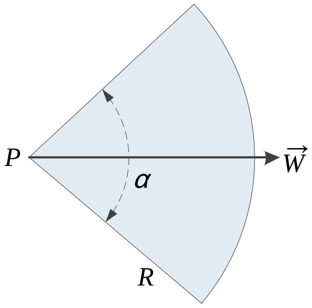


Figure 2. The directional sensing model.

We illustrate the characteristic of directional sensors:

1) Each directional sensor is homogeneous, such as: sensing angle, sensing radius, and communication radius.

2) Each directional sensor can sense only one limited angle of omni-direction.

3) Each directional sensor is fixed and can rotate arbitrary angle in sensing region.

4) The communication radius is twice than the sensing radius such that sensing neighbors can reliably communicate.

IV. The Proposed Greedy Algorithm

In this paper, we are going to improve the previous results in solving the Maximum Directional Area Coverage (MDAC) problem. The MDAC problem is shown to be *NP*-complete [9]. It is unlikely to solve the MDAC problem in polynomial time. Each directional sensor has an initial working direction and it has a lot of overlapped area in an interested region. Fortunately, we can rotate the sensing angle of sensors to avoid the overlapping among sensors which in a result can maximize the coverage area between directional sensors. However, since there is no global information available in a distributed environment, each directional sensor has to make its decision independently only based on its local information gathered from neighboring sensors. As we know that, although the distributed solution cannot be expected to achieve as maximal coverage as the centralized schemes, it is more computational scalable and does not incur high communication overhead as required by a centralized solution. Therefore, the localized solution is more practical and valuable. In this section, we present a distributed greedy algorithm for the MDAC problem.

Although we all know that a sensor can rotate its sensing direction to increase the coverage, we still do not know which direction is the best for a sensor to rotate. Therefore, the main idea of our proposed algorithm is to determine the most possible direction of a sensor to rotate. The following is our strategy for finding the rotating direction. We know that if there is some overlapped area in the whole circular area of two directional sensors, then these two directional sensors may have chances to overlap with each other in their sensing range after rotation. We also know that if some portion of arc on the circumference of a sensor is covered by another sensor, the whole circular area of these two sensors overlap. Therefore, we try to find the possible new direction for a directional sensor to rotate so that the possible overlapped area with its neighboring sensors is minimized. To do so, we evaluate each portion of arc on the circumference of a sensor associated with a weight indicating the degree of likelihood of that portion of arc that may be overlapped with neighboring sensors. Thus, the more weight of that portion of arc can be, the higher possibility of that portion of arc may be overlapped with other directional sensors. In our strategy, a sensor will rotate its direction to the portion of arc with least weight for reducing the overlapped area with other sensors. We describe our idea in detail in the following sections.

Let s_i and s_j be the two directional sensors, R be the sensing radius, and $d(s_i, s_j)$ be the distance between them. Then, we define the *degree of closeness* between sensors s_i and s_j, denoted as C_{ij}, as in the following equation:

$$C_{ij} = \frac{2R - d(s_i, s_j)}{2R}, if\ d(s_i, s_j) \le 2R$$

= 0, otherwise.

Note that, the degree of closeness of two sensors indicates the degree of overlapping, which will influence the size of overlapped area. Obviously, the range of C_{ij} is [0, 1] and as the value of C_{ij} increased, the overlapped area between sensors s_i and s_j also increased.

For convenience, we also define $Arc(P, R, \vec{V}, \alpha)$ to indicate a portion of arc on the circumference of a sensor. Here P is the location of a sensor, R is the sensing radius, $\vec{V}$ is the direction and α is the angle of view. Fig. 3 shows the meaning of $Arc(P, R, \vec{V}, \alpha)$.

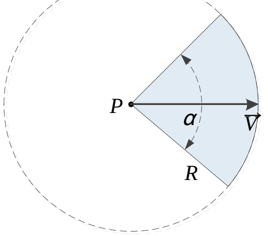
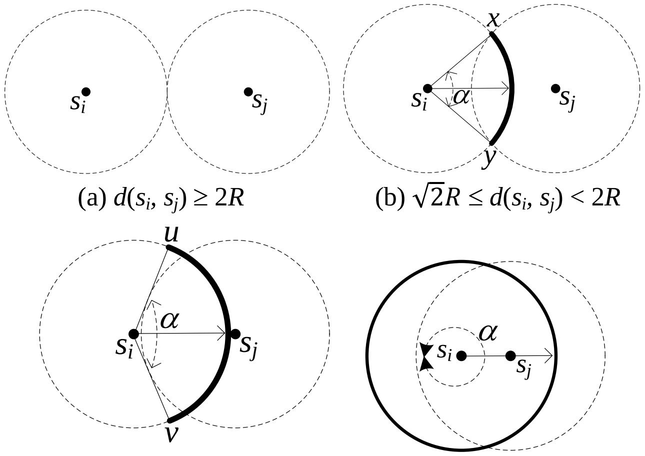


Figure 3. $Arc(P, R, \vec{V}, \alpha)$.

In order to determine which direction for a sensor s_i to rotate to achieve minimum overlapped area with other sensors, we evaluate the weights of the points on the arc of the circle of sensor s_i which can indicate the possibilities of s_i to overlap with another sensor, say s_j, if s_i rotates its working direction to face the new direction of the arc. The weights can be evaluated accordingly based on the following different overlapping situations: (a) $d(s_i, s_j) \ge 2R$, (b) $\sqrt{2}R \le d(s_i, s_j) < 2R$, (c) $R \le d(s_i, s_j) < \sqrt{2}R$, and (d) $d(s_i, s_j) < R$. Fig. 4 shows the different situations. In Fig. 4(a), since there is no overlapped area, the weights of all points on the circle of sensor s_i are zero. In Fig. 4(b), the points on the arc from intersection point x to point y along with the clockwise direction will be evaluated. In Fig. 4(c), the points on the arc from the point u to point v along with the clockwise direction will be evaluated, where u and v are the intersection points of the circle of s_i and the tangent lines from s_i to the circle of s_j. Finally, in Fig. 4(d), all the points on the circle of s_i will be weighted since there exist some overlapped area between sensors s_i and the circle of sensor s_j regardless the rotation of sensor s_i.



(a) $d(s_i, s_j) \ge 2R$ (b) $\sqrt{2}R \le d(s_i, s_j) < 2R$

(c) $R \le d(s_i, s_j) < \sqrt{2}R$ (d) $d(s_i, s_j) < R$

Figure 4. Four overlapping situations between s_i and s_j.

According to Fig. 4, we know that once two sensors, say s_i and s_j, are intersected, then part of the arc on the circles of s_i and s_j should be evaluated for the weight. The weight of a point on the arc can be evaluated as follows. Let $Arc(P, R, \vec{V}, \alpha)$ be an arc on the circle of a sensor , T be a point on the arc and β be the angle between $\overrightarrow{PT}$ and $\vec{V}$. Then the weight of T is evaluated according to the following equation:

$$W(T) = \frac{\alpha/2 - \beta}{\alpha/2} \cdot C_{ij}$$

where C_{ij} is the degree of closeness between s_i and s_j. Note that according to the above equation, the weights of the points on the arc $Arc(P, R, \vec{V}, \alpha)$ will be increasing from the starting point, say s, to the center line $\vec{V}$ and then decreasing to the ending point, say t, as shown in Fig. 5.

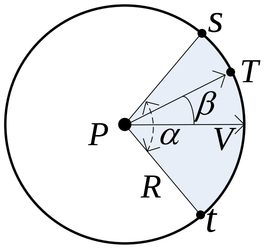


Figure 5. The weights of the points on an arc.

For clarity, we summarize the weight evaluation methodology in this study as follows:

1) $d(s_i, s_j) \geq 2R$. As shown in Fig. 4(a), the weights of all points on the arc of circles of s_i and s_j are zero.

2) $\sqrt{2}R \leq d(s_i, s_j) < 2R$. As shown in Fig. 4(b), the effective arc of sensor s_i is arc $\mathrm{Arc}(s_i, R, \vec{V}, \alpha)$, where α is the angle between $\overline{s_i x}$ and $\overline{s_i y}$, and the weights of points on the arc are computed according to the weighting function.

3) $R \leq d(s_i, s_j) < \sqrt{2}R$. As shown in Fig. 4(c), the effective arc of sensor s_i is arc $Arc(s_i, R, \vec{V}, \alpha)$, where α is the angle between $\overline{s_i u}$ and $\overline{s_i v}$, and the weights of points on the arc are computed according to the weighting function.

4) $d(s_i, s_j) < R$. As shown in Fig. 4(d), the effective arc of sensor s_i is arc $Arc(s_i, R, \vec{V}, \alpha)$, where $\alpha = 2\pi$, and the weights of points on the arc are computed according to the weighting function.

Here, we describe our proposed greedy algorithm for MDAC problem. The proposed algorithm is called the *Maximal Overlapped-Area First* (MOAF) algorithm. It should be recalled that the basic idea of FA algorithm is to find the largest angle between adjacent directions and makes the bisector to that angle as the new working direction. However, the FA algorithm did not take the overlapped area between sensors into account. Therefore, the increasing coverage rate that can be obtained from the FA algorithm is limited. On the contrary, our proposed MOAF algorithm will consider the overlapped area as priority of each sensor. As we mentioned above, we evaluate the weight of points on the circle of each sensor to indicate the possibilities that could be overlapped with other sensors after rotation. In our algorithm, the total weight of the circle of a sensor is considered to be the priority for the sensor. Therefore, if the weight of a sensor is high, meaning that the sensor has many neighbors and the size of overlapped area is large, which as a result the sensor has higher priority to be scheduled for rotation.

Once a sensor has determined to rotate, the new direction can be obtained by finding the point, say A, on the circle with least weight value. Then, the working direction $\vec{W}$ of the sensor will be rotate to the direction of $\overrightarrow{PA}$.

The pseudo-code of the *Maximal Overlapped-Area First* (MOAF) algorithm is shown as follows:

Algorithm: Maximal Overlapped-Area First algorithm

```
I. Initialization Phase (only performed once)
  send a coverage message containing sensor ID and location of sensor $s_i$
  calculate the weights of all points on the arc of circle that is overlapped with its neighbors after waiting for a period of time to collect the coverage messages from sensing neighbors
  determine the priority value $P_i$ and broadcast the value
  collect the priority values from all of its neighboring sensors and go to the Decision Phase.
II. Decision Phase
  while true do
      find the highest priority values, denoted as $P_{max}$, among neighboring sensors
      if $P_i > P_{max}$ then
          find the point, say $A$, on the circle with least weight value
          rotate its working direction to point $A$, set its priority value to 0 and send a scheduled message containing ID and priority value to its sensing neighbors
          Exit the while loop
      else
          if $P_i = P_{max}$ then
              wait for a random duration or a scheduled message sent by a sensor, say $s_j$, is received
              if no scheduled message received then
                  find the point, say $A$, on the circle with least weight value
                  rotate its working direction to point $A$, set its priority value to 0 and send a scheduled message containing ID and priority value to its sensing neighbors.
                  Exit the while loop
              end if
          else
              wait until a scheduled message sent by a sensor, say $s_j$, is received
          end if
          set the status of $s_j$ as "scheduled" and update its priority $P_i$ according to its remaining "unscheduled" neighboring sensors
          send $P_i$ to its "unscheduled" neighboring sensors
          collect the priority values from all of its "unscheduled" neighboring sensors
      end if
  end while
```

V. Simulation Results

This section describes the parameters and performance effects of different perspectives on our proposed algorithms.

We conducted our experiments on a computer with 3.0 GHz CPU and 4GB memory. All experiments are done in C# on .NET platform. Our simulation network consists of 50 to 200 directional sensor nodes placed randomly within a 500 *m* x 500 *m* area. Every experiment was repeated 100 times and the recorded data was averaged over those runs. Table 1 lists the values of the common parameters used in all the experiments.

TABLE I. EXPERIMENTAL PARAMETER

Parameters	Description
Network Size	500 × 500 (m^2)
Sensing Radius	30*m*, 35*m*, …, 60*m*
Sensing Angle	60°,80°, …, 180°
Number of Sensors	50, 75, …, 200

The main goal of our simulation is focused on the evaluation of the performance of our proposed algorithms in term of the coverage rate. The coverage rate *p* is defined as the ratio of the total covered area by all sensors over the network size. We evaluate the effects of our algorithm on three different perspectives. First, we examine the effect that the number of sensors *N* makes to the improvement of coverage rate *p*. Second, we evaluate the effect that the sensing radius improves the coverage rate *p*. Third, we examine the effect that the offset angle makes to the improvement of coverage rate *p*.

Following are our simulation results that demonstrate the effects of our coverage-enhancing algorithms.

VI. SIMULATION RESULTS

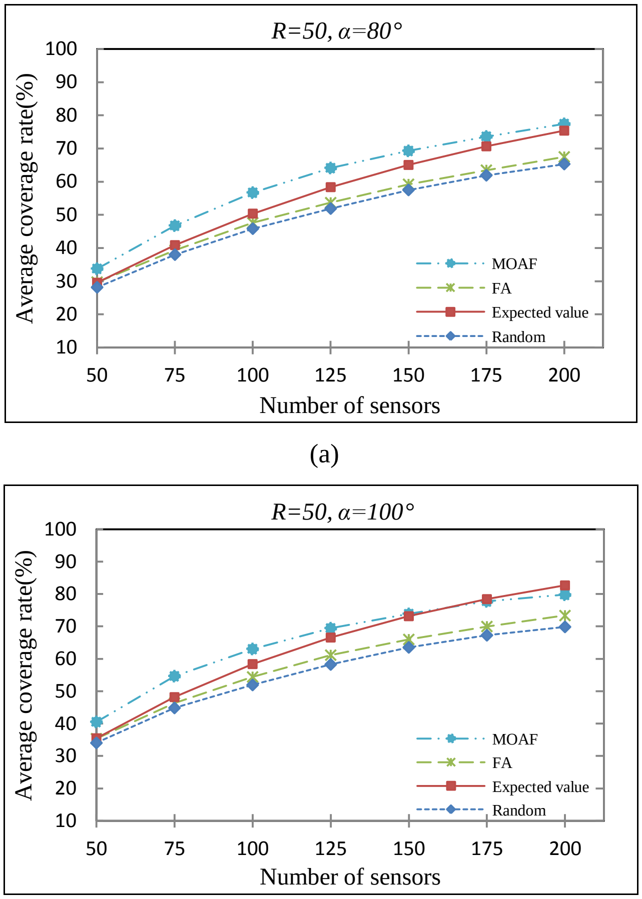
We evaluate the performance of the *Maximal Overlapped-Area First* (MOAF) algorithm, the *Random approach* (Random) in which each sensor select its sensing direction randomly and the *Face-Away* (FA) algorithm. Moreover, we compare the simulation results with the theoretic solution, denoted as *Expected Value*, which are obtained by (1).

A. Coverage rate vs. Number of sensors

This experiment evaluates the effect that the number of sensors *N* makes to the performance of coverage rate *p* of Random approach, FA algorithm, and the MOAF algorithm, respectively. The sensing radius *R* is set to 50m. We first set the sensing angle α to 80°, and then repeated the experiment with sensing angle equals to 100°. The results are shown in Figure 6.

In these graphs, we can see that our proposed MOAF algorithm outperforms Face-away (FA) algorithm and Random approach. For example, when the number of sensors is 200, the sensing radius is 50m and the offset angle is 40°, the coverage rates of Random approach, Face-away (FA) algorithm, and MOAF algorithm are 65.31%, 67.48%, , and 77.42% respectively. Thus, our proposed MOAF algorithm performs 9.94% better than FA algorithm. This is because that our MOAF algorithm can achieve the less overlapping area and the order of sensors chosen to rotate will influence the performance of coverage rate. Therefore, our proposed MOAF algorithm can get the most improvement on coverage rate among all algorithms.

Furthermore, in these graphs, we observe some similar behaviors. We can see from comparing Fig. 6(a) and Fig. 6(b), as the sensing angle α increases (α increases from 80° to 100° in this experiment), the coverage rates of all algorithms increase. This is obvious since the larger the offset angle is, the more area can be covered. Similarly, as the number of sensor nodes *N* increases, the average coverage rate *p* also rises. However, once the value of *N* exceeds a certain value (≥ 150 in this experiment), the increasing coverage rate becomes flat rising. This is because, when the sensing radius and offset angle are fixed, the greater the network density is, the smaller the possibility of uncovered area becomes.



(a)

(b)

Figure 6. Coverage rate – Different sensing angles. (a) α = 80° (b) α = 100°

B. Coverage rate vs. Sensing radius

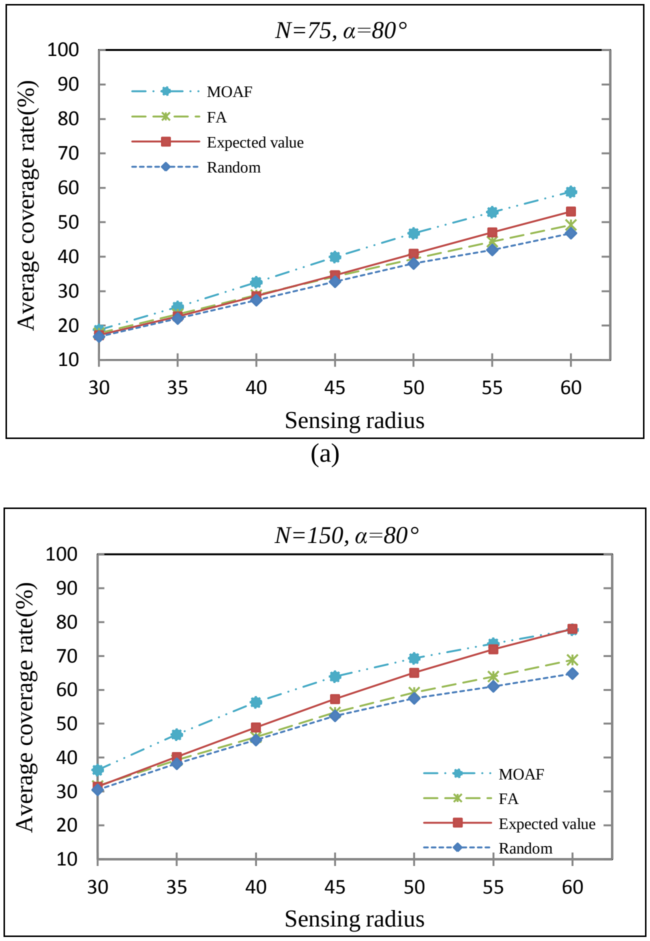
This experiment examines the effect that sensing radius *R* makes to the performance of coverage rate *p* of Random approach, FA algorithm, and the MOAF algorithm, respectively. The sensing offset angle is set to 80°. We first set the number of sensors to 75, and then repeated the experiment with the number of sensors equals to 150. The results are shown in Figure 7.

In these graphs, we can see that our proposed MOAF algorithm outperforms FA algorithm and Random approach. For example, when the number of sensors is 150, the sensing

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-207-3

radius is 50*m* and the sensing angle is 80°, the coverage rates of Random approach, Face-away (FA) algorithm, and MOAF algorithm are 57.50%, 59.22%, and 69.29% respectively. Thus, our proposed MOAF algorithm performs 10.07% better than FA algorithm. This is because that our proposed MOAF algorithm can achieve less overlapping area and higher coverage rate. We also note that, as the sensing radius increases, the coverage rates of all algorithms rise. This is obvious since the greater the sensing radius is, the more sensing area can be obtained.

Furthermore, we can see from comparing Fig. 7(a) and Fig. 7(b), as the number of sensors *N* increases (*N* increases from 75 to 150 in this experiment), the coverage rates of all algorithms increase. This is also obvious to be seen that, when the sensing radius and offset angle are fixed, the greater the network density is, the smaller the possibility of uncovered sensing area becomes.



(a)

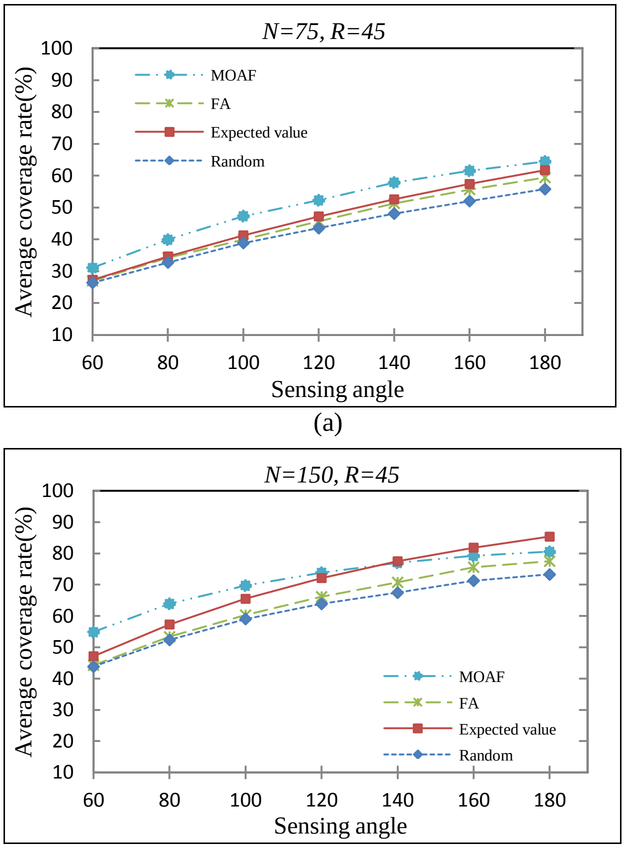
(b)

Figure 7. Coverage rate of sensing radius – Different number of sensors. (a) *N* = 75 (b) *N* = 150

C. Coverage rate vs. Network size

This experiment evaluates the effect that sensing angle α makes to the performance of coverage rate *p* of Random approach, FA algorithm, and the MOAF algorithm, respectively. The sensing radius is set to 45. We first set the number of sensors to 75, and then repeated the experiment with the number of sensors equals to 150. The results are shown in Figure 8.

In these graphs, we can see that our proposed MOAF algorithm outperforms Face-away (FA) algorithm and Random approach. For example, when the number of sensors is 150, the sensing radius is 45*m* and the sensing angle is 60°, the coverage rates of Random approach, Face-away (FA) algorithm, and MOAF algorithm are 63.86%, 66.15%, and 73.86% respectively. Thus, our proposed MOAF algorithm performs 7.71% better than FA algorithm. This is because that our proposed MOAF algorithm can achieve less overlapping area and higher coverage rate. We also note that, as the sensing angle α increases, the coverage rates of all algorithms rise. However, once the value of α exceeds a certain value (≥ 120° in this experiment), the increasing coverage rate becomes flat rising. This is because, when the network density and sensing radius are fixed, the larger the sensing angle is, the smaller the possibility of uncovered area becomes.



(a)

(b)

Figure 8. Coverage rate of sensing angle – Different number of sensors. (a) *N* = 75 (b) *N* = 150

ISBN: 978-1-61208-207-3

VII. CONCLUSION AND FUTURE WORK

In this paper, we investigated the Maximum Directional Area Coverage (MDAC) problem in which we are asked to maximize the area coverage by scheduling the sensing direction or rotating the working direction of each sensor. We propose a greedy algorithm, called the Maximal Overlapped-Area First (MOAF) approach, which is based on the size of overlapped area between directional sensors. Simulation results show that our proposed algorithms both outperform the previous algorithm in terms of coverage rate on different number of sensors, sensing radius and sensing angle. In the future, we will pay our attention to find the solutions for minimizing the energy consumption while maximizing the coverage rate.

REFERENCES

[1] M. Li and Y. Liu, “Underground Structure Monitoring with Wireless Sensor Networks,” Proceedings of ACM/IEEE IPSN, Massachusetts, USA, pp. 69-78, 2007.

[2] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, “An analysis of a large scale habitat monitoring application,” in ACM Conference on Embedded Networked Sensor Systems (SenSys), pp. 214-226, 2004.

[3] M. Amac Guvensan, and A. Gokhan Yavuz, “On coverage issues in directional sensor networks: A survey,” Ad Hoc Networks, vol. 9, no. 7, pp. 1238-1255, 2011.

[4] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, “Coverage problems in wireless ad-hoc sensor networks”, in IEEE INFOCOM, Anchorage, USA, pp. 1380-1387, 2001.

[5] H. D. Ma and D. Tao, “Multimedia sensor network and its research progresses,” Journal of Software, vol. 17, no.9, pp. 2013-2028, 2006.

[6] J. Djugash, S. Singh, G. Kantor, and W. Zhang, “Range-only slam for robots operating cooperatively with sensor networks,” Proceedings of IEEE International Conference on Robotics and Automation, pp. 2078-2084, 2006.

[7] C. Kandoth and S. Chellappan, “Angular Mobility Assisted Coverage in Directional Sensor Networks,” *Proceedings of International Conference on Network-Based Information Systems* (NBIS 2009), pp. 376-379, 2009.

[8] D. Tao, H. D. Ma, and L. Liu, “Coverage-Enhancing Algorithm for Directional Sensor Networks,” Proceedings of 2nd International Conference, Mobile Ad-hoc and Sensor Network, Hong Kong, China, pp. 256–267, 2006.

[9] W. Cheng, S. Li, X. Liao, C. Chen, and H. Chen , “Maximal Coverage Scheduling in Randomly Deployed Directional Sensor Networks” in International Conference on Parallel Processing Workshops, ICPPW 2007, pp. 68, 2007.

[10] Y. Cai, W. Lou, and M. Li, “Target-Oriented Scheduling in Directional Sensor Networks,” in IEEE INFOCOM, pp. 1550–1558, 2007.

[11] X. Han, X. Cao, E. L. Lloyd, and C.-C. Shen, “Deploying Directional Sensor Networks with Guaranteed Connectivity and Coverage,” *Proceedings of 5th Annual 2008 IEEE conference on SECON*, pp. 153-160, 2008.

[12] Y. Cai, W. Lou, and M. Li, “Cover Set Problem in Directional Sensor Networks,” *Proc. of IEEE International Conference on Future Generation Communication and Networking (FGCN ’07)*, pp. 274-278, 2007.

[13] U. R. Chen, B. S. Chiou, J. M. Chen, and W. Lin, “An Adjustable Target Coverage Method in Directional Sensor Networks,” in IEEE Asia-Pacific Services Computing Conference(APSCC 2008), pp. 174–180, 2008

[14] H. D. Ma and Y. Liu, “On coverage problems of directional sensor networks,” in International Conference on Mobile Ad-hoc and Sensor Networks (MSN), pp. 721-731, 2005.

Energy Reduction in Wireless Sensor Networks by Switching Nodes to Sleep During Packet Forwarding

Anne-Lena Kampen[1,2], Knut Øvsthus[1]
[1]Bergen University College, Bergen
[2]ITEM NTNU, Trondheim
Norway
{Anne-Lena.Kampen | knut.ovsthus}@hib.no

Lars Landmark, Øivind Kure
Centre for Quantifiable Quality of Service in
Communication Systems* (Q2S)
NTNU, Trondheim
Norway
{ larsla | okure }@q2s.ntnu.no

***Abstract*—Energy consumption determines the lifetime of Wireless Sensor Networks, WSN. In current radio chip sets the energy consumption for receiving a packet is of the same order as transmitting a packet. In such a setting, the transmission range and sleep strategies should be reevaluated. We present a simple extension to the MAC protocol that reduce the waste of energy for processing packets not addressed to a node by letting them sleep during transmission. The nodes enter sleep mode by means of a Transmission Announcement packet, TAN, sent by the transmitter. The performance is evaluated through simulation. Based on a simplified model, we show that the optimal transmission range in such a setting is given by the minimum needed to avoid partitioning. We use data sheet values from three different WSN Transceiver modules to derive parameter values to be used in the model. The model and related analysis concentrates on the energy consumption in transmitting and receiving, since the radio is the main contributor to energy consumption in WSN. We show that it is the energy consumption in receiving that is the main contributor to total energy consumption in WSN.**

Keywords-WSN; Energy Consumption; Sleep control; Optimal transmission range

I. INTRODUCTIO N

One of the most active research areas in Wireless Sensor Networks (WSN) concerns reduction of the energy consumption of the nodes to increase the lifetime of the WSN. A WSN node consists of several units such as the microcontroller, the memory and the radio, which consumes most energy [1]. Various energy efficient protocols have been proposed to reduce the radio energy consumption. These may be categorized as topology control protocols and sleep management protocols. Topology control protocols use hierarchies and transmission control to limit the number of neighbors (covered nodes) of a node to only those required to avoid network partitioning [2]. This is achieved by reducing the transmission power, and thus shortening the transmission range. But reducing transmission range may degrade the capacity of the network. In their seminal paper, Takagi and Kleinrock determined that the optimal transmission range is when the expected number of neighbors is 8 [3]. However, their work focused on the capacity, and they did not consider the energy consumed listening to packets. Hence, the optimal number of neighbors in order to maximize the lifetime is not evident. Reasons to avoid routing over many short hops are discussed in [4]. Among the listed reasons are interference, energy consumption, path efficiency and end-to-end reliability.

Sleep management protocols schedule redundant nodes to enter sleep mode in order to reduce energy consumption [5]. However, there exist no sharp distinctions between the two mentioned categories, as they may utilize each other qualities to get a more energy efficient network.

Information collected from datasheets for three different WSN Transceiver modules [6][7] shows that the receiving energy consumption is of the same order as transmission energy consumption. In addition, the average number of nodes in a randomly deployed WSN increases quadratic with transmission range, leading to a step increase of energy consumption as transmission range increases. Energy optimization in such a setting requires short transmission range or switching redundant receivers to sleep mode.

The contributions of this article are threefold. First, we present a simple model for calculating the total energy consumption in WSN, taking all the receiving nodes into account. Using the model, we analyze the energy optimal transmission range based on parameters from datasheets for three different WSN Transceiver modules. Last we present a simple energy efficient forwarding approach based on the findings in the analysis. The forwarding approach put redundant nodes to sleep as packets are forwarded.

The rest of the article is organized as follows: related work is introduced in Section 2; the energy consumption model are presented in Section 3; parameter estimation and analysis are done in Section 4; energy preserving forwarding is described in Section 5, and related simulations are presented in Section 6; Section 7 presents the conclusion.

II. BACKGROUND AND RELATED WORK

Two main classes of energy optimization solution are described herein. Sleep management and topology control.

Redundant nodes in a densely deployed network may switch to sleep mode without negatively affecting the communication. Sleep protocols may be divided in two

*Centre for Quantifiable Quality of Service in Communication Systems, Centre of Excellence, appointed by The Research Council of Norway, and funded by the Research Council, NTNU and Uninett (http://www.q2s.ntnu.no)”

groups: local-area-based approaches and backbone-based approaches. In the local-area-based approaches, a node's mode is determined by the mode of the neighboring nodes, and redundant nodes enter sleep mode if it does not negatively affect the connectivity of the network. Examples of protocols in this group are the ones presented in [8][9] [10][11] and [12]. The backbone-based approach selects some nodes to stay active to constitute the backbone of the network. These nodes are responsible for relaying data and scheduling the other nodes to operate in low duty cycles. Clustering is one way of creating a backbone network, in which the clusterhead constitutes the backbone. In LEACH [13], the network is partitioned into clusters and a clusterhead is responsible of organizing the communication in the cluster. Manjeshwar et al. [14] presented an enhanced clustering by letting redundant nodes alternate in handling quires from the clusterhead to avoid unnecessary communication cost. In [15], gridding and clustering are combined in a grid-based clustering technique and the energy-optimizing grid size is evaluated. An overview of sleep management protocols is provided in [5]. Generally, sleep management protocols add synchronization overhead, and are prone to added delay. Our energy optimized forwarding, called Transmitting Announcement (TAN), differentiates from traditional sleep modes. TAN does not require synchronization, and is totally decentralized by simply switching nodes to sleep mode as data packets progress to sink. A detailed description of the approach is given in Section 5.

Topology control approaches adjust the nodes output power to limit the energy consumption of the network. WSN are generally densely populated networks, hence the nodes' output power may be reduced without negatively affecting the connectivity of the network. ATPC [16] proposes a feedback scheme whereby the nodes find the optimal transmission power level for each individual neighbor dynamically. The smallest common transmission power that results in a connected network is found in COMPOW [17], and this power is used by all nodes. CLUSTERPOW [18] integrates routing table information and transmission range to optimize topology control. Dynamic adjustment of transmission range based on node degree is investigated in LINT and LILT [19]. Another example is the one used in CBTC [20], where transmission power is adjusted to reach one neighbor in every sector of a specific degree around the node. A third example is to use graph models, such as used in GG and RNG [21]. They minimize energy consumption by using relay node if this reduces transmission range. An overview of topology control issues and approaches is presented in [2]. Analysis of the energy optimal transmission range is given in Section 4.

There exist several energy consumption models for WSN [22][23][24][25]. However, few of the proposed models consider the receiving energy consumed by the nodes not forwarding packets. These nodes only receive packets to discard them, thus waist energy. The model presented in [26] includes all receivers as data are forwarded from source to destination. However, the distance between the nodes changes as the transmission range change. Hence, the number nodes within the range of a transmitter are constant. The model computes energy consumption for broadcasting. Opposed to the model in [26], the distance between nodes in our model is constant. Hence, the number nodes within the range of a transmitter changes with transmission range. Further, we consider unicast transmission. The energy consumption model is presented in Section 3.

III. ENERGY CONSUMPTION MODEL

Our goal is to investigate the relationship between the nodes transmission range, and total network energy consumption. The aim is to determine the energy optimal transmission range for a given node density. We focus only on the energy used for packet transmission and packet reception. Our scenario is a WSN where the nodes are randomly distributed.

The analysis of energy consumption assesses a source node that is located at a distance D from the sink, without accounting for the network edges. The energy consumption for transmitting data packets depends on the amplifier architecture. A common model for energy consumption per bit has a constant level, k_1, that is independent of the radiated power, plus an offset, k_2, proportional to the radiated power [27]. All nodes have the same transmission range d. Hence, the minimum number of times the packet has to be relayed to reach the sink is D/d. The expression for the energy required to transmit b bit of data is [27]

$$\mathbf{E_{TX} = (k_1 + k_2 * d^2) * \frac{D}{d} * b}. \qquad (1)$$

In addition, we assume that the energy a node uses for receiving data is constant equal to k_3, energy consumed per received bit [27]. The total number of active nodes receiving data is proportional to the density of active nodes, λ, times the area covered by the emission. The consumed energy per bit for one transmission accounting for the number of receivers is thus, $\pi d^2 * \lambda * k_3$. As stated above, the data must be relayed to reach the destination. Hence, the total consumed energy has to be multiplied by the number of times the data is relayed, D/d. The total energy consumed by nodes that receive b bits becomes

$$\mathbf{E_{RX} = (k_3 * \pi d^2 * \lambda) * \frac{D}{d} * b.} \qquad (2)$$

The total energy consumed in relaying the data from the source node to the sink is calculated by adding (1) and (2)

$$\mathbf{E_{TOT} = b * \frac{D}{d}((k_1 + k_2 * d^2) + (k_3 * \pi d^2 * \lambda))}\ . \qquad (3)$$

Our analysis is with respect to optimal transmission range. Constants that have no influence on the result are omitted for simplicity. The expression is normalized with respect to the constant level of the transmission energy.

$$\mathbf{E_{TOT,NORM} = \frac{1}{d} + \left(\frac{k_2}{k_1} + \frac{k_3}{k_1} * \pi * \lambda\right) * d} \qquad (4)$$

By differentiating (4), the energy optimum transmission range is

$$\mathbf{d_{opt}} = \sqrt{\frac{\mathbf{1}}{\frac{\mathbf{k_2}}{\mathbf{k_1}}+\frac{\mathbf{k_3}}{\mathbf{k_1}}*\boldsymbol{\pi}*\boldsymbol{\lambda}}}\ . \tag{5}$$

IV. DATASHEET-BASED ESTIMATIONS

In this section, the parameter values for k_1, k_2 and k_3 are estimated based on values extracted from datasheets, and the optimal transmission range are calculated using these parameter values. Three Transceiver modules are investigated: AT86RF230 [6], CC2420 and CC1000 (_868 and _433) [7]. The datasheet [6][7] provides data for transmission with different output powers, and power measurements for receiving, idle and sleep modes. Power measurements are converted to energy by multiplying with the bit-time calculated from the bit-rate of the Transceiver modules, which is 250 kbit/s for AT86RF230 [6] and CC2420 [7], and the highest bitrate for CC1000 [7] is 76.8 kbit/s.

The parameters k_1 and k_2 are estimated based on the relationship between transmission range and output power, which may be expressed by rearranging Friis [28] equation:

$$\mathbf{P_r} = \frac{\mathbf{P_t}*\mathbf{G_t}*\mathbf{G_r}*\left(\frac{\mathbf{c}}{\mathbf{f}}\right)^{\mathbf{2}}}{\mathbf{16}*\boldsymbol{\pi}^{\mathbf{2}}*\mathbf{d^n}} \tag{6}$$

Rearranging (6) gives:

$$\mathbf{d} = \left(\frac{\mathbf{P_t}*\mathbf{G_t}*\mathbf{G_r}*\left(\frac{\mathbf{c}}{\mathbf{f}}\right)^{\mathbf{2}}}{\mathbf{16}*\boldsymbol{\pi}^{\mathbf{2}}*\mathbf{P_r}}\right)^{\mathbf{1/n}} \tag{7}$$

The parameters used in these equations are as follows. P_r is the power received by an antenna through free space, P_t is the transmitted power, G_t is the transmitting antenna gain, Gr is the receiving antenna gain, c is the speed of light and d is the distance between the antennas. The red curves in Fig. 1 are plotted using (7) with datasheet values for P_r and P_t, using antenna gain of 1.64, which is the gain of a half wave dipole antenna, and choosing path loss exponent n=3 [29][30]. The curves show output power versus transmission range. To find k_1 and k_2 we need to define the red curves by their corresponding second order equations as k_1 and k_2 represent the parameter values in these second order equations (multiplied by bit-time to convert form power to energy). We use curve fitting to find the equations.

The middlemost of the blue dotted curves in Fig. 1 presents calculated curve fitted lines. The equations for these curves are presented in the respective display. Multiplying the parameter values in these equations with bit-time gives k_1 and k_2. The other two dotted curves show the fitted curve with a +/- 10% change of parameter values, indicating that the real values for k_1 and k_2 are within +/-10%.

The receiving power consumption is illustrated by the straight green line. k_3 is derived by multiplying receiving power consumption and bit-time.

Based on the equations for the curve fitted line for CC1000_868, the values for k_1, k_2 and k_3 are 36.1μJ/bit, 0.06pJ/bit/m2 and 37.5 μJ/bit respectively. Choosing λ=0.1 active nodes/m^2 give an optimal d=1.75m using (5). The average number of covered nodes is then 0.96. Performing the same calculations for CC2420, AT86RF230 and CC1000 433 gives optimal distances of 1.2, 1.4 and 1.7, and average number of covered nodes of 0.45, 0.62 and 0.96, respectively. The required number of neighbors to avoid partitioning is 4 according to the discussion presented in [31] that is based on results from [32][33]. As the calculated number of neighbors is lower than 1, the network is partitioned. Hence, using the energy optimal transmission distance, d, would probably lead to network partitioning.

A. *Critical parameters regarding energy consumption*

In order to present a clear understanding of the critical parameters determining the energy efficient transmission range, the derivative of the total energy consumption (4) with respect to range is rearranged as:

$$\left(\frac{\mathbf{k_2}}{\mathbf{k_1}}\right)\mathbf{d^2} + \left(\frac{\mathbf{k_3}}{\mathbf{k_1}}\right)*\boldsymbol{\pi}*\boldsymbol{\lambda}*\mathbf{d^2} = \mathbf{1} \tag{8}$$

The term, $\pi d^2 * \lambda$, is equal to the number of active nodes receiving data. Clearly, there must be at least one active receiver in order to make any progress in forwarding, this implies that $\pi d^2 * \lambda$ must be larger than 1. In (8), this means that there is no real value for d that gives a minimum point if k_3 approaches k_1. Estimations of the parameters based on datasheet [6] and [7] indicate that $k_2 << k_1$, and that $k_1 \approx k_3$,see above. Thus, the receiving energy consumption, k_3, is the main contributor to the short transmission length. The reason is that a linear increase of transmission range, d, causes an increase proportional to d^2 in the number of receiving nodes. This result is consistent with the result of the simulations in [34]. Hence, given that $k_3 \approx k_1$, these findings imply that topology control protocols should aim to reduce the transmission range as much as possible.

Fig. 2 shows how the node density impacts the energy optimal transmission range. Increased node density increases the number of receivers, thus, reducing the optimal transmission range. The values used for the parameters k_1, k_2 and k_3 reflects the relationship between the values as found above.

Keeping the number of receiving nodes constant would reduce the impact of k_3 on the optimal transmission range, and thus the total energy consumption.

V. TRANSMISSION ANNOUNCEMENT ,TAN, USED FOR ENERGY REDUCTION

The analysis in Section 4 shows that the receivers are the main contributor to the total network energy consumption. In WSN, generally all nodes within the transmitter vicinity receive the transmitted packet. However, according to the routing protocol, only one, or a subset, of the receivers are assign to forward the packet. The remaining nodes waste energy as they receive the packet just to discard it.

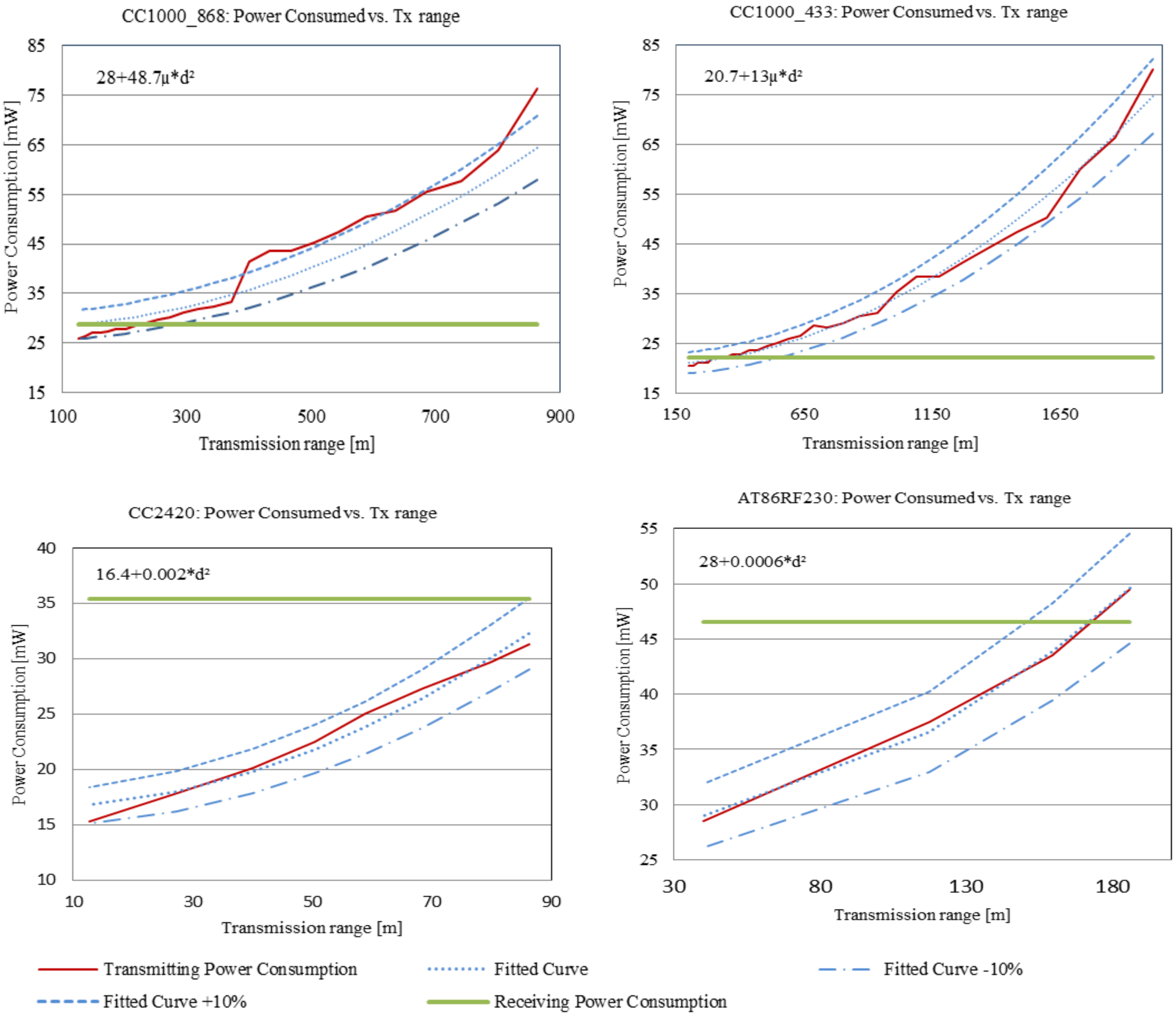


Figure 1. Red curve: power consumption vs. transmission range based on datasheet values. Blue curves: the curve fitted power consumption with +/- 10% change of parameter values. Green curve: receiver power consumption.

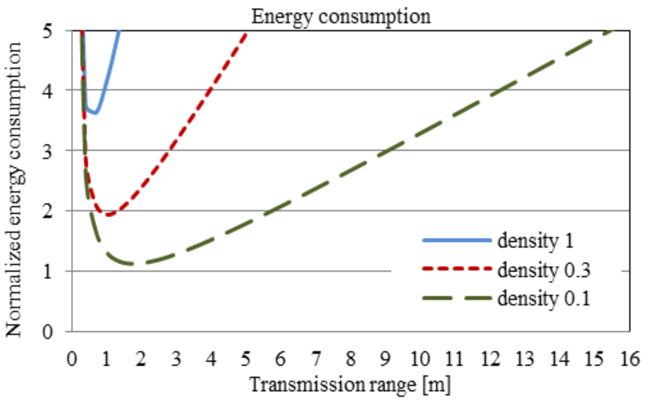


Figure 2. Total normalized energy consumption for sending from a source to the sink. k_1=1, k_2=0.005 and k_3=1.

Our proposal is to reduce energy consumption by preventing nodes form receiving packets not intended to them. This is done by the transmitting node. It prevents nodes from receiving ordinary data packets by sending a short signaling packet prior to the data packet.

The proposed data forwarding approach is as follows. Nodes within the range of the transmitter radio, except for the next-hop node, are switched to sleep mode using a signaling packet called TAN. The packet carries the transmission time for the following data packet, and is addressed to the next-hop node determined by the routing table. All nodes receiving the TAN packet not destined to them change to sleep mode during the corresponding data packet transmission. Radios in sleep mode do not amplify receiving data, which prevents the MAC layer form receiving data. The length of the sleeping period is: (2*SIFS) + (ACK length) + (Data packet length). SIFS is the waiting time between transmitting TAN and the data package, in addition to the waiting time between receiving a data package and transmitting ACK. TAN is only used for unicast transmission, since broadcast and multicast are intended for more than one receiver.

The conditions for TAN to be advantageous compared to plain Carrier Sense Multiple Access (CSMA) depend on: the ratio between data and TAN packet size, node density, and the distance between transmitter and receiver. The requirements on the data packet size are found by estimating the breakeven point when energy consumption using TAN equals the energy consumption using CSMA.

The breakeven point depends on the localization of the receiver inside the sender's transmission range, and two extreme cases are calculated: (1) when the transmitter and receiver share all neighboring nodes (co-located sender and receiver) and (2) when the receiver is localized on the circumference for the sender's transmission range. In the first case, the TAN energy consumption for a one hop communication is: $k*(N+1)*b_{TAN} + k*2*b_{Data} + k*2*b_{ACK}$, where the average number of neighbors is N, $b_{reference}$ is the number of data-bits in the referenced packet-type, and the receiving and transmitting energy consumption per bit is assumed to be equal (k). In the second case, the number of nodes receiving ACK increases, and is exactly those nodes that are inside the area of the receiver's transmission range but outside the sender's transmission range. This crescent shaped area may be calculated based on the formulas described in [31], and the number of nodes in the area is found by multiplying by the node density. Thus, the TAN energy consumption for the second cases is: $k*(N+1)*b_{TAN} + k*2*b_{Data} + k*\left(1 + N - 2\lambda d^2\left(\frac{\pi}{3} - \frac{\sqrt{3}}{4}\right)\right)*b_{ACK}$. The energy consumption using plain CSMA is for both cases: $k*(N+1)*b_{Data} + k*(N+1)*b_{ACK}$.

Based on the equations in the paragraph above and the assumption that ACK and TAN packets size are equal ($b_{ACK}=b_{TAN}$), the breakeven point for case one is:

$$b_{Data} > \frac{2}{N-1} * b_{TAN} \qquad (9)$$

Equation (9) shows for N larger than 3, TAN is advantageous even for data packet smaller than the TAN packet. Note that this occurs for co-located source and destination nodes, which is probably rarely the case as it would result in no progress of the forwarded packet.

By using the fact that $N=\lambda\pi d^2$, the equation for the breakeven point for case two is:

$$b_{Data} > \frac{\frac{N}{3}+1+\frac{\sqrt{3}N}{2\pi}}{N-1} * b_{TAN} \qquad (10)$$

TAN preserves energy, according to (10), if the data packet is smaller than the TAN packet when the number of neighbors is larger than ~ 5.2. On the average, the number of neighbors needed to make TAN energy efficient for data packet size no bigger than TAN packet sizes, lies between these two extreme values, 3 and 5.2. Clearly, the breakeven data packet size is reduced with an increased number of neighbors.

VI. SIMULATIONS

We evaluate our forwarding scheme in an extension of the OMNET ++ simulator [35] with the MiXiM module for wireless communication. The simulator is extended to separate the receiving and idle energy consumption, and to implement TAN. Our simulations are validated against analytic results.

The simulations compare the energy consumption for relaying unicasting traffic, using a plain CSMA MAC layer protocol and our TAN. The comparison is made by measuring the energy consumption when transmitting 1000 data packets from source to the sink. Edge effects are avoided by placing both the source and the sink at a distance from the edge of the network that is longer than the maximum transmission range. Data is transmitted using the maximum 802.15.4 data packet size, 127 bytes at PHY layer [36]. The size of the TAN packet used in the simulations is 30 bytes. Three scenarios with different number of nodes are simulated. The nodes are placed in a random pattern inside an area of 570 x570 m. The distance between source and the sink is 382 m. The presented simulated results are averaged over 30 simulation runs with different seeds for random deployment of nodes. RPL [37] is used for routing, and the routing tables in the nodes are completed before any data is being forwarded.

Simulations performed to compare the total average energy consumption for varying output power levels are shown in Fig. 3. The output power values are chosen based on datasheet values for CC2420. The simulated scenarios consist of 400 nodes. The related 95% confidence intervals are shown in the figure.

Figure 3 shows that the total network energy consumption is lower in TAN than in plain CSMA. In plain CSMA, the number of redundant receivers increases with increased output power. The energy consumption for next highest output power level is higher than for the highest output power level. This counter intuitive result is traced back to a higher hop count that outweighs the increase in the number of covered nodes. The added number of hops resulted in more transmissions draining more energy.

TAN has only one receiver for each transmission. However, there is a tiny increase of energy consumption as output power increases. It is caused by a higher number of receivers receiving the ACK packet sent from the receiving nodes. Similar to the CSMA, TAN experiences an increase in energy for the next highest power. The added energy consumption is caused by the increase in number of hops, and the corresponding number of ACKs. Note that, there is no difference in packet forwarding as routing is equal for both ordinary CSMA and TAN.

The broader 95% confidence interval at the output power level of -15dBm is caused by the larger deviation in path

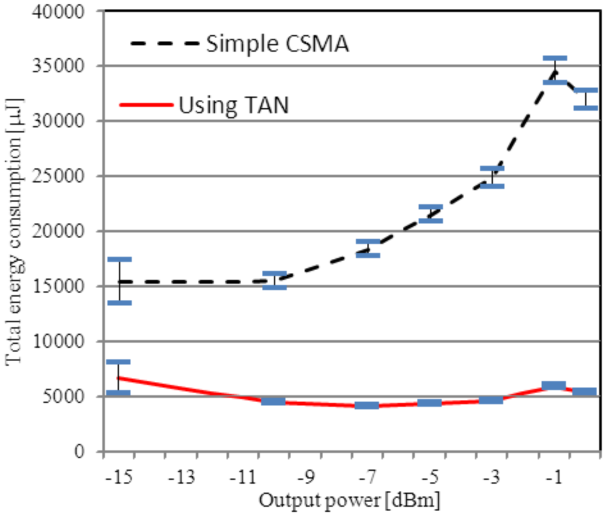


Figure 3. Energy consumption for transmitting 1000 packets in a network consisting of 400 nodes.

length. In addition, due to the low node density, some of the simulations at -15dBm do not have a connection from the sensor to the sink. These simulations are omitted as no results with respect to energy consumption due to data transfer are produced. The 95% confidence interval narrows as the output power increases.

Forwarding energy for different node densities versus transmission range is shown in Fig. 4. As expected, the difference between the plain CSMA and TAN increase with increased node density. This means that the advantage of using the TAN increases as the node density increases. Change of data packet size would give similar results. An increase in packet size would lead to higher difference between the TAN and the plain CSMA.

Simulations for -15 dBm output power are omitted in for the 200 nodes scenario in Fig. 4. The reason is that the network is partitioned for these low output powers.

If the number of neighbors is low, no energy is preserved when using TAN as no redundant nodes receives the transmitted data packets. Hence, if the simulation in Fig. 3 were extended with result for lower output powers, the graphs would eventually merge as the number of neighbors approaches one. Likewise, the graphs in Fig. 4 would merge for very low node densities.

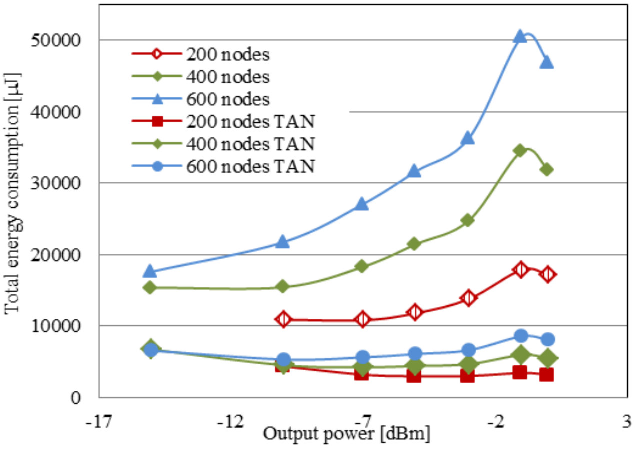


Figure 4. Energy consumption for transmitting with different node densities

Loss of data packet occurs if the intended receiver is in sleep mode caused by TAN packet received from another node. However, these packets would otherwise be destroyed by collision from the ongoing transmission. Thus, the number of lost packets is the same as with CSMA. The solution to avoid losing these packets is to combine TAN with RTS/CTS.

VII. Conclusion

Datasheets for WSN Transceiver modules shows that the receiving and transmitting energy consumption are of the same order of magnitude. Furthermore, the average number of receivers increases quadratic with transmission range in a randomly distributed network. Thus, the energy optimal transmission range is short. We calculate the range using parameter values estimated based on datasheet information. The calculation is performed using an energy consumption model that we present. The range is shorter than the minimum needed to avoid network partitioning. The required number of neighbors to keep a network connected is 4 according to [31] and its references, but the calculated optimal range covers less than one neighboring node. Thus, in order to energy optimize a WSN network the transmission range must be kept just large enough to ensure a connected network.

However, if the number of receivers is fixed, the receiving energy consumption is also fixed. Hence, we propose a solution that reduces the number of receivers to consist of only the next hop node towards the sink. The solution is a simple sleep management approach that makes redundant nodes switch to sleep mode during transmission of data packets. A small signaling packet sent prior to the unicast data packets announces the transmission. Simulations compare the proposed approach against simple CSMA using the maximum 802.15.4 packet size. The simulations show that there is a great reduction in total energy consumption when using the proposed approach. The energy savings depends on data packet size and node density.

References

[1] Enz, C.C., Scolari, N., and Yodprasit, U.: Ultra Low-Power Radio Design for Wireless Sensor Networks. IEEE Proceedings. International Workshop on Radio-Frequency Integration Technology, pp. 1-17. (2005)

[2] SANTI, P.: Topology Control in Wireless Ad Hoc and Sensor Networks. ACM Journal, Computing Surveys, Vol. 37, No. 2, pp. 164-194. (2005)

[3] Takagi, H. and Kleinrock, L.: Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals. IEEE Transactions on communications, Vol. 32, No. 3, pp. 246-257. (1984)

[4] Haenggi, M.: Twelve Reasons not to Route over Many Short Hops. IEEE Vehicular Technology Conference, Vol. 5, pp. 3130-3134. (2004)

[5] Wang, L. and Xiao, Y.: A Survey of Energy-Efficient Scheduling Mechanismsin Sensor Networks. ACM Journal, Mobile Networks and Applications, Vol. 11, No. 5, pp. 723-740. (2006)

[6] Atmel http://www.atmel.com/Images/doc5131.pdf. (April 24, 2012)

[7] Chipcon Products from Texas Instrument:
- CC2420: http://www.ti.com/lit/ds/symlink/cc2420.pdf. (April 24, 2012)
- CC1000: http://focus.ti.com/lit/ds/symlink/cc1000.pdf. (April 24, 2012)

[8] Xu, Y., Heidemann, J., and Estrin, D.: Geography Informed Energy Conservation for Ad Hoc Routing. ACM Proceedings of the 7th annual International Conference on Mobile computing and networking, pp.70-84. (2001)

[9] Ye, F., Zhong, G., Cheng, J., Lu, S., and Zhang, L.: PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks. IEEE In Proceedings of 23rd International Conference on Distributed Computing Systems, pp.1-10. (2003)

[10] Bulut, E. and Korpeoglu, I.: Sleep scheduling with expected common coverage in wireless sensor networks. ACM Journal Wireless Networks, Vol. 17, No. 1, pp. 19-40. (2011)

[11] Yardibi, T. and Karasan, E.: A distributed activity scheduling algorithm for wireless sensor networks with partial coverage. ACM Journal, Wireless Networks, Vol. 16, No. 1, pp. 213-225. (2010)

[12] Kumar, S., Lai, T.H., and Balogh, J.: On k-Coverage in a Mostly Sleeping Sensor Network. ACM Proceedings of the 10th annual international conference on Mobile computing and networking, pp. 144-158. (2004)

[13] Heinzelman, W.B., Chandrakasan, A.P., and Balakrishnan, H.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on Wireless Communications, Vol. 1, No. 4, pp.660-670. (2002)

[14] Manjeshwar, A., Zeng, Q.A., and Agrawal, D.P.: An Analytical Model for Information Retrieval in Wireless Sensor Networks Using Enhanced APTEEN Protocol. IEEE Transactions on Parallel and Distributed Systems, Vol. 13, No. 12, pp. 1290-1302. (2002)

[15] Zhuang, Y., Pan, J., and Wu, G.: Energy-Optimal Grid-Based Clustering in Wireless Microsensor Networks. 29th IEEE International Conference on Distributed Computing Systems Workshops, pp. 96-102. IEEE. (2009)

[16] Lin, S., Zhang, J., Zhou, G., Gu, L., He, T., and Stankovic, J.A.: Atpc: Adaptive Transmission Power Control for Wireless Sensor Networks. ACM, In Proceedings of the 4th international conference on Embedded networked sensor systems,pp.223-236. (2006)

[17] Narayanaswamy, S., Kawadia, V., Sreenivas, R.S., and Kumar, P.R.: Power Control in Ad Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW protocol. In Proceeedings of European Wireless Next Generation Wireless Networks: Technologies, Protocols, Services and Applications, pp. 156-162. (2002)

[18] Kawadia, V. and Kumar, P.R.: Power Control and Clustering in Ad Hoc Networks. IEEE, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, Vol. 1, pp. 459-469. (2003)

[19] Ramanathan, R. and Rosales-Hain, R.: Topology Control of Multihop Wireless Networks using Transmit Power Adjustment. IEEE, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 2, pp. 404-413. (2000)

[20] Li, L., Halpern, J.Y., Bahl, P., Wang, Y.M., and Wattenhofer, R.: A Cone-Based Distributed Topology-Control Algorithm for Wireless Multi-Hop Networks. IEEE/ACM, Transactions on Networking, Vol. 13, No. 1, pp.147-159. (2005)

[21] Jaromczyk, J.W. and Toussaint, G.T.: Relative neighborhood graphs and their relatives. IEEE Proceedings of the IEEE, Vol. 80, No. 9, pp. 1502-1517. (1992)

[22] Wang, Q., Hempstead, M., and Yang, W.: A Realistic Power Consumption Model for Wireless Sensor Network Devices. IEEE, Sensor and Ad Hoc Communications and Networks, pp. 286-295. (2006)

[23] Gogu, A., Nace, D., and Challal, Y.: A note on joint optimal transmission range assignment and sensor deployment for Wireless Sensor Networks. IEEE, Telecommunications Network Strategy and Planning Symposium, pp. 1-6. (2010)

[24] Xing, G., Lu, C., Zhang, Y., Huang, Q., and Pless, R.: Minimum Power Configuration for Wireless Communication in Sensor Networks. ACM, Transactions on Sensor Networks, Vol. 3, No. 2, Article 11, pp. 1-33. (2007)

[25] Zhang, R. and Gorce, J.M.: Optimal Transmission Range for Minimum Energy Consumption in Wireless Sensor Networks. IEEE, Wireless Communications and Networking Conference, pp. 757-762. (2008)

[26] Ingelrest, F, Simplot-Ryl, D, and Stojmenovi, I.: Optimal Transmission Radius for Energy Efficient Broadcasting Protocols in Ad Hoc and Sensor Networks. IEEE, Transactions on Parallel and Distributed Systems, Vol. 17, No. 6, pp. 536-547. (2006)

[27] Heinzelman, W.R., Chandrakasan, A., and Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. IEEE, Proceedings of the Hawaii International Conference on System Sciences, pp.1-10. (2000)

[28] Friis, H.T.: A note on a simple transmission formula. In Proceedings of I.R.E., pp. 254-256. (1946)

[29] Andersen, J.B., Rappaport, T.S., and Yoshida, A.: Propagation Measurements and Models for Wireless Communications Channels. IEEE, Communications Magazine, Vol.33, No. 1, pp.42-49. (1995)

[30] Seidel, S.Y. and Rappaport, T.S.: 914 MHz Path Loss Prediction Models for Indoor Wireless Communications in Multifloored Buildings. IEEE, Transactions on antennas and propagation, Vol.40, No.2. pp.207-217. (1992)

[31] Kleinrock, L. and Silvester, J.: Optimum transmission radii for packet radio networks or why six is a magic number. Proceedings of the IEEE National Telecommunications Conference, pp.431-435, (1978)

[32] Erdos, P. and Renyi, A.: On Random Graphs I. Publications Mathematicae, Debrecen 6, pp. 290-297. (1959)

[33] Dewitt, H.: The Theory of Random Graphs with Applications to the Probabilistic Analysis of Optimization Algorithms. Ph.D. Dissertation, Department of Computer Science, U.C.L.A., Los Angeles. (1977)

[34] Dallas, D. P. and Hanlen, L. W.: Optimal Transmission Range and Node Degree for Multi-hop Routing in Wireless Sensor Networks, ACM Proceedings, Workshop on Perfomance Moitoring and Measuremet of Heterogenous Wireless and Wired Networks, pp. 167-174. (2009)

[35] OMNET++ : http://www.omnetpp.org/. (April 24, 2012)

[36] IEEE Std 802.15.4™-2003. Chapter 6, Digital Object Identifier : 10.1109/IEEESTD.2003.94389. (2003)

[37] ROLL IETF Internet-Draft : RPL: IPv6 Routing Protocol for Low power and Lossy Networks. draft-ietf-roll-rpl-19 [work in progress], 13 Mar (2011)

Empirical Models for Predicting Radio Link Quality in Outdoor Deployment Environments

Sally K. Wahba
School of Computing
Clemson University
Clemson, SC 29634-0974
Email: sallyw@cs.clemson.edu

Jason O. Hallstrom
School of Computing
Clemson University
Clemson, SC 29634-0974
Email: jasonoh@cs.clemson.edu

***Abstract*—In this paper, we present two environment-specific models for predicting radio link quality in embedded wireless systems as a function of radio transmission power and inter-node distance. The models are empirically-based and developed using regression analysis. The underlying data was collected from over 1400 experiments conducted in open grass fields and dense forest environments using *Tmote Sky* nodes. We focus on this hardware platform due to its popularity in the domain of wireless sensing. Our models predict radio link quality in typical outdoor deployment environments, and achieve a goodness of fit of over 0.83.**

Keywords-Radio link quality modeling; wireless sensor networks; embedded network systems; radio link quality prediction.

I. Introduction

Large-scale embedded network systems have moved from imagination to reality. Applications of these systems vary from social networking to saving lives in the battlefield [3, 7, 15], and the domain is still evolving.

Although the community is growing, and interest in the field is increasing, the domain is still in its infancy. Developing embedded applications that behave as expected is a challenge. The main issue is the lack of tools available at the design stage of the application life cycle to help developers implement sound applications. In particular, there are few tools to assist in predicting radio link quality within an embedded network. This in turn leads to designers implementing applications that suffer from unpredictable and undesirable wireless performance [12].

We present two empirical models of radio link quality based on experiments conducted in common deployment environments (*i.e.*, an open grass field and a dense forest). We use *packet reception rate* (PRR) as the link quality metric, defined as the ratio of the number of messages received to the number of message sent. Using regression analysis, the resulting models predict the radio link quality within an embedded network based on transmission distance and radio power level. The models are based on data collected using *Tmote Sky* motes, a widely adopted hardware platform in the domain of wireless sensing.

Our models are different from existing empirical models in two significant ways. First, existing models predict link quality based on data collected from indoor testbeds [6]. Link quality in indoor deployments often varies significantly from link quality in outdoor deployments [9]. Our models rely on empirical data collected from open field and forest environments to predict link quality in such environments. These environments represent a large number of wireless sensor deployments. Second, existing models require users to perform their own empirical analyses (*e.g.*, measuring signal decay, noise fading), which limits ease-of-use [5, 6, 17]. Our models are more straightforward. Users select the deployment environment (*i.e.*, open grass fields and dense forests), radio power level, and inter-node distance. The models are then used to predict link quality based on these parameters.

The remainder of the paper is organized as follows. Section II discusses elements of related work. Section III presents the process for collecting data for our models. Section IV describes the process of filtering the data. Section V presents the link quality models. Finally, Section VI concludes and discusses future work.

II. Related Work

Seada *et al.* [10] study various energy-efficient forwarding strategies for routing in lossy wireless sensor networks. The authors conclude that PRR is a good metric for evaluating channel conditions. Accordingly, we use PRR as the link quality metric in our models.

Liu and Cerpa [6] use a three-phase model for predicting the estimated link quality between nodes. The authors collect PRR data from two indoor testbeds. The data is used as input for training a prediction model using machine learning. This model is then used by the network to adjust message routing. This approach suffers from two main drawbacks. First, the authors rely on indoor data to train their model, which may render the model inaccurate for outdoor deployments. Second, the authors require users to provide their own models for various network conditions.

Kamthe *et al.* [4] describe a statistical approach to modeling link quality variation over time. The authors collect PRR data from a testbed in an indoor lab. The data collected is used as a training set for a learning algorithm used to predict packet reception over time. The authors use two statistical models simultaneously, one for modeling short-term link quality dynamics, and another for long-term dynamics.

This work is different from ours in that the authors provide a statistical model to describe the *correlation* between successive packet failures and receptions. Our work provides a model for predicting radio link quality (*i.e.*, the probability that a packet sent on a link will be received).

Xu and Lee [16] present a regression-based algorithm for on-line estimation of link quality using spatial correlation of nodes. While the model can be used to dynamically adjust routing protocols, it relies on the network being deployed before estimating link quality. The drawback of this approach is that a better network layout can be achieved given a priori knowledge of link quality.

Cerpa *et al.* [1] provide a probability density function that characterizes the relationship between distance and link quality. The authors use the absolute physical location of nodes on a grid, as well as the relative physical proximity of nodes and their neighbors to represent distance. One drawback of the model is that the authors do not account for physical obstructions, although the data was collected from experiments run in environments that include physical obstructions.

Finally, Wahba *et al.* [14] provide a model for predicting radio link quality as a function of inter-node distance and radio transmission power. The model is limited to open grass fields that are obstruction- and interference-free. With the introduction of TinyOS 2.x, a new radio MAC protocol was adopted, which renders the TinyOS 1.x-based model obsolete. Consistent with the findings of Cerpa *et al.* [2], the authors observe that link quality generally falls into three regions: low, mid, and high. Low- and high-quality regions tend to be stable over time; mid-quality regions tend to be unstable. In other words, when PRR is in the high- or low-quality region, it tends to be immune to temporal changes and minor equipment adjustments (*e.g.*, distance and orientation). However, when PRR is in the mid-quality region, it exhibits significant variability, by as much as 100%.

III. DATA COLLECTION

Our work began with preliminary studies to control for factors that are not included as independent variables in our models. We conducted the first study to understand the noise floor in the environments where the experiments were conducted. This allowed us to select the radio channel with the least noise variability. In the second study, we investigated the effect of height differences between communicating devices (because we place the motes on risers). In the third study, we investigated the effect of device orientation on communication quality. The last study involved investigating the transmission rate used in our experiments to ensure that the chosen rate did not lead to network saturation and packet loss. The results of these studies appear in [13]. We use these results in the data collection process to ensure data quality.

To collect the empirical data for our models, we developed two nesC applications designed to run on three motes. The first, DC_1, is designed to run on a sender, S, and a receiver, R. The second application, DC_2, is designed to run on a noise floor data collector, N, and samples the RSSI register during data collection. We measured noise floor during our experiments to ensure that there was no significant interference affecting the quality of our data.

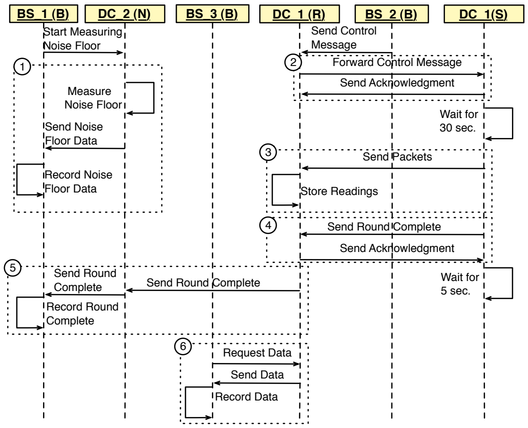


Fig. 1: Data Collection Sequence Diagram

We also developed three Java applications designed to run on a basestation, B. The first, BS_1, controls noise floor data collection. The second, BS_2, controls the main transmission experiment. The third, BS_3, controls data collection after the transmission experiment is complete.

To run an experiment, we first install DC_1 on S and R, and DC_2 on N. S, R, and N wait for a control message from B to start execution. (N and R are connected to the basestation via serial.) Figure 1 shows a sequence diagram representing the data collection process. The numbers in the figure highlight operations that can be repeated, as explained later. (It is worth mentioning that operations on the timeline are not to scale.) We first run the BS_1 application, which sends a serial message to N to start measuring the noise floor. Upon receiving the message, N begins to continuously measure the noise floor on the channel used by S and R, and sends the information to B. Operation (1) repeats for the duration of the experiment. N also marks the end of transmission for each experimental configuration for S and R. (This is discussed later.)

BS_2 and BS_1 begin at the same time. BS_2 sends a serial message to R with the parameters for the experiment, and then terminates. This message includes the following information: *(i)* initial radio power level, *(ii)* final radio power level, *(iii)* transmission rate, and *(iv)* period duration (for each power level). R, in turn, sends this control message via radio to S. Operation (2) repeats until R receives an acknowledgment. S then waits for 30 seconds before it starts sending the desired messages for the first radio power level. After the 30 seconds are over, operation (3) repeats. In this operation, S sends radio messages at the specified transmission rate for the specified duration. Throughout, R counts the number of messages received. When S finishes sending its messages for the first radio power level, it sends a message to R indicating that it has finished that power level. Operation (4) repeats until S receives an acknowledgment from R. R then saves the information associated with that power level in a corresponding array location and clears its counters. R also sends a message to N indicating that the messages for the first power level are complete. N, in turn, sends a serial message to BS_1, which records an end of round marker. S then waits for 5 seconds before it starts sending the next sequence of messages using the next radio power level. These steps repeat until the final power level is reached. At this time, N continues to measure the noise floor for another 30 seconds before it terminates. After noise floor data collection is complete, BS_1 terminates. Accordingly, operations 1, 3, 4, and 5 are repeated for each experimental setup (*i.e.*, distance). Operation 2 executes at the beginning of each trial.

To communicate the information stored on R to B, BS_3 executes. This program sends a serial message to R to instruct it to send the information for each power level. R, in turn, sends this information; the basestation saves it to a file and terminates. Operation (6) represents the steps performed at the end of each trial to collect all data points stored on R.

It is worth noting that each data point represents the PRR for a certain radio power level and inter-node distance value (within a specific deployment environment). For each data point (PRR), S sends 30 messages per second for 30 seconds.

A. *Experimental Parameters*

Here, we summarize the experimental parameters considered in our data collection experiments.

Radio Power Level. We ran the experiments using all available power levels (1 – 31), in increments of 1 unit. These values correspond to non-linear changes in dBm, ranging from less than -37 to 0 dBm, respectively [11].

Deployment Environment. We ran the experiments in an open grass field at Clemson University, and in a forest environment in the Clemson Experimental Forest. Figure 2 shows a typical deployment in the open grass field. Figure 3 shows a typical deployment in the forest. Note from the figures that S, R, and N were placed on 4-foot wooden stakes to avoid the effects of physical obstructions from the grass. Further, N and R were placed 6 inches apart. All motes had the same orientation. For the experiments conducted in the forest, we ensured that the motes were placed within a clear line of sight. We chose these two environments as they are typical outdoor deployment environments. The open grass field represents an outdoor environment that is free of physical obstructions, while the forest represents an outdoor environment containing physical obstructions (*i.e.*, trees).

Inter-node Distance. The distance between nodes was varied between 1 – 416 feet. Note that according to [11], the radio range is 410 feet. The distance increment applied in each step was based on the radio power level. For power levels 1 – 3, the distance was varied in increments of 2 feet. For power levels 4 – 16, the distance was varied in increments of 8 feet. For power levels 17 – 31, the distance was varied in increments of 16 feet. Without the increment variation, the number of experiments would have been prohibitively large.

Figures 4 and 5 illustrate the distances covered for each power level in the field and the forest, respectively. The distances covered in the forest are relatively sparse. From the experiments, no messages were received beyond 256 feet, as opposed to 416 feet in the open grass field. Accordingly, the furthest distance covered was 280 feet. In the dense forest, experiments were conducted over 20% of the distances covered in the grass field.

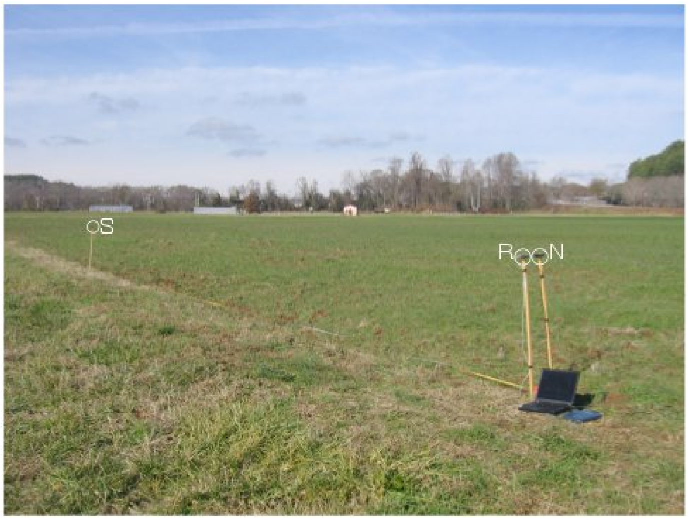


Fig. 2: Field Deployment for Data Collection

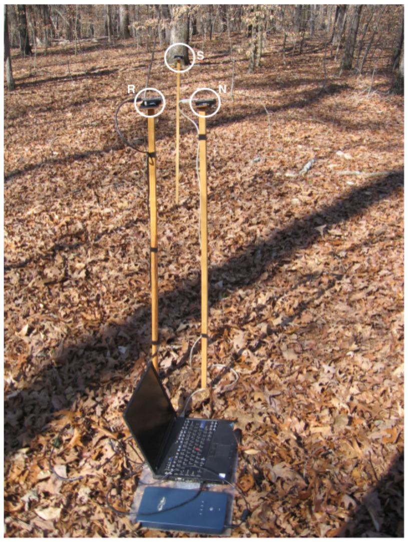


Fig. 3: Forest Deployment for Data Collection

IV. Data Processing

After all of the experimental data was collected, we applied a filtering process to ensure data validity. The first step was to eliminate outliers; the second was to eliminate data points collected during noise spikes.

We explain the process of eliminating outliers with example data shown in Figure 6. If a data point was different from the two "surrounding" data points by more than 20%, the experiment was repeated. For example, in Figure 6a, the PRR data point at a distance of 4 feet differs from the preceding distance (2 feet) and the succeeding distance (6 feet) by more than 20% (surrounding distances were within 20% of one another). Hence, the experiment at distance 4 feet was

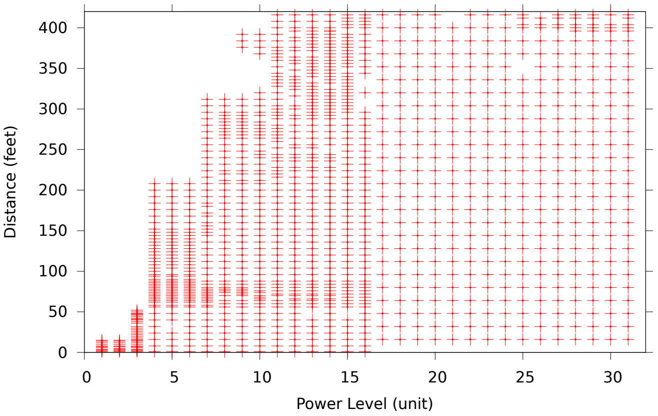


Fig. 4: Distances Covered in the Field

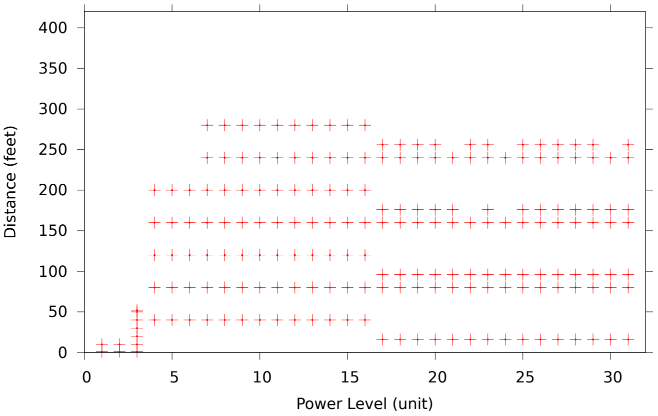
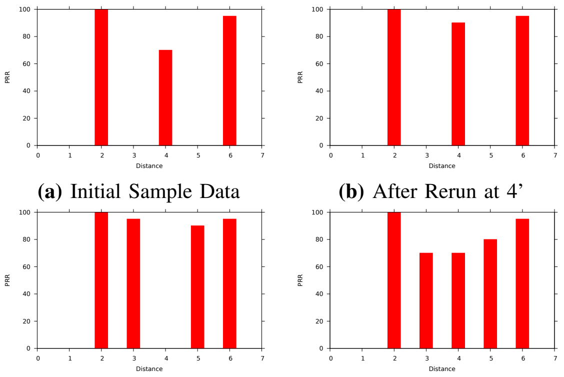


Fig. 5: Distances Covered in the Forest



(a) Initial Sample Data

(b) After Rerun at 4’

(c) After Run at 3’ and 5’ (Case 1)

(d) After Run at 3’ and 5’ (Case 2)

Fig. 6: Filtering Data based on Experiment Reruns

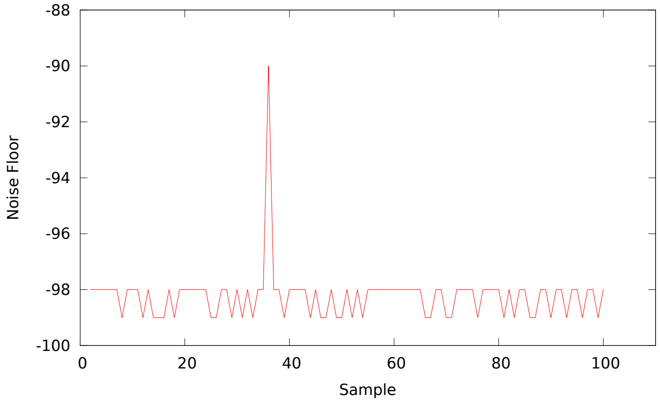


Fig. 7: Example of the Noise Filtering Process

repeated. If the new data was consistent with the surrounding data, the old data was replaced with the new data, as shown in Figure 6b. However, if the new data was still inconsistent with the surrounding data points, two more experiments mid-way between the outlier data point and the surrounding distances were run (*i.e.*, at distances of 3 and 5 feet), respectively. If the data points were within 20% of distances 2 and 6, we discard the data at distance 4, but keep the data from distances 3 and 5. This is the case in Figure 6c. However, if the data was still inconsistent, all three data points were kept (*e.g.*, distances 3, 4, 5 feet), as these data points were not outliers. This is the case in Figure 6d. This approach resulted in 67 reruns and 21 outliers in the field, and 17 reruns and 2 outliers in the forest.

When examining the noise floor data, we noticed that some experiments were associated with noise spikes, suggesting external interference in the environment. When examining some of the corresponding PRR data, we noticed that they were inconsistent with the PRR data collected from the surrounding distances. At these instances, the noise spikes were more than twice the standard deviation of the noise samples.

From the noise floor data and the associated PRR data collected, it was not feasible to determine whether the spikes in the noise floor occurred at the same time a transmission was sent/received, hence affecting the collected PRR data. In other words, we could not adjust PRR to account for noise spikes. Accordingly, we discarded the PRR data collected when spikes in the noise floor occurred.

To eliminate the data associated with spikes in the noise floor, we implemented a Java application that processed all noise samples collected during our experiments. If we found a noise spike above a certain threshold (*i.e.*, twice the standard deviation above the series), we discarded the PRR associated with the experiment during which the noise spike occurred.

To determine a noise spike, we consider all the noise floor measurements collected during a single experimental setup (*i.e.*, one inter-node distance and radio power level). We describe the process of determining a noise spike through an example. Figure 7 represents an example of 100 noise floor samples. The X-axis represents the sample number. The Y-axis represents the noise floor measurement in dBm. The data is divided into segments of 10 noise floor samples each, resulting in a total of 10 segments. For the sake of exposition, we refer to these segments as segments 1 to 10. The cumulative average and standard deviation are maintained, from the start of the series through the end of the last processed segment. For example, when processing the noise floor samples in segment 4, the average and standard deviation for all data points from segments 1 – 3 are maintained. After reading the noise floor samples associated with segment 4, the average of the noise floor samples in that segment is calculated (*i.e.*, the average across noise floor samples 31 – 40). If the average of the noise floor samples in segment 4 is greater than the cumulative

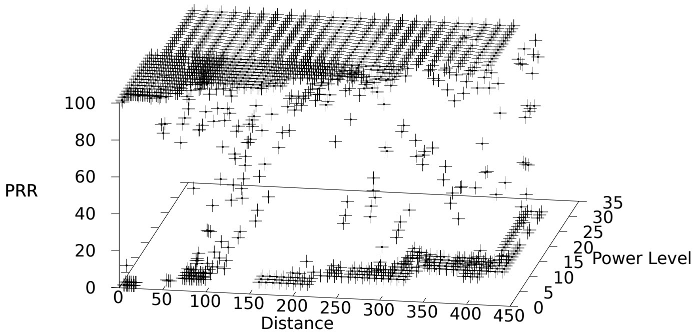


Fig. 8: Data Collected from an Open Grass Field

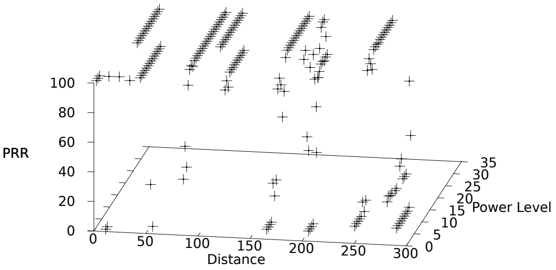


Fig. 9: Data Collected from the Experimental Forest

average plus twice the cumulative standard deviation, a noise spike is identified. When processing segment 4 in Figure 7, the cumulative average (*i.e.*, the average for segments 1 – 3) is -98.33 dBm, and the cumulative standard deviation is 0.479. (The minimum value the RSSI register can store is ≈-100 dBm [11].) The average of the noise floor samples in segment 4 is -97.4 dBm, which is greater than the cumulative average plus twice the cumulative standard deviation. Accordingly, a noise spike is detected. This is consistent with the noise floor samples in the figure, since the noise floor at sample 35 is -90 dBm, which is indeed a spike in the noise floor. At this point, when a noise spike is identified, the PRR data associated with the experimental setup is discarded. This process resulted in 56 data points being discarded from the field experiments and 4 data points being discarded from the forest experiments.

After filtering the data using both procedures, we were left with a total of 1,211 and 213 data points for the field and forest, respectively.

V. Link Quality Models

We used the processed data to design two environment-specific empirical models that predict link quality as a function of radio power level and inter-node distance. Figures 8 and 9 summarize the processed data collected from the field and forest, respectively. Consistent with our findings from [14], the link quality falls into three regions – high-, mid-, and low-quality. We characterize the high-, mid-, and low-quality regions as the regions where PRR is between 90% – 100% inclusive, 90% – 10% exclusive, and 10% – 0% inclusive, respectively. Figure 10 shows a comparison of the data collected from the field versus the data collected from the forest. Notice that the data is consistent in the high- and low-quality regions. However, in the mid-quality region, some data points in the forest have a higher PRR than their counterparts in the field and vice-versa. We suspect the mid-quality region as the cause.

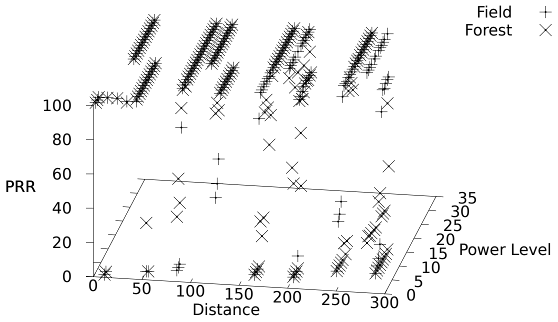


Fig. 10: Field vs. Forest Data

Using one algebraic model to predict link quality is not feasible given the variation among regions. Accordingly, for each deployment environment, we provide a model for the low-quality region, and another for the high-quality region. (Designers usually try to avoid the mid-quality region due to its increased variability [2].) To determine the data points needed for each model, the data was processed as follows. For each distance, we identified the value a, corresponding to the highest power level that resulted in a low PRR. Similarly, we identified the value b, corresponding to the lowest power level that resulted in a high PRR. For the field data, this resulted in 60 and 63 samples for a and b, respectively. The number of a and b data points was different because at shorter distances, some b values did not have corresponding a values. In some cases, PRR was in the high-quality region from the lowest radio power level. For the forest data, this resulted in 11 samples for both a and b.

For each deployment environment, we used the a values and linear regression analysis to determine the appropriate formula for the low-quality region. The process resulted in the following formulae:

$$Field: power = 2.213 + 0.0289 * distance \quad (1)$$

$$Forest: power = -0.1674 + 0.0514 * distance \quad (2)$$

These formulae predict the highest radio power level resulting in a PRR belonging to the low-quality region for a given distance. The R^2 values were 0.85 and 0.892 for the field and forest, respectively.

Similarly, we used the b values and linear regression analysis to determine the appropriate formula for the high-quality region. This process resulted in the following formulae:

$$Field: power = 3.307 + 0.0341 * distance \quad (3)$$

$$Forest: power = 1.4278 + 0.06155 * distance \quad (4)$$

These formulae predict the lowest radio power level resulting in a PRR belonging to the high-quality region for a given distance. The R^2 values were 0.848 and 0.762 for the field and forest, respectively.

In these models, radio power level is measured in discrete units, from 1 – 31, and distance is measured in feet. From the R^2 values, the models achieve a good fit to the actual

ISBN: 978-1-61208-207-3

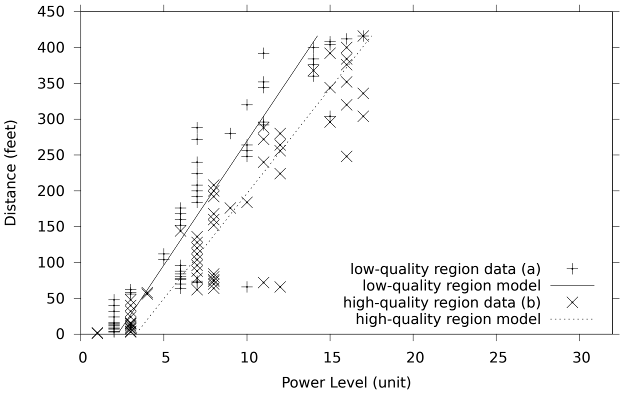


Fig. 11: Link Quality Model Compared to Field Data

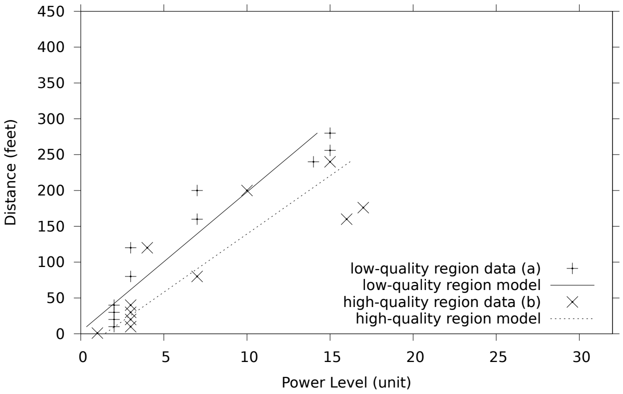


Fig. 12: Link Quality Model Compared to Forest Data

data. Figure 11 shows the plots for a and b values, along with the corresponding models for the field experiments. Similarly, Figure 12 shows the plots for a and b along with the corresponding models for the forest experiments.

Limitations. Three limitations of the current models should be noted. First, the results are specific to a single hardware platform. We do not expect the models to provide high accuracy for network deployments that employ different hardware platforms, such as those operating outside the Zigbee band (*i.e.*, 2.4Ghz). Second, our models are limited to interference-free environments. Given the effect of interference on network link quality [2, 8], these models are not likely to provide high accuracy in the presence of significant interference. Third, we note that the models assume only a single transmitting process. In scheduled transmission networks (e.g., using TDMA or FDMA) and networks with few concurrent transmitters, the models are directly applicable. However, in the presence of many concurrent transmitters, the accuracy of the models is expected to degrade.

VI. Conclusion and Future Work

The behavior of embedded network systems depends largely on radio link quality. Without a priori knowledge of link quality, reliable system behavior is difficult to achieve. As a result, designers often develop applications characterized by unpredictable wireless behavior.

We have developed environment-specific empirical models for predicting radio link quality as a function of inter-node distance and radio transmission power. The models help developers understand the behavior of radio links in open grass fields and dense forests, which correspond to a large number of network deployments. Hence, designers should be better able to develop applications that yield predictable performance. For example, designers will be able to use the models to determine which node layout and radio transmission power yield high performance for a given deployment site. Additionally, by adjusting radio transmission power, designers will be able to prolong application lifetimes by saving energy without sacrificing performance.

We have two elements of future work to extend our models. First, we plan to measure the degradation in the accuracy of the models when used in the presence of interference. Second, we plan to measure the accuracy of the models in the presence of common occurrences of noise spikes. In other words, we plan to measure the sensitivity of the models to the frequency of noise spikes as opposed to noise threshold – our current approach.

Acknowledgments. This work is funded by the National Science Foundation (awards CNS-1126344 and CNS-0745846). The authors wish to thank Alex Propst and Kevin Poole for their help in collecting the empirical data, as well as Dr. Wayne Goddard and Dr. Pradip Srimani for their assistance with the link quality models.

References

[1] A. Cerpa, J. Wong, L. Kuang, M. Potkonjak, and D. Estrin. Statistical model of lossy links in wireless sensor networks. In *Proceedings of the 4^{th} International Symposium on Information Processing in Sensor Networks*, pages 81–88, Los Alamitos CA, USA, April 2005. IEEE Computer Society.

[2] A. Cerpa, J. Wong, M. Potkonjak, and D. Estrin. Temporal properties of low power wireless links: Modeling and implications on multi-hop routing. In *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 414–425, New York NY, USA, May 2005. ACM Press.

[3] R. Jafari, A. Encarnacao, A. Zahoory, F. Dabiri, H. Noshadi, and M. Sarrafzadeh. Wireless sensor networks for health monitoring. pages 479–481, Washington DC, USA, July 2005. IEEE Computer Society.

[4] A. Kamthe, M. Carreira-Perpi, and A. Cerpa. M&M: multi-level Markov model for wireless link simulations. In *Proceedings of the 7^{th} ACM Conference on Embedded Networked Sensor Systems*, pages 57–70, New York NY, USA, November 2009. ACM.

[5] J. Leskovec, P. Sarkar, and C. Guestrin. Modeling link qualities in a sensor network. *Informatica (Slovenia)*, 29(4):445–452, 2005.

[6] T. Liu and A. Cerpa. Foresee (4C): Wireless link prediction using link features. In *Proceedings of the 10^{th} International Conference on Information Processing in Sensor Networks*, pages 294–305, Washington DC, USA, April 2011. IEEE.

[7] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. Eisenman, X. Zheng, and A. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proceedings of the 6^{th} ACM Conference on Embedded Network Sensor Systems*, pages 337–350, New York NY, USA, November 2008. ACM.

[8] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2^{nd} International Conference on Embedded Networked Sensor Systems*, pages 95–107, New York NY, USA, November 2004. ACM Press.

[9] N. Reijers, G. Halkes, and K. Langendoen. Link layer measurements in sensor networks. In *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 224–234. IEEE, Oct 2004.

[10] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari. Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. In *Proceedings of the 2^{nd} International Conference on*

Embedded Networked Sensor Systems, pages 108–121, New York NY, USA, November 2004. ACM Press.

[11] Texas Instruments. CC2420 2.4 GHZ IEEE 802.15.4 zigbee-ready RF transceiver data sheet (rev. 1.3). http://www-s.ti.com/sc/ds/cc2420.pdf, March 2012. (*last access*).

[12] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong. A macroscope in the redwoods. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, pages 51–63. ACM, Nov 2005.

[13] S. Wahba and J. Hallstrom. An empirical analysis of communication links in embedded wireless networks. In *Proceeding of the 49th ACM Southeast Conference*, pages 185–190, New York NY, USA, 2011. ACM.

[14] S. Wahba, K. LaForce, J. Fisher, and J. Hallstrom. An empirical evaluation of embedded link quality. In *Proceedings of the 2007 International Conference on Sensor Technologies and Applications*, pages 430–435, Washington DC, USA, Oct. 2007. IEEE Computer Society.

[15] 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 Int. Comp.*, 10(2):18–25, 2006.

[16] Y. Xu and W. Lee. Exploring spatial correlation for link quality estimation in wireless sensor networks. In *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications*, pages 200–211, Washington DC, USA, March 2006. IEEE Computer Society.

[17] M. Zuniga and B. Krishnamachari. Analyzing the transitional region in low power wireless links. In *Proceedings of the 1st IEEE Conference on Sensor and Ad Hoc Communications and Networks*, pages 517–526. IEEE, Oct 2004.

Non-Invasive Cognitive Radio for Firm Real-Time Sensor Applications in heterogeneous Radio Environments

Horst Hellbrück, Tim Esemann, Uwe Mackenroth, Marius Ciepluch, Arnaud Möschwitzer, Malte Ziethen
Lübeck University of Applied Sciences, Germany,
Department of Electrical Engineering and Computer Science,
Email: {hellbrueck,esemann,mackenroth}@fh-luebeck.de,
{marius.ciepluch,arnaud.matteo.moeschwitzer,malte.ziethen}@stud.fh-luebeck.de

***Abstract*—Some applications in Sensor Networks need firm real-time support in order to work properly. The difference to hard real-time systems is that this type of application can withstand minor violations of the maximum delay and minimum throughput if these violations are limited. Many standards like IEEE 802.15.4 provide standardized means to ensure delay and bandwidth constraints which work well when there are no interferers in the same frequency band. However, in a heterogeneous environment today these approaches fail when the interference is not aware of the IEEE 802.15.4 traffic. Switching the channel is one option to avoid this kind of interference. We suggest a new non-invasive cognitive radio protocol approach where all participants follow simple rules to enable firm real-time conditions in decentralized design. As a demonstrator we use a three-fold pendulum with firm real-time signal delay constraints of 5ms. The contributions of the paper comprise evaluation results by real measurements with the demonstrator system.**

***Index Terms*—Wireless Sensor; Cognitive Radio; Firm Real-time; Protocol;**

I. Introduction

Today's radio standards were continuously improved and engineered in the past. Therefore, new fields of application for wireless systems are enabled [1]. Systems which were impossible to setup can now be implemented. Especially wireless control system can gain from this development. Real-time applications are one example for a new application area of wireless technologies. Such applications can be divided into hard, firm and soft real-time applications [2]. In hard real-time missing a single system deadline results in a total system failure. Firm real-time systems tolerate missing a deadline infrequently until a certain marginal degradation of the system performance. Soft real-time applications accept missing deadlines and the corresponding degradation of the system performance. In our work, we implement a firm real-time application, the control of a three-fold pendulum, as a demonstrator to evaluate the performance of our new developed non-invasive cognitive radio sensor protocol.

Another important effect of the successful radio standards is that more and more wireless systems are deployed for all kinds of applications. With this increasing utilization of radio links the available frequency spectrum becomes a scarce resource. One typical example for such a heterogeneous radio environment is the 2.4GHz frequency band, utilized inter alia by WLAN, Bluetooth and IEEE 802.15.4. Cognitive systems, also known as Cognitive Radios (CR), are currently under investigation [3], [4]. They are aware of the current radio environment and have the potential to reach coexistence of wireless systems even in the future within heterogeneous environments. Our approach uses CR techniques to detect concurrent radio links and adapt transmission parameters accordingly to avoid collisions and reach firm real-time characteristics.

Therefore, contribution of our work is threefold. First, we present a new non-invasive cognitive radio protocol for a sensor application with firm real-time requirements. Our solution is based on a decentralized rule based approach beyond existing reservation and synchronous Time Division Multiple Access (TDMA) schemes. Second, we provide solutions in a heterogonous radio environment by switching to free channels without an additional control channel. We detect own data packets, interfering packets using same wireless technology AND other interfering wireless technologies. Third, we present a firm real-time demonstrator based on 802.15.4 physical layer with delay requirements less than 5ms to evaluate and prove the effectiveness of the approach. To the best of our knowledge, this is the first time that such a non-invasive cognitive radio sensor protocol including demonstrator in such a heterogeneous environment has been presented.

The rest of this paper is structured as follows: Section II will introduce related work and demonstrate the need for new approaches. We will describe our approach in more detail in Section III. Section IV presents our new cognitive radio sensor protocol. In Section V we describe the demonstrator that was used for the evaluation. The results of our evaluation are shown in Section VI. The paper will conclude with a short summary and presents future work in Section VII.

II. Related Work

Cognitive Radios are developed to find and utilize free frequency channels in a heterogeneous radio environment. Haykin [5] defines a Cognitive Radio (CR) as an intelligent wireless system that is aware of its surrounding radio environment. It changes its operating parameters according to the learning from the environment. Akyildiz et al. [3] describe the tasks which are required for an adaptive operation in an open radio environment, referred to as the cognitive cycle. Therefore, a Cognitive Radio has to follow three steps, Spectrum Sensing, Spectrum Analysis and Spectrum Decision. Spectrum Sensing monitors the actual occupancy of the spectrum. According to the monitored and analyzed results (Spectrum Analysis) the CR decides for a suitable dynamic spectrum access (Spectrum Decision). Akan et al. [4] describe the main design principles, potential advantages and application areas for Cognitive Radio Sensor Networks. It is shown that wireless sensor nodes equipped with cognitive radio can benefit from potential advantages of dynamic spectrum access. As already mentioned, one major task of a CR is to perform Spectrum Sensing. A detailed summary of spectrum sensing schemes is given by Yücek and Arslan [6]. Two important schemes - which are also utilized in our work - are energy detection and matched filter. Energy detection simply measures the energy transmitted by other wireless nodes over a corresponding frequency channel to detect if the channel is free or occupied. Matched filter analyzes the received data to check if a channel is occupied by a concurrent radio link with the same radio technology. Therefore, prior knowledge of concurrent radio links is required to demodulate and decode received signals. CRs deploy the Spectrum Decision over a common control channel according to the results of Spectrum Sensing and corresponding Spectrum Analysis.

Another important issue to be solved in our work is to guarantee certain Quality of Service (QoS) requirements by an appropriate MAC protocol. Chen et al. [7] propose an improved low-latency IEEE 802.15.4 MAC protocol. The improvement includes the modification of the superframe structure and the reduction of the MAC overhead. The contention access period is removed within the modified superframe. Therefore, the guaranteed time slots need to be pre-allocated to each of the participating devices. Additionally, the MAC layer data frame overhead is reduced. It only includes a payload of one byte and a frame checksum of two bytes. With these modifications the latency for 20 participating devices can be reduced from 17ms to 8ms. Another example for guaranteeing real-time services with IEEE 802.15.4 is presented by Yoo et al. in [8]. The approach proposes a modification of the guaranteed time slot allocation, but does not reach a real-time interval smaller than 10ms. Both schemes do not consider a heterogeneous radio environment and therefore can react on occurring interference.

The WirelessHART standard was developed by companies to meet the stringent requirements of control applications as presented by Song et al. in [9]. It adopts the IEEE 802.15.4 physical layer. On top of that it defines an own MAC. The MAC layer utilizes CSMA, TDMA and channel hopping to avoid interference with other concurrent radio links. Mesh topology with a network manager which configures the network, schedules and manages the communication between the nodes. Even though it avoids channels with interference it does not meet the requirements of real-time intervals smaller than 10ms.

Our approach follows the idea to avoid channels with interference or other concurrent IEEE 802.15.4 links. This idea, switching to another channel when an interference occurs, was also presented in one of our previous work in a different context in [10].

The control of a single inverted pendulum over an IEEE 802.15.4 wireless sensor and actuator network was presented by Hernandez et al. in [11]. The proposed approach provides only a sampling period of 32ms and also cannot deal with interference on the used frequency channel. The 3fold pendulum in our work needs control cycles less than 5ms. In [12] Yang et al., present the control of even two inverted pendulum, but for wireless communication an IEEE 802.11b radio link was used and also a heterogeneous radio environment was not considered.

III. Approach

To achieve firm real-time requirements in heterogeneous environments we make assumptions about the environment. First, there is free spectrum available, so called holes that change dynamically over time as we cannot solve the problem if there are no resources left. We further assume that the application data traffic is variable bit rate (VBR) where the system without the presence of interference is capable to fulfill the required real-time characteristics like data rate and delay with a certain margin left to tolerate some interference. Additionally we restrict the solution of this problem for single-hop scenarios in this work. Multi-hop scenarios need more complex solutions to provide the required QoS for firm real-time requirements. We plan to extend our approach to multi-hop capabilities in the future. To achieve this goal the system needs to be aware of the environment in the sense of Cognitive Radio Research.

We describe the idea with the help of a scenario where there are many sensor sources and a single data sink. The approach is not limited to this setup but for now this specific scenario helps to understand the design.

Channel Switching: In our approach sources and sinks agree on a channel usage sequence similar to a hopping sequence in frequency hopping. This happens before starting the system and can be dynamically adapted if needed. The predefined channel usage sequence improves the performance of the system as in the case of interference the nodes implicitly agree to hop to the next channel even if the interference does not allow to communicate on the actual channel. In contrast to other approaches previously published that are based on Cognitive Radio, we do not assume a dedicated reliable control channel for resource planning and cooperation.

The channel usage sequence can be extended in the future by blacklisting like Adaptive Frequency Hopping (AFH) does for Bluetooth. The advantage of blacklisting is that interferences of crowded channels can be omitted and the system has a better chance to achieve the QoS requirements of the application.

Detecting Interferences: To avoid ambiguity only one dedicated node in the network listens to the medium to detect interferences of other wireless technologies. This node is called Full Cognitive Node (FCN). Consequently, we name the other nodes Reduced Cognitive Nodes (RCN). If the measured interference of the FCN reaches a limit, this node sends a packet switch RF channel to all nodes. All reduced cognitive nodes in the network listen to these packets continuously and switch accordingly.

Additionally, all cognitive nodes switch channels if they receive a large packet that violates the predefined timing constraints of the application. The latter decision is helpful if other applications on nearby nodes start to send large packets. Optionally, the FCN can send an additional switch RF channel in this case too if delay constraints allow so.

Non-Invasive Behavior: Instead of starting to negotiate resource usage with the other applications we implement a non-invasive strategy by switching to another channel as fast as possible to keep the firm real-time constraints. We prefer this strategy as cooperation between wireless firm real-time applications is standardized only in optional sections of the specifications. Therefore, in real scenarios, we cannot assume that all wireless nodes in range are capable of this cooperation.

In the following, we summarize the approach starting from the detection to the switching rules. All nodes count, measure, analyze received data and apply a matched filter. The FCN additionally detects interference of other concurrent wireless transmissions by energy detection or more advanced schemes. Cognitive nodes switch to a new channel based on a predefined channel usage sequence in either two cases: (1) A large packet or (2) a switch RF channel packet is received. Thereby, we have defined a simple robust decentralized rule based system. We discuss some details of the implementation in the next section.

IV. Implementation Details

In this section, we provide more details of the protocol using flow charts and present the timing model to calculate the parameter settings.

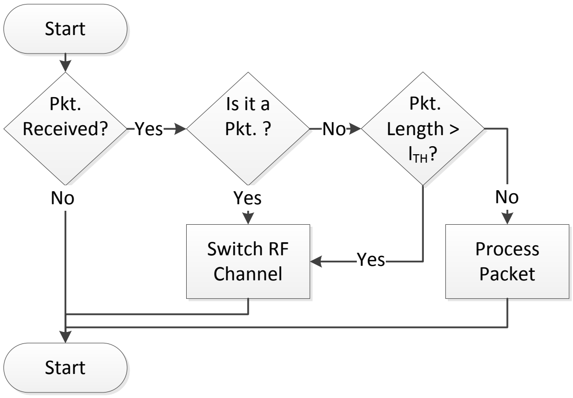


Fig. 1. Flow Diagram of Receiving Process Reduced Cognitive Node

The reduced cognitive node is kept very simple and does not have any timing conditions as can be seen in Figure 1. The only parameter needed for the protocol is the threshold length of a packet l_{TH} in bytes if a fixed data rate is used in the system or in packet duration if adaptive data rates like in IEEE 802.11 occur. In IEEE 802.11 depending on the signal to noise ratio and capabilities the transmission rate is between 1Mbps up to 11Mbps in IEEE 802.11b, but downwards compatible to original IEEE 802.11.

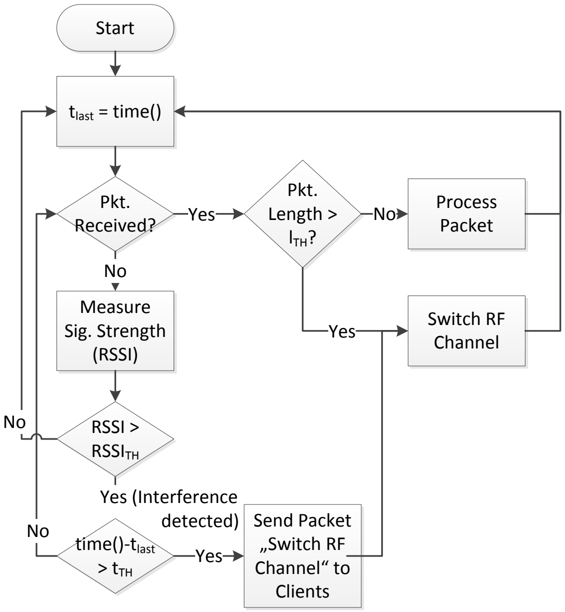


Fig. 2. Flow Diagram of Interference Mitigation Full Cognitive Node

For detection of the interfering signals we need access to the signals on the RF-Chip. We identified the following characteristics for concurrent transmissions and noise or other radio interference that can be measured with common RF-Chips.

- Interfering packets change the RSSI value of the receiver due to the preamble signal.
- Nearby transmitters result in high RSSI values.
- After the complete preamble the RF-Chip enters the state ”receiving”.

In contrast, noise or other radio interference results in the following characteristics.

- Interfering signals change the RSSI value of the receiver.
- Strong interference results in high RSSI values.
- Even with long periods of interference the RF-Chip does not enter the state ”receiving”.

Figure 3 illustrates the typical scenario where for a short time the FCN detects the violation of the critical RSSI threshold and starts a timer. When the interference stops before the timer expired, the FCN resets its timer again.

In Figure 4, the timer expires after the threshold t_{TH} so that the FCN decides to send a packet ”switch RF channel” to all nodes. If the RSSI is suitable to send, the FCN can send this packet successfully and all nodes change channels synchronously.

Figure 5 illustrates the case when interfering packets from other applications using the same channel occur. In Figure 5

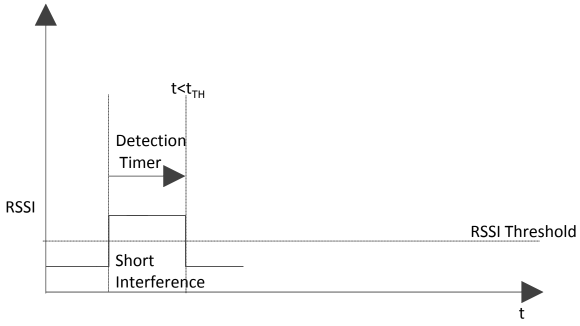


Fig. 3. Short Interference FCN

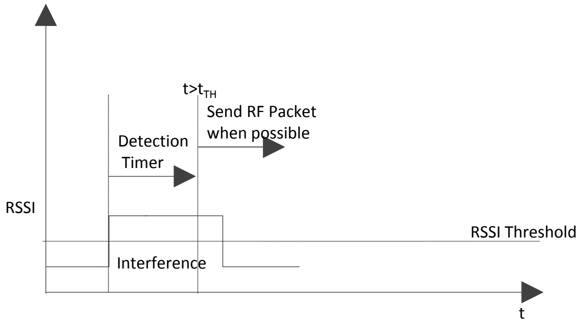


Fig. 4. Long Interference FCN

we see the case where a large packet is transmitted that violates the packet length threshold l_{TH}. All nodes will switch channels accordingly. The Reduced Cognitive Nodes will start to measure with the reception of the packet without the first part where the RSSI value hits the threshold. Table I summarizes the settings for our Protocol.

TABLE I
SETTING FOR NON-INVASIVE COGNITIVE RADIO

Parameter	Value	Description
$RSSI_{TH}$	$> CSMA_{TH}$	Threshold where Interference Detection Timer of FCN is started
t_{TH}	$> t_{maxDelay}$	Threshold when nodes switch channel due to long Interference
l_{TH}	$> t_{maxDelay}$	Threshold when nodes switch channel due to large packets

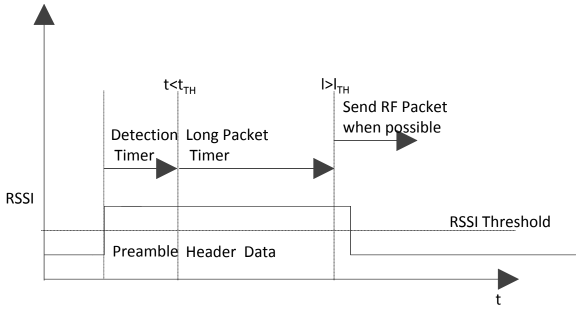


Fig. 5. Large Interfering Packet FCN

V. DEMONSTRATOR

The demonstrator for our new approach is a threefold pendulum which is controlled by a Matlab-driven Target-PC. The control loop of the system consists of sensors measuring angle values from all bearings of the pendulum. On a PC, a Matlab application calculates a control value, accordingly. Due to the weight and the handling of the cables it is not possible to mount wired sensors to each of the bearings. Therefore, wireless sensors with rotary encoders are mounted on each bearing. The wireless sensors in our systems are Reduced Cognitive Nodes as described in Section IV. A Full Cognitive Node collects the angle information from the RCNs and transfers it to the PC via serial interface. The setup of the demonstrator is depicted in Figure 6.

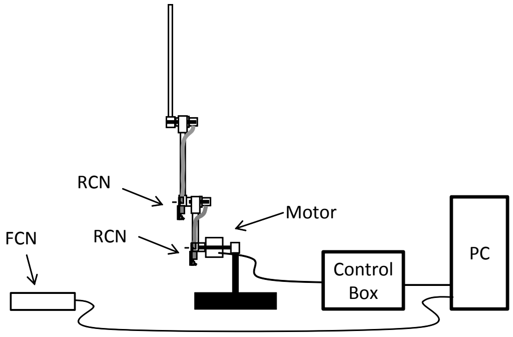
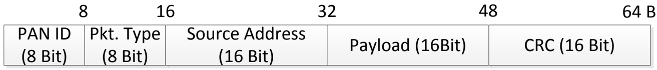


Fig. 6. Demonstrator - 3fold Pendulum

For real-time applications the packet size needs to be small to fulfill the timing constraints. The packet consists of 8 Bytes in five fields as illustrated in Figure 7.



8	16	32	48	64 Bit
PAN ID (8 Bit)	Pkt. Type (8 Bit)	Source Address (16 Bit)	Payload (16Bit)	CRC (16 Bit)

Fig. 7. Packet Format

The PAN ID allows to distinguish own or interfering packets. The packet type can be *Registration Request*, *Registration Response* or *Data*. The source field specifies the originator of the packet and Data as well as CRC contain the obvious content.

As shown in Figure 8, each client working as a RCN starts to search for a FCN node. FCNs operate as a data sink transferring the data to the PC via serial interface. The RCN hops according to the channel usage sequence searching for a FCN that answers its request. In the reply the FCN returns also the timing conditions and could add further optional settings like a new channel usage sequence. Register Reply packets need to be considered also by already connected nodes as timing conditions like maximum waiting time might change during runtime in our setup. The RCN nodes send their data in a round robin fashion in a non-invasive manner by waiting according to the settings of the FCN in the reply packet as illustrated in Figure 9.

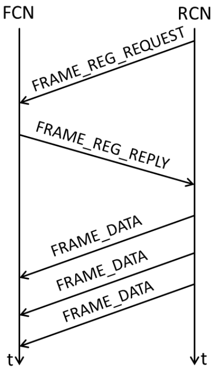


Fig. 8. Protocol

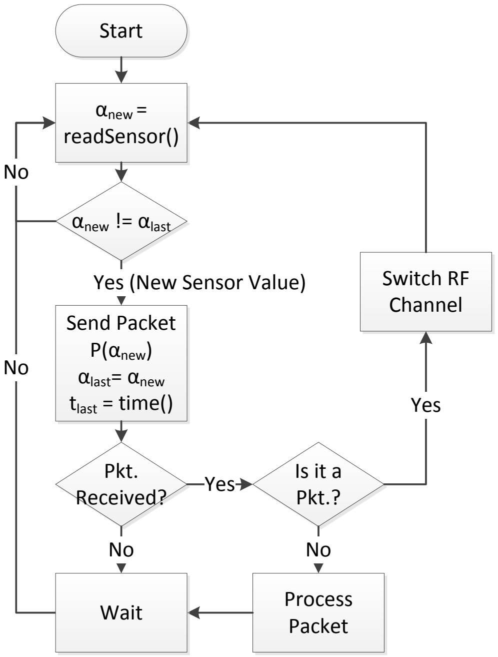


Fig. 9. Complete Flow Diagram of Reduced Cognitive Node

VI. Evaluation

The demonstrator introduced in Section V serves for evaluation of the cognitive radio sensor protocol. It is a very intuitive measurement tool for proofing the keeping of the firm real-time requirements as the violations of the requested max-delay of 5 ms can be noticed by strong control movements of the system. For the evaluation setup an additional interfering wireless sensor was introduced to the radio environment as well as a WLAN IEEE 802.11g access point. The corresponding network topology is shown in Figure 10.

To demonstrate the effect of a concurrent transmitter, the transmitted and captured angle information were displayed in Figure 11 to Figure 13. In Figure 11, both angle values, transmitted by two RCNs and received by the FCN, are displayed in the time interval of 5s. During this reference measurement we switched off all interferences.

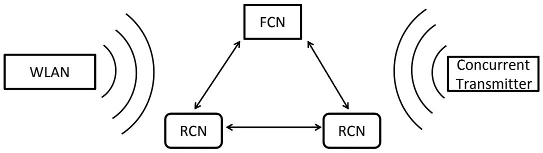


Fig. 10. Radio Environment for Evaluation

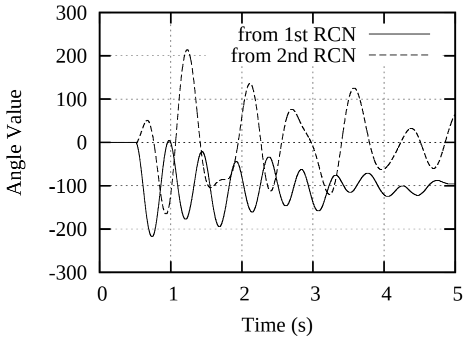


Fig. 11. Complete Measurement Cycle for two Sensors as RCNs

In the next setup, an interfering wireless sensor transmits short packets with 40Bytes every 5ms. The length of packets is shorter than the defined threshold of 50Bytes according to the timing constraints The transmission of 50Bytes with 250kbps bit rate lasts for 50*8b/250kbps resulting in approx. 2ms which is less than the maximum delay of 5ms. The transmission of the nodes themselves also lasts for approx. 1ms including internal processing and RF Chip processing. Therefore, these disruptions can be tolerated by the pendulum and consequently no channel switching is necessary. The influence on the measured angle values can be seen in Figure 12.

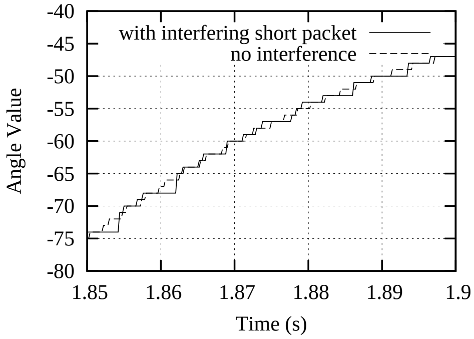


Fig. 12. Received Angle Values from one RCN with and without interfering IEEE 802.15.4 Packets of Size 40Byte

The displayed angle values show two evaluation runs, without interference (dashed line) and with interfering packets with length of 40Byte (solid line). In Figure 12, each 5ms a short distortion of the captured angle values occurs. Due to the fact that these short interfering packets can be tolerated no channels switching was invoked as predicted. This is why we see these effects occur consecutively.

In the last setup, a concurrent wireless sensor transmits packets of size 70Byte which is above the defined threshold. The resulting angle values are depicted in Figure 13.

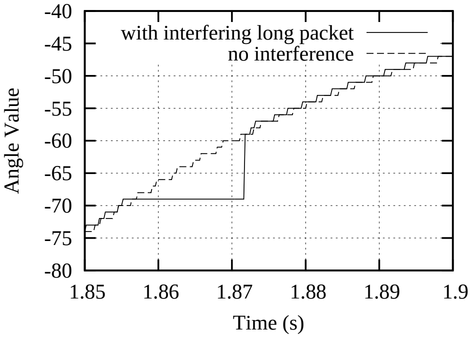


Fig. 13. Received Angle Values from one RCN with and without interfering IEEE 802.15.4 Packets of Size 70Byte

At 1.855s, the concurrent wireless sensor transmits a packet of 70Bytes which causes a distortion of the captured angle value and a corresponding channel switch. After all nodes have switched to the next channel, values are smooth again at time 1.872 and no consecutive distortions of the angle value occur. If the nodes switch to a channel where WLAN is transmitting, measurements show the same behavior as it shows with large interfering packets. After switching to a new channel the values are smooth again.

Figure 14 shows the frequency spectrum in a waterfall plot to illustrate the switching to the next channel.

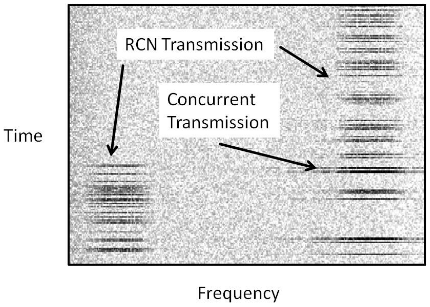


Fig. 14. Switching to the next Channel due to interfering large packets

The top of the plot displays the past whereas the bottom displays the most recent measurements. In the beginning, only the RCNs are transmitting their angle information. As soon as the concurrent wireless sensor starts its transmission, all RCNs and FCN switch to the next channel and proceed with the transmission of the angle information.

The presented evaluation results demonstrate the correct functionality of the system and illustrate the cognitive capabilities of the sensor application.

VII. Conclusion and Future Work

In this paper, we presented a new non-invasive cognitive radio protocol for a sensor application with firm real-time requirements. Our decentralized rule based approach works beyond existing reservation and synchronous Time Division Multiple Access (TDMA) schemes. We demonstrated that we provide solutions in a heterogonous radio environment by a firm real-time demonstrator based on 802.15.4 Our approach detects own data packets, concurrent transmissions using same wireless technology AND other interfering wireless technologies. Finally, we present the control of a pendulum with delay requirements less than 5ms to evaluate and prove the effectiveness of the approach.

For the future, we will complete the protocol with blacklisting of crowded channels and security mechanisms that avoid the risk that unauthorized nodes send packets ”switch RF channel”. Furthermore we will work on multi-hop capabilities.

Acknowledgment

This work has been supported by the Federal Ministry of Education and Research of Germany: Förderkennzeichen 17N3809, SoFT.

References

[1] C. F. Garcia-Hernandez, P. H. Ibargüengoytia-Gonzalez, J. Garcia-Hernandez, and J. A. Perez-Diaz, “Wireless sensor networks and applications: a survey,” *International Journal of Computer Science and Network Security*, vol. 17, no. 3, pp. 264 –273, 2007, survey.

[2] P. A. Laplante, *Real-time systems design and analysis - an engineer’s handbook (3. ed.)*. IEEE, 2004.

[3] I. Akyildiz, W. Y. Lee, and K. Chowdhury, “Crahns: Cognitive radio ad hoc networks,” *Ad Hoc Networks (Elsevier) Journal*, vol. 7, no. 5, pp. ’810–836’, Jul. 2009.

[4] O. B. Akan, O. B. Karli, and O. Ergul, “Cognitive radio sensor networks,” *Netwrk. Mag. of Global Internetwkg.*, vol. 23, no. 4, pp. 34–40, Jul. 2009.

[5] S. Haykin, “Cognitive radio: brain-empowered wireless communications,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. ’201–220’, Feb. 2005.

[6] T. Yücek and H. Arslan, “A survey of spectrum sensing algorithms for cognitive radio applications,” *IEEE Communications Survey & Tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

[7] F. Chen, T. Talanis, R. German, and F. Dressler, “Real-time Enabled IEEE 802.15.4 Sensor Networks in Industrial Automation,” in *IEEE Symposium on Industrial Embedded Systems (SIES 2009)*. Lausanne, Switzerland: IEEE, July 2009, pp. 136–139.

[8] S. eun Yoo, P. K. Chong, D. Kim, Y. Doh, M.-L. Pham, E. Choi, and J. Huh, “Guaranteeing Real-Time Services for Industrial Wireless Sensor Networks With IEEE 802.15.4,” vol. 57, no. 11, pp. 3868–3876, November 2010.

[9] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt, “WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control,” in *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2008)*. St.Louis, USA: IEEE, 2008, pp. 377–386.

[10] T. Esemann and H. Hellbrück, “Non-invasive cognition driven spectrum access in medical application via baseband processing,” in *Proceedings of the 7th Karlsruhe Workshop on Software Radios*, Karlsruhe, Germany, Mar. 2012.

[11] A. Hernandez, J. Faria, J. Araujo, P. Park, H. Sandberg, and K. H. Johansson, “Inverted Pendulum Control over an IEEE 802.15.4 Wireless Sensor and Actuator Network,” in *Proceedings of the European Wireless Sensor Networks (EWSN)*, Bonn, 2011.

[12] A. Yang, G. Irwin, W. Naeem, and K. Li, “Application of wireless network control to a two inverted pendulum system,” in *Proceedings of the 22nd IET Irish Signals and Systems Conference*, Dublin, 2011.

A Study on the Effect of Packet Collisions on Battery Lifetime of 802.15.4 Motes

Jose M. Cano-García, Eduardo Casilari, Farah Adbib
Departamento de Tecnología Electrónica at University of Málaga
Malaga, Spain
cano@dte.uma.es, ecasilari@uma.es

Abstract— **An empirical study of the power consumption of commercial IEEE 802.15.4/ZigBee motes is presented. The analysis investigates the current that is drained by an 802.15.4 based module when the radio channel is occupied or packet losses take place. For this purpose, we developed a simple testbed where problems in the radio medium are emulated in a controlled way. This is accomplished by artificially introducing in the protocol stack of the nodes a probability that a Clear Channel Assessment (CCA) failure or a packet collisions occurs. The results demonstrate the importance of CCA failures and especially packet collisions in determining the consumption of Wireless Sensor Networks with a moderate or high traffic load. Thus, problems related to the occupation of radio channel can easily more than halve the battery lifetime in networks with just some tens of nodes where data must be updated several times per second.**

Keywords-IEEE 802.15 .4/ZigBee; Wireless Sensor Networks; CSMA/CA; Clear Channel Assessmen; packet collision.

I. Introduction

IEEE 802.15.4 is a leading standard in the ambit of wireless sensor networks (WSNs). IEEE 802.15.4 [1] specifications define the physical and Medium Access Control (MAC) layers for networking architectures of low-cost, low-bandwidth, short-range wireless nodes. IEEE 802.15.4 protocols offer the basis for WSN and Personal Area Network (PAN) standards, mainly ZigBee [2] but also ISA100.11a, WirelessHART or MiWi. The 802.15.4 chipset market is growing dramatically. In 2010, ZigBee/802.15.4 chipset shipments nearly doubled while the annual revenues for ZigBee/802.15.4 modules are expected to reach $1.7 billion in 2015 [3]. Applications for smart homes are the main target of these modules, but other fields (such as smart cities, industrial plant-process, medical monitoring or wellness) are gaining attention from vendors and developers.

IEEE 802.15.4/ZigBee compliant nodes (or ‘motes’), which may operate in the ISM 868 MHz, 915 MHz or 2.4 GHz frequency bands, are designed to minimize the power consumption. They are normally battery powered and used in applications where battery replacement is generally unfeasible or too costly.

Aiming at predicting the battery lifetime in a real application of 802.15.4/ZigBee technology, the current drain in the motes must be thoroughly evaluated. In most cases, wireless communications are the main source of battery usage. IEEE 802.15.4 MAC is conceived to switch off the transceiver when no packet has to be transmitted or received. As a result, the nodes may remain in a power-saving (sleep) mode most of the time, so that the batteries can be operative for years However, due to the contention method applied by 802.15.4 to access the medium (CSMA/CA), the consumption of the radio transceiver is strongly related to the status of the radio channel. Thus, if the medium is not found to be idle or a packet collision occurs (because two nodes transmit simultaneously), the protocol may induce a non-negligible increase of power consumption. In this article, we empirically analyze the impact of both the channel occupation and the packet collision on the battery drained by commercial 802.15.4/ZigBee nodes. The paper extends an initial study already published in [4] where the collisions were not taken into consideration.

The paper is organized as follows: Section 2 summarizes the behavior of 802.15.4 MAC as well as the dynamics of CSMA/CA algorithm. Section 3 briefly comments some related literature on 802.15.4/ZigBee consumption. Section 4 details the utilized experimental testbed, while Section 5 shows and discusses the performed measurements. Finally, Section 6 draws the main conclusions of the paper.

II. IEEE 802.15.4 MAC

According to IEEE 802.15.4 specification there are two types of network devices. Full-Function Devices (FFD) may perform as the ‘coordinator’ or central node of a star topology or otherwise interact on a peer-to-peer basis forming a multi-hop mesh network. On the other hand, Reduced-Function Devices (RFDs), which are normally battery powered devices with limited capabilities, can only communicate with its coordinator (residing in a specific FFD).

Additionally, IEEE 802.15.4 MAC layer defines two possible operational modes:

(1) Under the beacon enabled mode, the coordinator node periodically broadcasts a special frame (a beacon) informing about the existence of the network and allowing the synchronization of the ‘children’ nodes. Children nodes must wake-up just in time to receive the beacon from their Coordinator and keep synchronized to the network. After every beacon and during a special period called superframe, packet exchanges between the coordinator and the devices take place. When the superframe is finished, all the nodes (including the Coordinator) can enter into the sleep mode. Thus, battery consumption can be also reduced in the Coordinators (which can also act as intermediate router in a multi-hop 802.15.4/ZigBee cluster-tree). This can be an important issue if Coordinators are also powered by batteries. However, long Beacon Intervals and extended sleep periods (apart from increasing packet delay) may provoke serious problems to keep the nodes synchronized because of clock inaccuracies. In fact, most commercial 802.15.4/ZigBee motes do not support beacon mode presently, most probably due to the difficulty to enable an efficient beacon tracking in the end devices.

(2) Under the non beacon or point-to-point mode, coordinators do not send beacons. As no synchronization

exists, end devices can wake up (from its sleep mode) in any moment to send a data packet to the coordinator. In the opposite sense, if the coordinator wishes to send a data packet to an end device, it has to wait to be polled by the end device with a specific poll frame requesting the data.

Non-beacon mode is more appropriate for networking applications which can be implemented by a simple star topology consisting in a set of wireless sensors/actuators and a Coordinator powered from the main source. In these scenarios (which correspond to many practical cases of WSN applications), the Coordinator can maintain its radio receiver active all the time so it can communicate with any device in any moment. The permanent activity of the Coordinator allows clients to be in a power saving mode for long intervals of time. Thus, the devices can wake up at their will (on a periodic or event-driven basis) just to transmit the sensed data or to poll the Coordinator to check if there is any pending message.

A. CSMA/CA Algorithm

In both the beacon or point-to-point mode, the access is regulated by CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance). This medium access protocol obliges nodes to sense the radio medium before sending a data packet. So, according to IEEE 802.15.4 MAC, nodes willing to transmit data have to contend for the radio channel following the CSMA/CA protocol. Thus a source node must initially delay its transmission a random number of slots or backoff periods of 20 symbols (0.32 ms when the standard works in the 2.4 GHz band with 62.5 Ksymbols/s). This number is selected in the range [0, 2^{BE}-1], where BE is the backoff exponent, a variable that regulates the CSMA waiting process. After this inactive time, the node performs a Clear Channel Assessment (CCA) to test the availability of the radio channel. If the channel is not detected to be free, the BE exponent is increased in one unit (up to a maximum) and the procedure is reiterated. If the CCA operation consecutively fails a predetermined number of times, a channel access failure is assumed and the packet is discarded. On the contrary, if any CCA is successful, the radio transceiver of the node switches from the reception mode to the transmission mode (as 802.15.4 communications are half-duplex) and the data frame is emitted.

Normally, the packet is only considered to be adequately transmitted if a specific acknowledgment packet (ACK) is received from the target node within a certain time interval. This ACK response is emitted by the destination upon the reception of the data packet as long as CSMA/CA algorithm does not apply for ACK packets. However, both the data packet or the ACK itself may not arrive properly because of a transmission error or a packet collision. Collisions can be produced by the activity of other 802.15.4 nodes or by interfering devices performing in the same 2.4 GHz band. When the acknowledgment is not received, the source node iterates the whole CSMA/CA process (resetting BE to its initial value). The number of times that the transmission can be repeated is also bounded by the specification. Thus, when the transmitter reaches this maximum, without any acknowledging, the MAC layer presumes a sending failure and the packet transmission is cancelled.

III. Related Work

Initial empirical works on the consumption of sensors in WSNs were devoted to devices which utilize proprietary stacks or just the physical layer of 802.15.4 (see, for example, the study in [5] about the CC1000 radio module of Mica2 motes by Crossbow [6]). However, many recent theoretical, simulation-based and, in less proportion, laboratory studies have focused on modeling and characterizing the performance of 802.15.4/ZigBee WSNs.

The experimental testbeds presented in [9] [10] analyzed the coexistence of 802.15.4 with other wireless technologies (802.11 and/or Bluetooth) operating in the same 2.4 GHz ISM band. Results suggested that 802.15.4 throughput may be seriously affected by such interferences. In [11] authors compared non-beacon and beacon transmission modes in a realistic scenario with two IEEE 802.15.4 development boards through different performance metrics. The study in [12] briefly summarizes the current consumption of commercial chipsets of diverse standards for wireless communications, including Bluetooth, Ultrawideband (UWB), 802.11 (Wi-Fi) and 802.15.4/ZigBee technologies, during packet transmission and reception.

The performance of CSMA/CA algorithm in 802.15.4 networks, has been analytically modeled in many articles such as [13][14][15][16][17] for both beacon-enabled and/or beaconless topologies. The correctness of these models is assessed by simulations. On the other hand, the datasheets of 802.15.4 radio modules normally describe the current consumption of the motes for the different basic states of the transceiver (idle, sleep, transmitting or receiving modes). Thus most battery models in the literature are merely based on the data offered by the vendors, without providing any validation with actual motes.

In [18], the proposed model for slotted (beaconed) 802.15.4 MAC is employed to predict the energy consumption per received data bit. However, the utilized consumption model for the different states of the nodes is not justified. A similar study, also focused on beacon enabled cluster-trees, is presented in [19]. The study offers a mathematical formulation to compute the consumption of the ZigBee Coordinator and the end devices of the cluster-tree depending on the emitted traffic and the beacon timing. For the calculus of the power consumption the model (which assumes that the radio state is idle during the CSMA/CA backoff time) utilizes the values offered by the datasheets of Chipcon (now acquired by Texas Instruments) CC2420 radio transceiver and the Microchip PIC18LF8720 low-power microcontroller.

The consumption in beaconed networks is also characterized in [20]. In that interesting paper authors present their own measurements of the power consumption of a CC2420 transceiver (although the experimental testbed for the measurements is not described). The paper also empirically characterizes the relationship between the received power and the bit error probability. As a result, the proposed model, which takes into account the dynamics of CSMA/CA mechanism, permits to calculate the mean required energy per data bit as a function of the path losses.

The consumption of IRIS sensors, which employ an ATMEL AT86RF230 transceiver, is studied in [21]. The performed tests allow characterizing the current drained during the basic operations of the motes (association, binding and data transmission). The deployed testbed does not isolate the motes and does not either consider the effect of the activity in the radio medium on the consumption. The study in [22] develops a simple linear model to estimate the upper lifetime bound of a WSN. The model is based on measurements of the energy consumption and execution time of different operations on a Tmote Sky sensor mote (which is provided with a CC2420 radio module). As the study is intended to predict the longest possible sensor lifetime, neither the measurements nor the model contemplate the extra energy due to failed attempts to access the channel or lost messages provoked by collisions.

The work in [23] assesses the applicability of beaconed 802.15.4/ZigBee to industrial plant control applications. The evaluation is carried out through OMNeT++ simulations. The mean energy consumption per transmitted byte is estimated by assuming the battery model of a CC1000 radio module, which is not compliant with 802.15.4 standard.

The feasibility of using 802.15.4 specifications for medical Personal Area Networks is analyzed in works such as [24][25][26]. In particular [24] presents an analytical model to compute the lifetime of a hypothetic network of implanted 802.15.4 sensors. The study, which utilizes the typical consumption of a CC2420 chip, is carried out for both beacon and beaconless modes concluding that beaconed networks present more restrictions in term of available data rate and crystal tolerance. The applicability of 802.15.4 communications in medical WSNs is investigated in [25] and [26] through systematic simulations with OPNET and OMNeT++ tools. Aiming at calculating the energy consumed per message, authors in [26] utilize the model documented in the datasheets of Jennic JN5139 ZigBee modules.

The study in [27] analyses the reliability of 802.15.4 cluster-trees when three different sets of values are employed to define the parameterization of CSMA/CA algorithm. The same authors propose in [28] a cross-layer technique to tune the 802.15.4 MAC parameters. According to this technique, which is evaluated through ns-2 simulations of both single-hop and multihop WSN, the MAC parameters are adaptively defined to minimize battery usage. By means of simulations with Castalia 3.0 simulator, the article in [29] evaluates the energy consumption of sensor nodes in a multi-hop beacon-enabled WSN when using physical and logical channel quality estimators. All these three studies also employ the battery consumption model of a Texas Instruments CC2420 radio transceiver. In [30], CSMA/CA parameterization is studied in a real testbed with Jennic JN5139 modules. Authors measure a message loss rate in the range [0-5%] although the conditions in which these losses are induced are not under control. Furthermore the goal of the measurements is to evaluate the delivery ratio of the 802.15.4 motes, so power consumption is not considered.

IV. EMPLOYED TESTBED

The employed testbed network consisted of a simple 802.15.4 star topology. The star comprises a Coordinator node (acting as the network sink) and a (leaf) end-device, which performs as the sensor mote. The network was put into operation with two MSP4302618 Experimenter Boards [31] by Texas Instrument (TI), one of the most widespread vendors of 802.15.4/ZigBee technology. These boards incorporate a last generation MSP430 microcontroller and can be extended with different TI low-power RF wireless modules. For our experiments, the boards were connected to an 802.15.4-compatible CC2520EMK [32] transceiver working in the 2.4 GHz ISM band. The nodes were powered by two AAA 1.5V batteries.

In contrast with previous and other existing 802.15.4 transceivers, CC2520EMK enters into a sleep mode during most part of the random CSMA wait periods. Consequently, the consumption of these idle CSMA waiting times (which we measured in our previous work [4]) has been practically removed. This fact is coherent with the analytical models of battery consumption existing in the literature related to 802.15.4 technology, which almost unanimously assume that transceiver is turned off during the CSMA waits. Anyhow, CCA failures increase the number of CCA operations, so it may still impact on the current consumption.

In our experiments, we measured the current drained from the batteries by the whole board. To isolate the effects of radio communications on battery utilization, all the peripherals included in the board (e.g. LED diodes) were carefully turned off for the measurements. Similarly, non utilized GPIO (General Purpose Input Output) pins were set as outputs to minimize the consumption of not-connected inputs.

As the majority of commercial 802.15.4 radio transceivers, the CC2520EMK module works in the 2.4 GHz band. This chip implements the physical layer of IEEE 802.15.4 as well as some functionalities (such as CCA operation, frame filtering or automatic ACK generation) corresponding to the IEEE 802.15.4 MAC layer. The CC24XX & CC25XX families of IEEE 802.15.4 compliant transceivers are conceived to be utilized together with Z-Stack, the version of the 802.15.4/ZigBee stack designed by TI. In the boards of our testbed, Z-stack is loaded and run by the MSP430 microcontroller.

To emulate CCA failures and packet losses, the C source code of Z-stack was intentionally modified and recompiled before being installed in the microcontroller of the sensor mote. In particular we altered the procedure that executes the CCA for unslotted CSMA transmissions in the transceiver as well as the function that informs the sending node about the reception of ACK messages. So, the CSMA wait is performed (or not) depending on a constant probability and not on the actual state of the medium (which will be always free because wired transmissions are employed). Thus, this parameter, which is defined by the user for every experiment, decides the probability of assuming that the channel is busy and, consequently, the existence of a CCA failure. Similarly, packet retransmission is uniquely based on another user-defined probability, which determines the possibility of not detecting the reception of the ACK packets (which is basically equivalent to a packet collision). In the code, for every CCA operation and every packet transmission, a pseudo-random integer between 0 and (2^{16}-1) is generated. This number is normalized to 1 and compared with the existing CCA failure (or packet collision) probability to decide if a failure (or a collision) must take place.

The utilized testbed for the consumption characterization is depicted in Figure 1. The goal is to measure the current required by a generic sensing node (performing as an 802.15.4 end device) when it regularly sends data to a sink node (with the role of the Coordinator). This upstream traffic closely approximates the typical application of a Wireless Sensor Network.

As radio modules incorporate an SMA antenna connector, the communication between the motes is achieved through a 0.5 m long SMA-to-SMA cable. Thus, the interferences of any other device operating in the same unlicensed band (e.g. through Bluetooth or Wi-Fi connections) are avoided. The transmission power of the transceiver is chosen to be 0 dBm (1 mW) while the attenuation provoked by the cable and each SMA connector is under 0.1 dB/m and 1 dB respectively. Consequently, the power at the receptor is about -2 dBm, which is far from the limits imposed by the saturation of the radio receptor (6 dBm) and by the transceiver sensitivity (-98 dBm). This guarantees that any detected CCA failure or packet collision is caused by the failure and collision probabilities introduced in the ZigBee stack of the end device.

To estimate the mean current required by the sensor mote for the different considered scenarios, we utilized a true-rms Fluke 289 digital multimeter. For the range of 50 mA, this piece of equipment measures the DC current with an accuracy of 0.05% and a resolution of 1 µA. The multimeter is connected between the voltage source (of 3 V) and the supply pin of the experimental board (as it is reflected in Fig.1). In this board, the consumption of the transceiver or the microcontroller cannot be easily segregated from that of the rest of the board. Therefore the measurements compute the current drained by the whole board. After minimizing the effects of the peripherals, this consumption is essentially caused by the aggregated activity of the microcontroller and the radio transceivers.

The applications loaded in the motes were part of a control application provided with the demonstration kit. In the application, the end device, acting as a switch (e.g. a lamp switch), may send a simple command to the coordinator (which could be located in a bulb). In our experiments this command are programmed to be transmitted at regular intervals with a programmable periodicity. Each command is conveyed in a single 802.15.4/ZigBee packet with a MAC data payload of 25 bytes (9 bytes of application data plus the 16 byte overhead introduced by the ZigBee Network Layer and the ZigBee Application Support Sub-Layer). This scenario can represent the typical case of a ZigBee WSN where sensors periodically transmit a simple parameter which can be codified in some bytes, within the payload of a small packet.

V. OBTAINED RESULTS

We executed a series of systematic experiments using the previous testbed and modifying the probabilities of a CCA failure and packet collision. In all the experiments, the algorithms involved in the CSMA/CA access method were parameterized with default values defined by the 802.15.4 specification. (e.g. the minimum and maximum value for the Backoff exponent).

Packet rate was fixed to 5 data packets per second. In addition, the end device (in a typical ZigBee application) is programmed to poll the Coordinator in a periodic basis to request possible data. In our experiments, this poll process was programmed to be executed just one time every 5 seconds. Similarly, after sending any packet, end devices normally transmit a poll packet to the Coordinator to enable a response to the sent data. In our case, the time between the data packet and its corresponding poll packet was also set to a maximum of 5 seconds (thus, only one poll frame of this type is transmitted for every 25 packets). As a consequence, poll packets have a minor impact on battery consumption, which is mainly due to the data packets.

Different transmission scenarios were considered by varying the probabilities of experiencing a CCA failure or a packet collision.

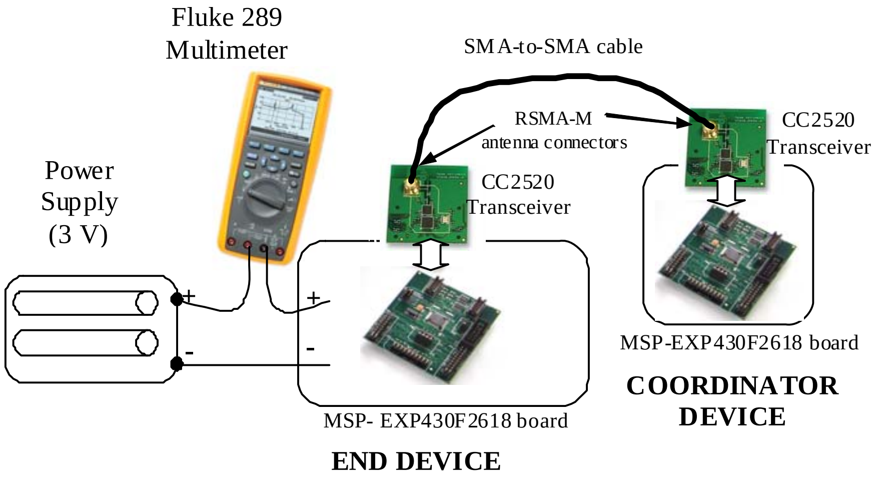


Figure 1. Experimental testbed

Authors in [16] analytically compute both the CCA failure probability and the packet collision in a beaconless 802.15.4 network. Their analysis assume that motes only employ carrier sensing techniques so that only the activity of other 802.15.4 nodes in the network can cause CCA failures. This implies that external interferences (by devices of other technologies working in the same band) and other channel errors are not considered to compute the CCA failure or packet collision probabilities. Even neglecting the effects of the interferences, author show that for data rates of 5 packets per second and per node, networks of 10 to 100 nodes may suffer probabilities of CCA failure and packet collisions in the range [0.1-0.9]. These probabilities clearly drop only if the nodes present a lower activity. Therefore, even in networks with a not very high number of nodes, a high rate of CCA failures and packet collisions must be expected if the motes update and transmit their sensed data frequently. Interferences just can deteriorate this behavior.

Taking into account this realistic data, we performed different experiments by modifying the probabilities of CCA failure and packet collisions from 0.0 (ideal case where the channel is always available and no loss occurs) to 1.0 (worst case where channel is always busy and all packets are lost), with constant increments of 0.1.

The measured mean current drained by the end device is depicted in Figure 2. We repeated the experiments for three limit cases: a) CCA operations may fail but no collision takes place; b) only packet collisions can occur (CCA always successes); c) CCA operation and collisions happen with the same probability. The two first cases allow isolating the effect of each process whereas the third case corresponds to the most realistic situation where collisions and CCA fails are strongly correlated.

Each displayed point in the figure represents the measurement of the mean drain current after the transmission of 9000 packets (about 30 minutes) under a constant probability of CCA failure and/or packet collision.

With collision probabilities higher than 0.8, the losses cause the sensor to disassociate from the coordinator very often. This resulted in an extremely high consumption (25.662 mA) as the end device is trying to re-associate to the Coordinator almost permanently.

The graphs show that CCA failures increase the power consumption. In particular, the extra consumption due to the repetition of the CCA operation ranges from 10% to 50%. The rise rate in the battery usage is accelerated for higher values of the probability of a CCA failure. This can be explained by the fact that the utilized version of ZigBee stack tries to retransmit the data packet once again whenever a channel access failure occurs (after 5 consecutive CCA failures). Thus, the increase does not follow a linear function.

On the other hand, packet collision is shown to have a higher impact on the battery usage. As long as the packet has to be retransmitted for every loss, the consumption rapidly grows with the collision probability. Figure also illustrates that the combination of both effects (collisions and CCA failures) strongly degrades the lifetime of the battery. So, for this more realistic situation, the drain current rockets for collision (and CCA failure) probabilities higher than 0.4. This fact should be carefully taken into account when designing a WSN where sensors are expected to have a short duty-cycle. In that scenario, collisions and channel access failures could easily reduce the battery lifetime (in a network with some tens of nodes) by a factor of 3 or 4 with respect to the ideal case with a free radio medium. Moreover, the employed uncorrelated model which decides the timing of CCA failures or packet losses can also be considered too optimistic. In fact, the periods of channel occupation or the activity of interferences normally follow correlated patterns which are better characterized by Markov processes. The existence of long periods of channel occupancy or other radio channel problems should even decrease the node lifetime in a real WSN application.

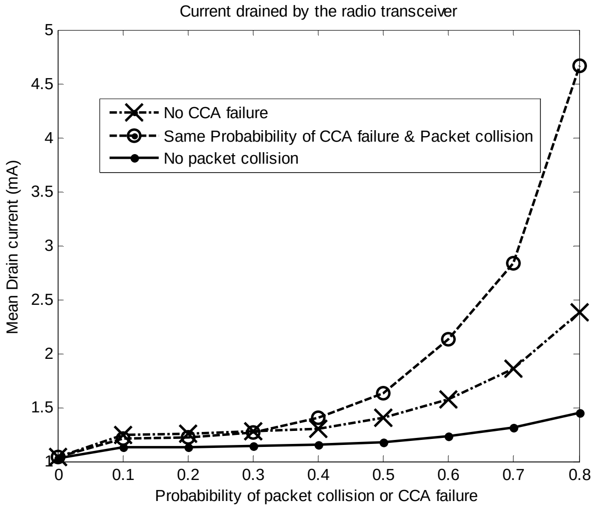


Figure 2. Average measured drain current as a function of the probability of CCA failure and/or packet collision.

VI. CONCLUSIONS

This paper has empirically studied the impact of CCA failures and packet collisions on the current consumption of an actual 802.15.4/ZigBee mote.

While other practical studies in the literature introduce CCA failures or packet losses in the 802.15.4 communications by adding wireless interfering sources, our study has implemented a simple testbed of two actual nodes where channel occupation and packet collisions are emulated via software by altering the protocol stack of the motes. Thus, the utilized testbed has permitted to carry out a set of systematic and repeatable measurements of the battery consumption for diverse preset values of the channel occupation and packet collision probabilities (which is not possible in a scenario where radio communication problems are induced by a background wireless traffic.)

Achieved results indicate that the combination of CCA failures and packet collisions may produce a severe drop in the battery lifetime of the nodes.

The paper has presented the preliminary results of an ongoing investigation. Future work should investigate the effects of other factors, considering a more complex stochastic process to simulate and correlate radio channel access failures and packet collisions. The accuracy of theoretical consumption models in the literature should be equally contrasted against the obtained measurements.

Acknowledgment

This work was partially supported with European and Spanish public funds by Project No. TEC2009-13763-C02-01.

References

[1] 802.15.4-2003 IEEE Standard for Information Technology-Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), 2003.

[2] ZigBee-Alliance. Available online: http://www.ZigBee.org (access on 28 January 2010).

[3] ON World report, July 2011. URL: http://www.onworld.com

[4] J.M. Cano-García, and E. Casilari, "An Empirical Evaluation of the Consumption of 802.15.4/ZigBee Sensor Motes in Noisy Environments", *Proc. of IEEE International Conference on Networking, Sensing and Control (ICNSC 2011)*, Delft (The Netherlands), April 2011.

[5] O. Landsiedel, K. Wehrle, and S. Gotz, "Accurate prediction of power consumption in sensor networks", *Proc. of Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*, Sydney, Australia, pp.37-44m May 2005.

[6] Crossbow Technology. Available online: http://www.xbow.com (access on 28 January 2010).

[7] J. Zheng, and M.J. Lee, "A Comprehensive Performance Study of IEEE 802.15.4, Sensor Network operations"; IEEE Press: Wiley Interscience: New Jersey, NJ, USA, pp. 218-237, 2006.

[8] W.T.H. Woon, T. C. Wan, "Performance evaluation of IEEE 802.15.4 wireless multi-hop networks: simulation and testbed approach", *International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)*, Vol. 3, Issue 1, pp. 57-66, 2008.

[9] K. Shuaib, M. Alnuaimi M., M. Boulmalf, I. Jawhar, F. Sallabi and A. Lakas, "Performance evaluation of IEEE 802.15.4: experimental and simulation results",. *Journal of Communications*, Vol 2, No 4, pp. 29-37, June 2007.

[10] M. Petrova, J. Riihijarvi, P. Mahonen, and S. Labella "Performance study of IEEE 802.15.4 using measurements and simulations", *Proc. of IEEE Wireless Communications & Networking Conference (WCNC)*, Las Vegas, NV, USA, April 2006.

[11] J.S. Lee, "Performance evaluation of IEEE 802.15.4 for low-rate wireless personal area networks", *IEEE Trans. on Consumer Electronics*, Vol. 52, Issue 3, pp. 742-749, Aug. 2006.

[12] J. S. Lee, Y.W. Su, and C.C. Shen, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi", *Proc. of 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Taipei, Taiwan, pp 46-51, Nov. 2007.

[13] P.K. Sahoo and J.P. Sheu, "Modeling IEEE 802.15.4 based wireless sensor network with packet retry limits", *Proc. of the 5th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks* (PE-WASUN 2008), Vancouver, Canada, pp. 63-70, Oct. 2008.

[14] C. Buratti, and R. Verdone, "A mathematical model for performance analysis of IEEE 802.15.4 non-beacon enabled mode", *Proc. of 14th European Wireless conference (EW 2008)*, Prague, Czech Republic, June 2008.

[15] Z. Chen, C. Lin, H. Wen, and H. Yin, "An analytical model for evaluating IEEE 802.15.4 CSMA/CA protocol in low-rate wireless application", *Proc. of 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW '07)*, Niagara Falls, Canada, pp. 899-904, May 2007.

[16] M. Goyal, D. Rohm, H. Hosseini, K.S. Trivedi, A. Divjak, and Y.A. Bashir, "A stochastic model for beaconless IEEE 802.15.4 MAC operation", In *Proc. of International Symposium on Performance Evaluation of Computer & Telecommunication Systems (SPECTS 2009)*, Istanbul, Turkey, Vol. 41, pp. 199- 207, July 2009.

[17] T.J. Lee, H.R. Lee, and M.Y. Chung, "MAC throughput limit analysis of slotted CSMA/CA in IEEE 802.15.4 WPAN", *IEEE Communications. Letters*, Vol. 10, No. 7, pp. 561-563, July 2006.

[18] S. Pollin, M. Ergen, S.C. Ergen, B. Bougard, L. van der Perre, F. Catthoor, I. Moerman, A. Bahai, and P. Varaiya, "Performance analysis of Slotted Carrier Sense IEEE 802.15. 4 Medium Access Layer", *IEEE Trans. on Wireless Communications*, Vol. 7, Issue 9, pp. 3359-3371, Sept. 2008.

[19] M. Kohvakka, M. Kuorilehto, M. Hännikäinen, and T.D. Hämäläinen, "Performance analysis of IEEE 802.15.4 and ZigBee for large-scale wireless sensor network applications", *Proc. of the 3rd ACM International Workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks (PE-WASUN'2006)*, Torremolinos, Spain, pp. 48-57, Oct. 2006.

[20] B. Bougard, F. Catthoor, D.C. Daly, A. Chandrakasan, and W. Dehaene, "Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: Modeling and Improvement Perspectives", *Proc. of Design, Automation and Test in Europe (DATE'05)*, Munich, Germany, Vol. 1, pp.196-201, March 2005.

[21] P. Moravek, D. Komosny, M. Simek, and L. Mraz, "Energy demands of 802.15.4/ZigBee communication with IRIS sensor motes", *Proc. of 2011 34th International Conference on Telecommunications and Signal Processing (TSP)*, Budapest, Aug. 2011, pp. 69 – 73.

[22] M. Amiri, *Wireless Sensor Networks: Evaluation of Power Consumption and Lifetime Bounds*, Editor LAP Lambert Academic Publishing, 2011

[23] F. Chen, N. Wang, R. German, and F. Dressler, "Simulation study of IEEE 802.15.4 LR-WPAN for industrial applications", *Wireless Communications & Mobile Computing*, Vol. 10, Issue 5, pp. 609-621, May 2010.

[24] N.F. Timmons, and W.G. Scanlon, "Analysis of the performance of IEEE 802.15.4 for medical sensor body area networking", *Proc. of the 1st IEEE international conference on Sensor and ad hoc communications and networks (SECON'04)*, Santa Clara, CA, USA, , pp. 16-24, Oct. 2004.

[25] N. Golmie, D. Cypher, and O. Rebala, "Performance analysis of low rate wireless technologies for medical applications", *Computer Communications.*, Vol. 28, Issue 10, pp. 1266-1275, June 2005.

[26] H. López-Fernández, P. Macedo, J.A. Afonso, J.H. Correia, and R. Simões, "Performance evaluation of a ZigBee-based medical sensor network", *Proc. of International Workshop on Wireless Pervasive Healthcare (WiPH 2009)*, London, UK, March 2009.

[27] G. Anastasi, M. Conti, M. Di Francesco, and V. Neri, "Reliability and Energy Efficiency in Multi-hop IEEE 802.15.4/ZigBee Wireless Sensor Networks", Proc. of the IEEE Symposium on Computers and Communications (ISCC 2010), Riccione, Italy, June 22-25, 2010.

[28] M. Di Francesco, G. Anastasi, M. Conti, S. K. Das, V. Nerim "Reliability and Energy-Efficiency inIEEE 802.15.4/ZigBee Sensor Networks: An Adaptive and Cross-Layer Approach", *IEEE Journal on Selected Areas in Communications*, Vol. 29, No. 8, pp. 1508-1524, Sept. 2011.

[29] M. Tariq, M. Macuha, Y.J.Park, and T. Sato, "Performance evaluation of the IEEE 802.15.4 multi-hop communications in error-prone wireless sensor networks", *Proc. of the 9th ACM International Symposium on Mobility Management and Wireless Access (MOBIWAC'2011)*, Miami (FL, USA), Nov. 2011.

[30] G. Anastasi, M. Conti, and M. Di Francesco, "The MAC unreliability problem in IEEE 802.15.4 wireless sensor networks", Proc. of ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2009), October 2009, Tenerife (Spain), pp. 196-203.

[31] Texas Instruments. MSP430F5438 Experimenter Board. Datasheet Available online at: http://focus.ti.com/docs/toolsw/folders/print/msp-exp430f5438.html (access on 8 October 2010).

[32] Texas Instruments. CC2520 Second generation 2.4 GHz ZigBee/IEEE 802.15.4 RF transceiver. Datasheet Available online at: http://focus.ti.com/lit/ds/symlink/cc2520.pdf (access on 28 January 2010).

Maximizing Transported Data in a Wireless Sensor Network: How Much can the Network Transport Before Partition?

Samta Shukla
Department of Electronic Systems Engineering
Indian Institute of Science
Bangalore, India
samta@cedt.iisc.ernet.in

Joy Kuri
Department of Electronic Systems Engineering
Indian Institute of Science
Bangalore, India
kuri@cedt.iisc.ernet.in

***Abstract*—We consider a scenario where the communication nodes in a sensor network have limited energy, and the objective is to maximize the aggregate bits transported from sources to respective destinations before network partition due to node deaths. This performance metric is novel, and captures the useful information that a network can provide over its lifetime. The optimization problem that results from our approach is nonlinear; however, we show that it can be converted to a Multicommodity Flow (MCF) problem that yields the optimal value of the metric. Subsequently, we compare the performance of a practical routing strategy, based on Node Disjoint Paths (NDPs), with the ideal corresponding to the MCF formulation. Our results indicate that the performance of NDP-based routing is within 7.5% of the optimal.**

***Index Terms*—Network Partition Time (NPT), Maximal Independent Sets(MIS), Node-disjoint Paths (NDPs), Link-Contention graph (LCG)**

I. Introduction

The model we consider in this paper is motivated by a wireless sensor network deployed for soil moisture monitoring in a semi-arid region in the state of Karnataka in South India [1], [2], [3]. The sensors are embedded at chosen spots where moisture is to be measured. Usually, the chosen spots are well-illuminated by sunlight, so that solar panels can be used to supply energy to the sensors and associated electronics. The data collection point or Base Station (BS) is several hundred meters away, and is assumed to be close to an electrical outlet.

The area between the spots being monitored and the BS is covered by a multihop wireless communication network. However, there are constraints on the placement of the communication nodes; for example, there may be patches of ground that are well-illuminated, but inaccessible (for example, a patch of well-illuminated ground may be walled off as it belongs to a third party). This forces the communication nodes to be located at spots that may be illuminated poorly. In our *model*, we capture these realistic aspects by assuming that sources and destinations are equipped with infinite energy, but the communication nodes have access to finite supplies of energy only.

Given such a network, we are interested in *maximizing* the bits transported from source(s) to destination(s) before the communication network runs out of energy. A solution with high throughput, but low network lifetime may not be good for our application scenario; similarly, a solution with long network lifetime, but low throughput may be undesirable: A network may remain idle for most of the time claiming a higher *Network Partition Time* (NPT), but the *useful* data transferred might be substantially less. Therefore, we seek to maximize the *product*, which is the aggregate bits transported before network partition.

Sensor networks have wireless links. Links that interfere cannot be activated simultaneously. We utilize the notions of *maximal independent sets* (MIS) in the *link contention graph* [4] to obtain an *equivalent wired network model*. Then, we study the problem of route selection in this equivalent wired network model so that the objective, *viz.,* aggregate bits transported, is maximized, subject to the nodes' "available energy" values. We show that the problem of maximizing the aggregate bits transported before network partition can be formulated as a Linear Program. Next, we consider a routing strategy based on the *maximum number of Node Disjoint Paths between each source-destination pair*, and compare its performance with that of the ideal above. We find that the NDP-based strategy is able to transport an aggregate number of bits that is within 5%–7% of the optimal, indicating that it is a promising strategy.

The contributions of this paper are as follows.

- The metric that we seek to optimize, *viz.,* aggregate bits transported before network partition, is natural. However, to the best of our knowledge, the metric has not been used in the literature.
- The equivalent wired network model results after all wireless-specific aspects (link technology, MAC duty cycles, transmit powers) are encapsulated into a module. The resulting model makes it easier to analyse the performance of routing strategies, in terms of aggregate bits transported, with limited nodal energies.

ISBN: 978-1-61208-207-3

- We show that the our problem can be recast as a Linear Program (a Multicommodity Flow problem), even though the original formulation results in a nonlinear program.
- We examine the performance achievable by a practical routing protocol based on the principle of Node Disjoint Paths (NDPs). Our results indicate that this protocol performs well, achieving an aggregate bit transfer within 7.5% of the ideal value given by the Multicommodity Flow problem.

In Section II, we survey the related work. The models we study in this paper are introduced and analyzed in Section III. A practical routing protocol is discussed in Section IV. Section V compares the performance of the routing protocol, in terms of aggregate bits transported, with the ideal obtained from the models in Section III. We conclude in Section VI.

II. Related Work

Our application context requires sources to send data to the destination at regular intervals. In this context, the "pull" model of operation, where the network sends data only in response to queries from the outside, does not seem appropriate; this is because queries would have to be sent repeatedly, leading to avoidable energy expenditure. So, a lot of the query-based protocols in the literature are difficult to utilize. Further, our objective is to maximize the (weighted) aggregate bits transferred before network partition, and we are unaware of any work that considers this metric while designing protocols. Consequently, the protocols we survey below do not really address our concerns. Nevertheless, for completeness, we discuss several prominent routing protocols in the rest of this section.

In [5], Heinzelman et al. proposed a variant of flooding, SPIN, which disseminates data from one node to every other node, so that, when needed, required data can be extracted from any of the nodes. In our context, replication of data at various nodes is hardly required. Moreover, SPIN does not guarantee reliable data delivery [6].

Both Directed Diffusion (DD) [7] and its variant Gradient Based Routing (GBR) [8] are event-driven protocols, that supply data in response to external queries. Both of them use in-network aggregation. In [9], Yao et al. discussed COUGAR, which is a query-driven, in-network aggregation based protocol in which a group of nodes choose a leader node who aggregates their data and sends to Base-Station. ACQUIRE [10] is similar. However, as mentioned before, event-driven protocols are not natural in our application scenario.

Similar to DD, in [11], Ganesan et al. presented a Braided Multipath Routing Protocol, which has 50% more fault resilience as compared to the Node-Disjoint Multipath protocols, but it uses only one path for data transfer; hence the overall bits transferred is limited by the capacity of a single path. Shah and Rabaey in [12] suggested the alternative-route routing paradigm. Data is routed via primary path, in an energy efficient way; if a fault occurs, alternative-route is chosen for data transfer.

A sensor network multipath version of AODV, AOMDV [13], routes along multiple link disjoint paths, where route-discovery is done through flooding, which is not energy-efficient [14]. In some protocols, for the sake of improving reliability, multiple instances of a single packet are routed through multiple paths. Examples are ReInForm [15], MCMP [16], ECMP [17]. ReInForm provides reliability at the cost of high-energy dissipation. MCMP tries to optimise for reliability and delay, but is highly interference-prone, which limits successful data transmissions. ECMP, in addition to MCMP, optimises for energy-efficiency too, but the packet delivery ratio for ECMP equals that of MCMP. Various protocols which use packet splitting as a packet salvaging technique like N to 1 Multipath [18], or for Forward Error Check like, EQSR [19], H-SPREAD [20] suffer heavily due to interference.

Moving to protocols following the node-disjoint routing paradigm: NDMR [21] chooses three least hop, node disjoint paths (even if many paths exist) for routing, and switches to other paths if any of these three paths fails. REER [22] routes data along two paths; primary path and alternate path, which are selected on the basis of link-cost. The interference-free routing protocols like, I2MR [23], EECA [24], limit the number of paths assigned for routing data to three and two, respectively. In [25], Radi et al. came up with a throughput aware multipath node-disjoint protocol; however, it is event-driven.

In summary, none of the protocols mentioned here raises the question of "maximising aggregate bits transported before network partition." In many cases, the scenario addressed is such that this question does not arise naturally (*e.g.*, application scenarios where minimizing delay is the main objective).

III. Network Model and Analysis

In this paper, we assume the following.

- Each source and destination has access to infinite energy, but the intermediate communication nodes are powered by finite energy sources only.
- Each node has a single radio interface, and a node cannot transmit and receive at the same time.
- Data flows according to a *fluid model*.
- Sources have infinite backlogs.
- Propagation delay is negligible.
- Nodes are static.

We represent the network as a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ with multiple sources given by $S_1, S_2, \ldots, S_K$, multiple destinations by $D_1, D_2, \ldots, D_K$ where $S_i, D_i \in \mathcal{N}, 1 \leq i \leq K$. Let $N = |\mathcal{N}|$, $L = |\mathcal{L}|$. Let $w_1, w_2, ..., w_K$ be the weights associated with $S_1, S_2, ..., S_K$, where, $\sum_{i=1}^{K} w_i = 1$. The weight assigned to a source reflects its priority. Let E_i denote the energy available with node i (in Joules), γ_t the energy consumed for transmitting a data bit (Joules/bit), γ_r be the energy consumed for receiving a data bit (in Joules/bit), irrespective of the node chosen.

Network partition time, denoted by $\widehat{T}$, is defined as the instant when the *first* source-destination pair disconnects (it is possible that several sources are disconnected from their respective destinations at this time). We consider the network to be *partitioned* because there is at least one source that is unable to reach its destination.

In a wireless network, links that interfere with one another cannot be activated simultaneously. To address link interference issues, the notions of *Link Contention Graph* and *Maximal Independent Sets* have been introduced [4]. In the LCG, each vertex represents a *link*, and there is an edge between two vertices if the corresponding links interfere. A maximal subset of non-adjacent nodes in the LCG defines a MIS. Evidently, the links in a MIS can be activated simultaneously because they do not interfere.

A *schedule* of wireless link activation can be viewed as a sequence of MIS-es that are activated in sequence. Typically, a periodic sequence of MIS-es is used.

A. Scheduling of wireless links

Let the bit rate supported by link l be denoted by $C^{(l)}$, $1 \leq l \leq L$. Let $M_1, M_2, \ldots, M_W$ denote the various MIS-es for the given graph, and let a periodic schedule be given. Let a_i denote the fraction (of the period of the schedule) for which the i^{th} MIS is *on*. A link $l \in \mathcal{L}$ can be present in any of the MIS-es (not necessarily disjoint). The fraction of time for which link l is active is obtained by

$$\nu_l = \sum_{i | l \in M_i} a_i$$

The equivalent link capacity for link l is given by,

$$C_l = \nu_l \times C^{(l)}$$

As the equivalent link capacity C_l is obtained by taking into account the fraction of time for which link is active, we can substitute this link by an equivalent *wired* link of capacity C_l that is active all the time. Evidently, the bits transported by the link over the schedule period remains the same, no matter which view is adopted. This observation forms the basis underlying our approach of constructing an *equivalent wired network*.

Let $\mathbf{C}$ be the capacity vector of size L, the l_{th} entry of which represents the effective capacity of the link l for the specified schedule. We represent an independent set M_j by a column vector $\mathbf{r}_j$; the element $r_l = C^{(l)}$ if $l \in M$, and 0 otherwise. Thus, we can write the vector $\mathbf{C}$ as,

$$\mathbf{C} = \sum_{j=1}^{j=W} a_j \mathbf{r}_j$$

B. Multi-commodity Flow Problem

The Linear programming Problem Formulation: The network topology can be represented by a $N \times L$ node-link incidence matrix $\mathbf{A}$, with the entries A_{nl}, $n \in \mathcal{N}$, $l \in \mathcal{L}$ being "1" if l originates at n, "-1" if l terminates at n and "0" otherwise. Set of end-to-end multi-hop flows constitutes the network traffic where each flow has a source and destination. Flow traffic is routed by the network along one or multiple paths from source to destination. For each flow, we denote by x^i the amount of flow (measured in bit/s) injected into the network per unit of time by the i^{th} $S-D$ pair, x denotes the amount of flow injected into the network due to all $S-D$ pairs. y_l is the amount of flow traversing link l per unit of time. Let y_l^i denote the flow through link l lying in the $S_i - D_i$ path. Hence, $x = (x^i)$, $\forall i \in [1, 2..., K]$ and $y_l = (y_l^i)_{l \in \mathcal{L}}$, $\forall i \in [1, 2..., K]$ are the flow rate vector and flow link rate vector, respectively. Unless otherwise stated, we will assume vectors are always column vectors. For a vector z, z^Tdenotes transpose of z.

- **Flow Constraint**

Assuming lossless transmission, the flow conservation law implies that for $n \in \mathcal{N}$:

$$\begin{bmatrix} \\ A_{nl} \\ \\ \end{bmatrix}_{N \times L} \times \begin{bmatrix} : & & : \\ y_l^1 & .. & y_l^K \\ : & & : \end{bmatrix}_{L \times K} = \begin{bmatrix} x^1 & \\ : & ... \\ -x^1 & \end{bmatrix}_{N \times K}$$

and can be compactly written as

$$\mathbf{A}\mathbf{y}^i = \mathbf{u}^i, \; \forall i \in [1, 2, ...K]$$

where $\mathbf{u}^i = (u_n^i)$, $n \in \mathcal{N}$, and u_n^i denotes the amount of flow injected (removed) to (from) the network at node n, i.e., $u_n^i = x^i$ if $n = S_i$, $u_n^i = -x^i$ if $n = D_i$ and $u_n^i = 0$ otherwise, $\forall i \in [1, 2, ...K]$.

- **Energy Expenditure Constraint**

The amount of data transmitted and received by any node is limited by the maximum allowable energy available *i.e.*, the initial nodal energy. Let $\widehat{\mathbf{A}}$ denote the node-link incidence matrix A_{nl} where all the "-1"s are replaced by γ_r and all the "$+1$"s are replaced by γ_t. Recalling the definition of the NPT $\widehat{T}$, we have:

$$\begin{bmatrix} \\ \widehat{A}_{nl} \\ \\ \end{bmatrix} \times \begin{bmatrix} : & : & : \\ y_l^1 & .. & y_l^K \\ : & : & : \end{bmatrix} \times \begin{bmatrix} 1 \\ : \\ 1 \end{bmatrix} \times \widehat{T} \leq \begin{bmatrix} E_1 \\ : \\ E_N \end{bmatrix}$$

Or

$$\widehat{\mathbf{A}} \times \mathbf{y}^i \times \mathbf{1} \times \widehat{T} \leq \mathbf{E}, \; \forall i \in [1, 2, ...K]$$

where $\mathbf{E}$ is $N \times 1$ vector representing the energy available with each node in network.

- **Capacity Constraints**

The capacity of a link limits the aggregate bit rate that it can carry. We have:

$$\begin{bmatrix} : & : & : \\ y_l^1 & .. & y_l^K \\ : & : & : \end{bmatrix} \times \begin{bmatrix} 1 \\ : \\ 1 \end{bmatrix}_{K \times 1} \leq \begin{bmatrix} C_1 \\ : \\ C_L \end{bmatrix}_{L \times 1}$$

Compactly,

$$\sum_{i=1}^{K} y_l^i \leq C_l, \; \forall l \in \mathcal{L}$$

- **Problem Formulation**

Our aim is to maximize the (weighted) aggregate bits carried till NPT. Let w_i, $1 \leq i \leq K$, represent the weights associated with the K source-destination pairs; the weights reflect the "importance" of each source-destination pair. Then the problem can be stated as:

Problem P*:

$$max \sum_{i=1}^{K} w_i x^i \widehat{T}$$

$$s.\ t.:\ \mathbf{A}\mathbf{y}^i = \mathbf{u}^i,\ \forall i \in [1, 2, ...K]$$

$$\widehat{\mathbf{A}} \times \mathbf{y}^i \times \mathbf{1} \times \widehat{T} \leq \mathbf{E},\ \forall i \in [1, 2, ...K]$$

$$\sum_{i=1}^{K} y_l^i \leq C_l,\ \forall l \in \mathcal{L}$$

$$y_l^i \geq 0,\ \forall l \in \mathcal{L}$$

$$x^i \geq 0,\ \forall i \in [1, 2, ...K]$$

$$\widehat{T} \geq 0$$

The formulation above is *not* a linear program owing to the presence of the product of unknowns x^i, $\widehat{T}$ in the objective function. However, we can convert this problem to a Linear Program in terms of the new variables $(x^{(i)}\widehat{T})$, $1 \leq i \leq K$, $(y_l^i\widehat{T})$, $1 \leq i \leq K$, $1 \leq l \leq L$, and $\widehat{T}$, with suitable modifications of two of the constraints:

Problem MCF:

$$max \sum_{i=1}^{K} w_i (x^i \widehat{T})$$

$$s.t.:\ \mathbf{A}(\mathbf{y}^i\widehat{T}) = (\mathbf{u}^i\widehat{T}),\ \forall i \in [1, 2, ...K]$$

$$\widehat{\mathbf{A}}(\mathbf{y}^i\widehat{T}) \times \mathbf{1} \leq \mathbf{E},\ \forall i \in [1, 2, ...K]$$

$$(\sum_{i=1}^{K} y_l^i\widehat{T}) \leq (C_l\widehat{T}), \forall l \in \mathcal{L}$$

$$y_l^i\widehat{T} \geq 0,\ \forall l \in \mathcal{L}$$

$$(x^i\widehat{T}) \geq 0, \forall i \in [1, 2, ...K]$$

$$\widehat{T} \geq 0$$

Lemma 1. *The optimal objective function values in **Problem P*** and **Problem MCF** are equal.*

Proof: Let $\widehat{T}^*$, $x^{*,i}$, $1 \leq i \leq K$, $y_l^{*,i}$, $1 \leq i \leq K$, $1 \leq l \leq L$, represent the optimal solution to **Problem P***. We generate the following point which is feasible for **Problem MCF**: $\widehat{\tilde{T}} = \widehat{T}^*$, $(\tilde{x}^i\widehat{\tilde{T}}) = x^{*,i}\widehat{T}^*$, $1 \leq i \leq K$, and $(\tilde{y}_l^i\widehat{\tilde{T}}) = y_l^{*,i}\widehat{T}^*$, $1 \leq i \leq K$, $1 \leq l \leq L$. Clearly, the point $\left(\widehat{\tilde{T}}, (\tilde{x}^i\widehat{\tilde{T}}), (\tilde{y}_l^i\widehat{\tilde{T}})\right)$ is feasible for **Problem MCF** as the point $\left(\widehat{T}^*, x^{*,i}, y_l^{*,i}\right)$ is feasible for **Problem P***, and the objective function for **Problem MCF** at $\left(\widehat{\tilde{T}}, (\tilde{x}^i\widehat{\tilde{T}}), (\tilde{y}_l^i\widehat{\tilde{T}})\right)$ is equal to the *optimal* objective function for **Problem P***. Therefore,

$$\text{Optimal objective in } \textbf{Problem MCF} \geq$$
$$\text{Optimal objective in } \textbf{Problem P}*$$

Similarly, starting from the optimal solution $\left(\widehat{\tilde{T}}^*, (\tilde{x}^i\widehat{\tilde{T}})^*, (\tilde{y}_l^i\widehat{\tilde{T}})^*\right)$ to the **Problem MCF**, we can generate a feasible point for the **Problem P***, and the objective function value of **Problem P*** at *this* point will equal the *optimal* objective function value of **Problem MCF**. This implies that

$$\text{Optimal objective in } \textbf{Problem P}* \geq$$
$$\text{Optimal objective in } \textbf{Problem MCF}$$

From these two inequalities, the claim follows. □

Example 1

In Fig. 1, nodes $N1$, $N2$, $N3$ have energy 60, 80, 80 J, respectively. Energy consumed per bit of transmitted and received data are denoted by $\gamma_t = 10^{-4}\ J/bit$ and $\gamma_r = 10^{-6}\ J/bit$, respectively. The raw capacity of each wireless link is taken to be $C = 8 \times 10^4\ bits/sec = 80\ Kbps$.

The wireless link interference model is as follows. Two links L_1 and L_2 interfere with each other if either node of L_1 is within two hops of one of the nodes of L_2. With this model, the MIS-s are: {(1,7),(3,8),(1,8),(3,7),(2),(5),(4),(6)}. The wired equivalent link capacities [4] are:
$C_1 = C_3 = C_7 = C_8 = (2/8) \times C = 20Kbps$
$C_2 = C_4 = C_5 = C_6 = (1/8) \times C = 10Kbps$

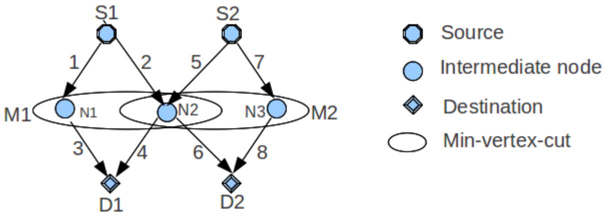


Fig. 1. An example showing 2-Ss'transmitting to their respective Ds'. The paths $S_1 - N_1 - D_1$, $S_1 - N_2 - D_1$ for $S_1 - D_1$ and $S_2 - N_2 - D_2$, $S_2 - N_3 - D_2$ are the Node-disjoint-paths for $S_2 - D_2$, respectively. The links are marked from 1 to 8.

The solution of **Problem MCF** yields a weighted aggregate bits transported till NPT of $\mathcal{F}_{MCF}^1 = 16.71C\ bits$ for $w_1 = 0.25$, $w_2 = 0.75$, and $\mathcal{F}_{MCF}^2 = 13.62C\ bits$ for $w_1 = 0.5$, $w_2 = 0.5$.

It is interesting to note that the solution obtained by solving **Problem MCF** exactly determines the solution for **Problem P***. Consider the equal weight case for **Problem P***, the optimal solution for the link flow vector is $\begin{bmatrix} 14289.52 & 9870.35 & 14289.52 & 9870.35 \\ 9182.33 & 9182.335 & 19052.68 & 19052.68 \end{bmatrix}$ bits/sec. The data injected into the network by the two sources', flow rate vector, is $\begin{bmatrix}2415.98 & 2823.50\end{bmatrix}$ bits/sec. The disconnection time $\widehat{T}$ is 41.57 sec.

- Clearly, all the link-flows are less than the link-capacity which indicates that for maximizing aggregate bits transported, a greedy strategy is not required.
- **Problem MCF** insists on maintaining the flow rate vector constant over time, leading to an optimal solution where all source-destination pairs disconnect at the same time. So, even though our formulation considered the time till at least one $S-D$ pair is disconnected, the MCF formulation provides a solution where all sources get disconnected at the same time. This is proved in the following Lemma.

Lemma 2. *The optimal solution to* ***Problem MCF*** *yields a Network Partition Time* $\widehat{\tilde{T}}^*$ *at which* ***all*** *source-destination pairs are disconnected.*

Proof: Suppose that the optimal solution to **Problem MCF** has been found, and the NPT $\widehat{\tilde{T}}^*$ is such that at least one source-destination pair is not disconnected. Without loss of generality, we assume that there is just one source-destination pair that is not disconnected; in case there is more than one, the argument below can be repeated.

Consider the source that is not disconnected from its destination. There must be at least one path from this source to its destination such that all nodes on this path have positive residual energy. Therefore, this path can sustain data transfer for a longer duration $\widehat{\tilde{T}_1}$ (say), where $\frac{\widehat{\tilde{T}_1}}{\widehat{\tilde{T}}^*} =: \alpha > 1$. Using $\widehat{\tilde{T}_1}$, we can generate a solution to **Problem MCF** at which the objective function increases strictly. To do this, we consider a solution where all other source rates are decreased to $x^{*,i}/\alpha$, which allows path lifetimes to increase to $\alpha\widehat{\tilde{T}}^* = \widehat{\tilde{T}_1}$. This means that the aggregate bits transported by these sources till time $\widehat{\tilde{T}_1}$ remains the same as before. The source rate for the pair that is not disconnected is maintained at the same level as before. Then, because $\widehat{\tilde{T}_1} > \widehat{\tilde{T}}^*$, we conclude that the aggregate bits transported till $\widehat{\tilde{T}_1}$ is strictly greater than that transported till $\widehat{\tilde{T}}^*$. This is a contradiction, and hence the claim follows. □

IV. Routing

In the previous section, we presented an approach to obtain the maximum (weighted) aggregate bits transported, before all sources lose connectivity to their respective destinations. In this section, we turn to the practical issue of routing. Our objective is to examine the routing strategy based on Node Disjoint Paths (proposed in the literature) in terms of the same criterion, *viz.,* aggregate bits transported before disconnection.

Node-Disjoint-Path Routing

NDP routing is discussed prominently in the literature [18] [19] [20] [23] [24] [25]. However, many papers use limited number of NDPs for routing; we consider routing through the *maximum* possible number of NDPs.

1) Finding the maximal NDP set for routing: Our approach is based on Menger's Theorem[26], which states that "The *min-cut* for a $S-D$ pair equals the *maximum number of NDP*." The Node-Disjoint-Paths for each $S-D$ pair is constructed such that:

1) Each path contains exactly *one* node belonging to the min-vertex cut for the $S-D$ pair.
2) The total number of paths formed for a $S-D$ pair equals the cardinality of the min-vertex cut set for that $S-D$ pair.

Remark: In case of overlapping min-vertex cut-sets for the $S-D$ pairs, the total number of NDP might exceed the total cardinality of min-vertex cut-set for the graph.

2) To calculate the Total Data Transferred: After having identified a NDP set for a source-destination pair, our approach is to consider each of the node disjoint paths, and obtain the data transferred through it. We assume that *sources are greedy*, in the sense that they transmit at the maximum rate possible, subject to the constraints that link capacities are finite and have to be *shared* among all flows that pass through those links.

Focusing on a specific path in the NDP path set, our approach is to find the lifetime of each node on the path. The node lifetime is determined by the node's available energy, and the total data rate through it. The smallest node lifetime is the path lifetime. Finally, we calculate the total number of bits transported from each source that uses the path to the corresponding destination. Iterating over paths, we obtain the weighted aggregate bits transported before disconnection.

Example 2

Referring to Fig 1, let T_{N1}, T_{N2} and T_{N3} denote the lifetimes of nodes $N1$, $N2$ and $N3$, respectively. Using the equivalent wired link capacity values obtained earlier (just above Fig 1), and recalling that $\gamma_t = 10^{-4}$ and $\gamma_r = 10^{-6}$, we have, $T_{N1} = 60/((10^{-4}+10^{-6})\times(2\times10^4)\times1) = 29.71\ sec$, $T_{N2} = 80/((10^{-4}+10^{-6})\times(1\times10^4)\times2) = 39.61\ sec$, $T_{N3} = 80/((10^{-4}+10^{-6})\times(2\times10^4)\times1) = 39.61\ sec$. Let the weights given to $S1$ be $w_1 = 0.25$, to $S2$ be $w_2 = 0.75$. Data transferred across $S_1 - D_1$ is $(29.7\times(C/4) + 39.6\times(C/8)) = 12.38C$. For $S_2 - D_2$ it is $(39.6\times(C/8) + 39.6\times(C/4)) = 14.85C$. Hence the total data transferred is given by $\mathcal{F}^1_{NDP} = (0.25\times12.38 + 0.75\times14.85)\times C = 14.23C$. Now, let weights be $w_1 = 0.5$, $w_2 = 0.5$. Total data transferred is given by $\mathcal{F}^2_{NDP} = (0.5\times(12.38+14.85)\times C) = 13.62C$.

Remark: The data obtained for this example when both the sources have equal weights are same for both the methods, but discrepancy in results exist when the weights are unequal.

V. Results

Following are the results obtained for $\gamma = \gamma_t = \gamma_r = 10^{-2}$ Joules/bit, $C = 10^2$ Bits/sec for given number of nodes (maximum till 40) and edges (maximum till three times number of nodes) where the graphs are formed randomly. The energy allotted to nodes is chosen randomly from values 1 to 100. The weights given to sources are all equal.

After 50 runs in MATLAB, the average percentage deviation in aggregate bits transported based on the NDP approach as compared to the MCF approach turned out to be (approximately) the following:

1) *7.5% for two S-D pairs*

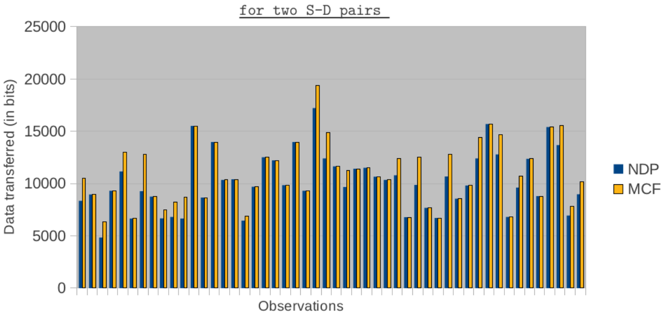


Fig. 2. Performance Comparison of NDP vs MCF for multiple nodes and two S-D pairs.

2) *4.5% for three S-D pairs*

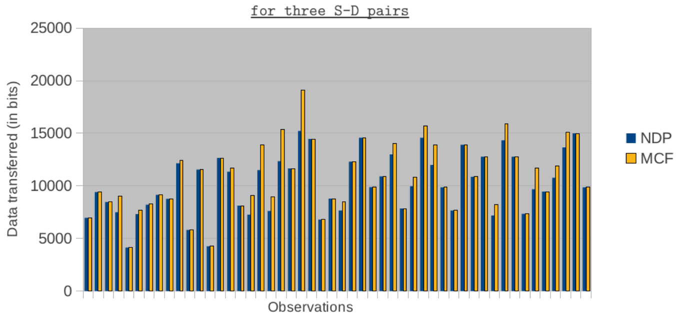


Fig. 3. Performance Comparison of NDP vs MCF for multiple nodes and three S-D pairs

3) *5% for five S-D pairs*

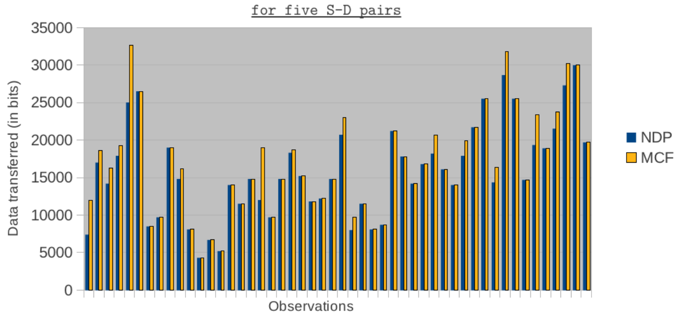


Fig. 4. Performance Comparison of NDP vs MCF for multiple nodes and five S-D pairs.

4) *6% for ten S-D pairs*

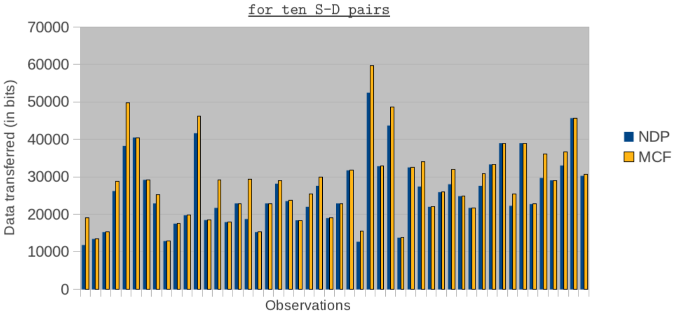


Fig. 5. Performance Comparison of NDP vs MCF for multiple nodes and ten S-D pairs.

It was noted that the deviation in results decreases with larger number of observations. Thus, the performance of the NDP approach is close to the optimal that can be achieved.

VI. Conclusion and Future Work

We considered a scenario where the communication nodes in a sensor network had limited amounts of available energy, and the objective was to maximize the aggregate bits transported from sources to respective destinations before network partition due to node deaths. The metric "aggregate bits transported" results from considering *both* network throughput and network lifetime, and captures the useful information that a sensor network can provide; by itself, neither factor can provide this. We formulated an optimization problem that turned out to be non-linear; however, we showed that it could be converted to a Linear Program, the solution of which yielded the optimal objective value. Next, we compared the performance of a practical routing strategy, based on Node Disjoint Paths (NDPs), with the ideal obtained from the Linear Program, and found that the aggregate bits transported by NDP-based routing strategy was within 7.5% of the optimal, indicating that it is a promising strategy for our metric.

As part of future work, we will develop a framework for evaluating routing and MAC protocols in sensor networks, leveraging the approach given in this paper. The framework will allow an *analytical comparison* between candidate routing and MAC protocol combinations. Each MAC protocol will lead to a specific equivalent wired network model, resulting from MAC characteristics like transmit powers, link speeds (depending on the MAC technology), duty cycle, etc. The equivalent network will allow a study of routing protocols in terms of the aggregate bits transported.

References

[1] J. Panchard, S. Rao, T. V. Prabhakar, J.-P. Hubaux, and H. S. Jamadagni, "Commonsense Net: A Wireless Sensor Network for Resource-poor Agriculture in the semi-arid areas of Developing Countries," *Information Technologies and International Development archive*, vol. 4, no. 1, 2007.

[2] T. V. Prabhakar, H. S. Jamadagni, N. V. C. Rao, and A. Pittet, "Localized Data Gathering Paradigms for Small and Marginal Farm Lands in Semi-Arid regions — Issues and Concerns," in *Proceedings of the UNESCO Conference, Lausanne, Switzerland*, Feb 2010.

[3] T. V. Prabhakar, H. S. Jamadagni, A. Sahu, and R. Venkatesha Prasad, "Lessons from the Sparse Sensor Network Deployment in Rural India," in *Proceedings of the 11th International Conference on Distributed Computing and Networking (ICDCN), Kolkata, India*, Jan 2010.

[4] F. Lo Presti, "Joint Congestion Control: Routing and Media Access Control Optimization via Dual Decomposition for Ad Hoc Wireless Networks," in *Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '05, 2005.

[5] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, ser. MobiCom '99, 1999, pp. 174–185.

[6] 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.

[7] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networking," *Networking, IEEE/ACM Transactions on*, vol. 11, no. 1, pp. 2 – 16, Feb 2003.

[8] C. Schurgers and M. Srivastava, "Energy efficient routing in wireless sensor networks," in *Military Communications Conference. Communications for Network-Centric Operations: Creating the Information Force in MILCOM 2001, IEEE*, vol. 1, 2001, pp. 357 – 361.

[9] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *SIGMOD Rec.*, vol. 31, no. 3, pp. 9–18, 2002.

[10] N. Sadagopan, B. Krishnamachari, and A. Helmy, "The acquire mechanism for efficient querying in sensor networks," in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, May 2003, pp. 149 – 155.

[11] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-Resilient, Energy-efficient Multipath Routing in Wireless Sensor Networks," *SIGMOBILE*, vol. 5, no. 4, pp. 11–25, 2001.

[12] R. Shah and J. Rabaey, "Energy-aware Routing for Low Energy Ad Hoc Sensor Networks," in *Proceedings of IEEE WCNC 2002*, vol. 1, Mar 2002, pp. 350 – 355 vol.1.

[13] P. Hurni and T. Braun, "Energy-efficient multi-path routing in wireless sensor networks," in *Ad-hoc, Mobile and Wireless Networks*, ser. Lecture Notes in Computer Science, D. Coudert, D. Simplot-Ryl, and I. Stojmenovic, Eds. Springer Berlin / Heidelberg, 2008, vol. 5198, pp. 72–85.

[14] M. Radi, B. Dezfouli, K. A. Bakar, and M. Lee, "Multipath routing in wireless sensor networks: Survey and research challenges," *Sensors*, vol. 12, no. 1, pp. 650–685, 2012.

[15] B. Deb, S. Bhatnagar, and B. Nath, "Reinform: reliable information forwarding using multiple paths in sensor networks," in *Local Computer Networks, 2003. LCN '03. Proceedings. 28th Annual IEEE International Conference on*, Oct 2003, pp. 406 – 415.

[16] X. Huang and Y. Fang, "Multiconstrained qos multipath routing in wireless sensor networks," *Wirel. Netw.*, vol. 14, no. 4, pp. 465 – 478, Aug 2008.

[17] A. Bagula and K. Mazandu, "Energy constrained multipath routing in wireless sensor networks," in *Ubiquitous Intelligence and Computing*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, vol. 5061, pp. 453–467.

[18] W. Lou, W. Liu, and Y. Zhang, "Performance optimization using multipath routing in mobile ad hoc and wireless sensor networks," in *Combinatorial Optimization in Communication Networks*. Springer US, 2006, vol. 18, pp. 117 –146.

[19] B. Yahya and J. Ben-Othman, "An energy efficient and qos aware multipath routing protocol for wireless sensor networks," in *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, Oct. 2009, pp. 93 –100.

[20] W. Lou and Y. Kwon, "H-spread: a hybrid multipath scheme for secure and reliable data collection in wireless sensor networks," *Vehicular Technology, IEEE Transactions on*, vol. 55, no. 4, pp. 1320 –1330, Jul 2006.

[21] X. Li and L. Cuthbert, "Stable node-disjoint multipath routing with low overhead in mobile ad hoc networks," in *MASCOTS 2004*, Oct 2004, pp. 184 – 191.

[22] B. Yahya and J. Ben-Othman, "Reer: Robust and energy efficient multipath routing protocol for wireless sensor networks," in *GLOBECOM 2009. IEEE*, Dec 2009, pp. 1 –7.

[23] J.-Y. Teo, Y. Ha, and C.-K. Tham, "Interference-minimized multipath routing with congestion control in wireless sensor network for high-rate streaming," *Mobile Computing, IEEE Transactions on*, vol. 7, no. 9, pp. 1124 –1137, Sept 2008.

[24] Z. Wang, E. Bulut, and B. Szymanski, "Energy efficient collision aware multipath routing for wireless sensor networks," in *Communications, 2009. ICC '09. IEEE International Conference on*, Jun 2009, pp. 1 –5.

[25] M. Radi, B. Dezfouli, S. Razak, and K. Bakar, "Liemro: A low-interference energy-efficient multipath routing protocol for improving qos in event-based wireless sensor networks," in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, Jul 2010, pp. 551 –557.

[26] K. Menger, "Zur allgemeinen kurventheorie," *Fund. Math*, vol. 10, pp. 95–115, 1927.

Simulation Issues in Wireless Sensor Networks: A Survey

Abdelrahman Abuarqoub, Fayez Al-Fayez, Tariq Alsboui, Mohammad Hammoudeh, Andrew Nisbet
School of Computing, Mathematics and Digital Technology
Manchester Metropolitan University
Manchester, UK
f.a.alfayez@gmail.com{A.Abuarqoub, M.Hammoudeh, T.Alsboui, A.Nisbet}@mmu.ac.uk

***Abstract*—This paper presents a survey of simulation tools and systems for wireless sensor networks. Wireless sensor network modelling and simulation methodologies are presented for each system alongside judgments concerning their relative ease of use and accuracy. Finally, we propose a mixed-mode simulation methodology that integrates a simulated environment with real wireless sensor network testbed hardware in order to improve both the accuracy and scalability of results when evaluating different prototype designs and systems.**

Keywords-Wireless Sensor Networks; Simulation tools; Survey; Testbeds; Mix-mode simulation.

I. INTRODUCTION

A successful large-scale Wireless Sensor Network (WSN) deployment necessitates that the design concepts are checked before they are optimised for a specific hardware platform. Developing, testing, and evaluating network protocols and supporting architectures and services for WSNs can be undertaken through test-beds or simulation. Whilst test-beds are extremely valuable, implementing such test-beds is not always viable because it is difficult to adapt a large number of nodes in order to study the different factors of concern. The substantial cost of deploying and maintaining large-scale WSNs and the time needed for setting up the network for experimental goals makes simulation invaluable in developing reliable and portable WSNs applications.

In WSNs, simulation provides a cost effective method of assessing the appropriateness of systems before deployment. It can, for example, help assess the scalability of algorithms free of the constraints of a hardware platform. Furthermore, simulators can be used to simplify the software development process for a particular WSN application. For instance, TOSSIM [1] utilises the component based architecture of TinyOS [2] and provides a hardware resource abstraction layer that enables the simulation of TinyOS applications which can then be ported directly to a hardware platform without further modifications.

Simulation is hence the research tool of choice for the majority of the mobile ad hoc network community. An examination of research papers published in SENSORCOMM 2011 [3] reveals a significant increase in using real testbeds compared to the study published by Kurkowski et al. [4]. Yet, 53% of the authors used simulation in their research. Apart from the self-developed simulators, there are a few widely used network simulators including NS-2 [5] , OPNET [6], MATLAB [7], IFAS [8], and OMNet++ [9]. Figure 1 shows the simulator usage following a survey of simulation based papers in SENSORCOMM 2011 conference. Simulation of ad hoc wireless capabilities for WSNs have been addressed by extending existing simulators, or specifically building new ones, such as NS-3 [10]. The latter class of simulators mostly focus on protocols and algorithms for layers of the network stack, but they do not directly support WSNs.

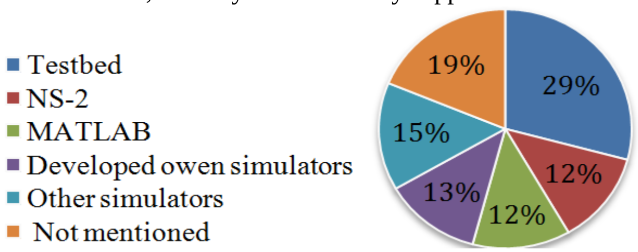


Figure 1. Simulator usage results from a survey of simulation based papers in SENSORCOMM 2011.

Recently, several simulation tools have appeared to specifically address WSNs, varying from extensions of existing tools to application specific simulators. Although these tools have some collective objectives, they obviously differ in design goals, architecture, and applications abstraction level. In the next section, we review some of the important WSNs simulation tools and explore their characteristics.

The rest of the paper is organised as follows: In Section II, the most popular WSNs simulators are outlined and their strengths and weaknesses are discussed. Section III, presents our views about the future of WSNs testing and evaluation methods. Section IV concludes the paper.

II. WSNS NETWORK SIMULATION TOOLS

A. *SensorSim*

SensorSim [11] builds on the NS-2 simulator providing additional capabilities for modelling WSNs. The main features of this platform are: power and communication protocol models; sensing channel and sensor models; scenario generation; and support for hybrid simulations.

The public release of the SensorSim suite of tools was withdrawn due to its unfinished nature and the inability of authors to provide the needed level of support.

Georgia Tech SensorSimII [12] is written in a modular style, where sensor nodes are organised into three components: application, network, and link. The work in SensorSimII may be divided into two areas: the simulator core and the visualisation tools. The simulator core essentially manages an array of independent sensor nodes throughout time. The visualisation tools provide views of both individual node state and communication traffic between nodes.

Both SensorSim projects are open source and free to use. However, the simulators are limited in their realism because (apart from SensorSim's power modules) neither simulator considers the limited resources of sensor nodes such as memory, and real-time computational capability. Moreover, it is not always required by the WSN to validate the functional correctness and/or, to provide performance guarantees. SensorSim simulates the complete WSN protocol stack, although this can be regarded as overkill and adding unnecessary complexity as this is not required in order to simulate the expected behaviour. This makes the SensorSim platform complex and difficult to use.

B. TOSSIM

There are platforms specifically designed to simulate WSNs, such as TOSSIM [1] which is a part of the TinyOS development efforts [2]. TOSSIM is a discrete-event simulator for TinyOS applications [13]. It aims to assist TinyOS application development and debugging by compiling applications into the TOSSIM framework, which runs on a PC instead of compiling them for a mote. Using the TOSSIM framework, programs can be directly targeted to motes without modification. This gives users a bigger margin to debug, test, and analyse algorithms in a controlled and repeatable environment. In TOSSIM, all nodes share the exact same code image, simulated at bit granularity, and assuming static node connectivity is known in advance. Therefore, TOSSIM is more of a TinyOS emulator than a general WSN simulator. It focuses on simulating TinyOS rather than simulating the real world. This has the advantage that the developed algorithms can be tested on a target platform. However, this may place some restrictions of the target platform on the simulation. TOSSIM is not always the right simulation solution; like any simulation, it makes several assumptions about the target hardware platform, focusing on making some behaviour accurate while simplifying others [1]. TOSSIM can be used as a tool for absolute evaluation of some causes of the behaviour observed in real-world network deployments.

C. TOSSF

TOSSF [14] is a simulation framework that compiles a TinyOS application into the SWAN [15] simulation framework. It can be viewed as an improvement over TOSSIM with a primary focus on scalability. It allows simulation of a heterogeneous collection of sensor nodes and a dynamic network topology. TOSSF suffers from potentially long test-debug cycles because it does not provide a scripting framework for experimentation. Although it enables development of custom environmental models, the absence of a scripting framework requires those models to be compiled into the simulation framework. Given that both of these simulators are tightly coupled with TinyOS, they may be unsuitable for early prototyping, or developing portable WSN applications.

D. GloMoSim

GloMoSim [16] is a scalable simulation environment for wireless and wired network systems. Its parallel discrete-event design distinguishes it from most other sensor network simulators. Though it is a general network simulator, GloMoSim currently supports protocols designed purely for wireless networks. GloMoSim is built using a layered approach similar to the seven layer network architecture of the OSI model. It uses standard APIs between different simulation layers to allow rapid integration of models developed at different layers, possibly by different users.

As in NS-2, GloMoSim uses an object-oriented approach, however for scalability purposes; each object is responsible for running one layer in the protocol stack of every node. This design strategy helps to divide the overhead management of a large-scale network. GloMoSim has been found to be effective for simulating IP networks, but it is not capable of simulating sensor networks accurately [17]. Moreover, GloMoSim does not support phenomena occurring outside of the simulation environment, all events must be gathered from neighbouring nodes in the network. Finally, GloMoSim stopped releasing updates in 2000 and released a commercial product called QualNet.

E. Qualnet

Qualnet is a commercial network simulator tool released by Scalable Network Technologies [18] that is derived from GloMoSim. Qualnet significantly extends the set of models and protocols supported by GloMoSim. It also provides a comprehensive set of advanced wireless modules and user-friendly tools for building scenarios and analysing simulation results. Qualnet is a discrete-event simulator, as such, it is event driven and time aware. It uses a layered architecture that is run by each node. When a protocol resides in a particular layer at one node, the packets are passed down crossing the remaining layers at the sending node, across the network, and then up to the protocol stack at the receiving node. Qualnet has a modular design and an intuitive GUI that make it easy to use to learn and modify.

F. OPNET

OPNET [19] is a further discrete event, object oriented, general purpose network simulator. The engine of OPNET is a finite state machine model in combination with an analytical model. It uses a hierarchical model to define each characteristic of the system. The top hierarchy level contains the network model, where the topology is designed. The second level defines the data flow models. The third level is the process editor, which handles control flow models defined in the second level. Finally, a parameter editor is included to support the three higher levels. The hierarchical models result in event queues for a discrete event simulation engine and a set of entities that handle the events. Each entity represents a node which consists of a finite state machine which processes the events during simulation.

Unlike NS-2 and GloMoSim, OPNET supports modelling sensor-specific hardware, such as physical-link transceivers and antennas. It also enables users to define custom packet formats. An attractive feature of OPNET is its capability of recording a large set of user defined results. Furthermore, the GUI (Graphical User Interface), along with the considerable amount of documentation and study cases that come along with the license are another attractive feature of the simulator. This GUI interface can also be used to model, graph, and animate the resulting output. The network operator is provided with editors that are required to simplify the different levels of modelling. Though model parameters can be changed, the simulation accuracy is influenced because OPNET is not open source software. Similar to NS-2, the object-oriented design of OPNET causes scalability problems. It does not have a high number of protocols publicly available possibly because of source code licensing constraints. Finally, OPNET is only available in commercial form.

The second class of simulators are application-oriented simulators, including EmStar [20], SENS [21], J-Sim [22], Shawn [23], and Dingo [24].

G. EmStar

EmStar [20] is a component based, discrete-event framework that offers a range of run-time environments, from pure simulation, distributed deployment on iPAQs [25], to a hybrid simulation mode similar to SensorSim. Emstar supports the use of simulation in the early stages of design and development by providing a range of simulated sensor network components, including radios, which provide the same interfaces as actual components. It supports hybrid mode with some actual components and some simulated components, and full native mode with no simulated components. As in TOSSIM, EmStar uses the same source code that runs at each of these levels to run on actual sensors. Amongst other simulators, such as TOSSIM, EmStar provides an option to interface with actual hardware while running a simulation. EmStar is compatible with two different types of node hardware. It can be used to develop software for Mica2 motes [26] and it also offers support for developing software for iPAQ based microservers. The development cycle is the same for both hardware platforms. The next step in the development cycle following the simulation is data replay. In this model, EmStar uses data collected from actual sensors in order to run its simulation. Leading directly from this, Emstar uses the half-simulation methodology similar to SensorSim's, where the software is running on a host machine and interfacing with a real physical communication channels. The final step in the development cycle is deployment.

EmStar combines many of the features of other WSNs simulators. Its component based design allows for fair scalability. Moreover, each aspect of the network can be logically fine-tuned due to its development cycle design. Because it targets a particular platform, many protocols are already available to be used. At the deployment step in the development cycle, only the configuration files have to be designed. This potentially adds constraints on the user as they must either ensure that the hardware configuration being used matches the existing configuration file, or they must write their own files.

The main goal of Emstar is to reduce design complexity, enabling work to be shared and reused, and to simplify and accelerate the design of new sensor network applications. While not as efficient and fast as other frameworks like TOSSIM, Emstar provides a simple environmental model and network medium in which to design, develop and deploy heterogeneous sensor network applications. When used as a migration platform from code to real sensor environment, the environment model may be sufficient for most developers. Another drawback of Emstar is that the simulator supports only the code for the types of nodes that it is designed to work with.

H. SENS

SENS [21] is a customisable component-based simulator for WSN applications. It consists of interchangeable and extensible components for applications, network communication, and the physical environment. In SENS, each node is partitioned into four main components: application, simulates the software application of the sensor node; network, handles incoming and outgoing packets; physical, reads sensed information; and environment, network propagation characteristics. Multiple different component implementations offer varying degrees of realism. For example, users can choose between various application-specific environments with different signal propagation characteristics. As in TOSSIM, SENS source code can be ported directly into actual sensor nodes, enabling application portability. Moreover, it provides a power module for development of dependable applications.

SENS defines three network models that can be used. The first successfully forwards packets to all neighbours, the second delivers with a chance of loss based on a fixed probability, and the third considers the chance of collision at each node. The physical component includes the non-network hardware for the sensor such as the power, sensors, and actuators. At a lower level, the environment component models the physical phenomena and the layout. The layout model includes different types of surfaces, each affecting radio and sound propagation in a different way.

SENS is less customisable than many other simulators, providing no chance to alter the MAC protocol, along with other low level network protocols. SENS uses one of the most sophisticated environmental models and implements the use of sensors well. However, the only measurable phenomenon is sound.

I. J-Sim

J-Sim [22] is a component-based discrete event simulator built in Java and modelled after NS-2. The design of this simulator aims at solving many of the shortcomings of comparable object-oriented simulators like NS-2. J-Sim uses the concept of components instead of the concept of having an object for each individual node. J-Sim uses three top level components: the target node which produces stimuli, the sensor node that reacts to the stimuli, and the sink node which is the ultimate destination for stimuli reporting. Each component is broken into parts and modelled differently within the simulator; this eases the use of different protocols in different simulation runs.

J-Sim claim has several advantages over NS-2 and other simulators. First its component based architecture scales better than the object oriented model used by NS-2 and other simulators. Second, J-Sim has an improved energy model and the ability to simulate the use of sensors for phenomena detection. Like SensorSim, there is support for using the simulation code for real hardware sensors. However, J-Sim is comparatively complicated to use. While no more complicated than NS-2, the latter simulator is more popular and accepted in the sensor network research community and more community support is available, therefore, more people are keen to spend the time to learn how to use it.

Though it is scalable, J-Sim has a set of inefficiencies. First, there is unnecessary overhead in the intercommunication model. The second problem is inherited by most sensor networks simulators that are built on top of general purpose simulators, 802.11 is the only MAC protocol that can be used in J-Sim. Finally, Java is possibly less efficient than many other languages.

J. Dingo

Dingo [27] provides a workbench for prototyping algorithms for WSNs taking a top-down design methodology. Having no target platform means the full functionality of a programming language can be used. This eases the design process as prototype algorithms can be tested before optimisation for the target platform. Dingo consists of a fixed API, with customisable internals. It has a simple graphical user interface and a set of base classes, which are extended by the user to create simulation. Each simulated sensor node runs in its own thread and communicates using the same protocols that would be deployed on a physical node. Sensors are modelled using a pool of concurrent, communicating threads. Individual sensors are able to: (1) Gather and process data from a model environment; (2) Locate and communicate with their nearest neighbours; (3) Determine whether they are operating correctly and act accordingly to alter the network topology in case of faulty nodes being detected. Nodes may be configured differently to simulate a heterogeneous sensor network. Dingo comes with a set of application level routing packages including simple multi-hop flooding, MuMHR [28] and LEACH [29].

Dingo features a significant improvement in the simulation performance by giving the option to split the visualisation from the simulation. It provides tools for the simulation and deployment of high-level, Python code on real sensor networks. For example, Dingo-boom provides a two-way interface between MoteIV's Boomerang class motes and Dingo. Dingo-top is another tool which is used to dump network topology data to a text file and generate a graphical representation of that topology. Furthermore, Dingo has several features in the form of plugins. These can be activated/deactivated on the plugin menu.

As with SensorSimII, Dingo provides an extensible visualisation framework that aims at easing the life for sensor network debugging, assessment, and understanding of the software by visualising the sensor network topology, the individual node state, and the transmission of the sensed data. Dingo comes with an interface between the simulation environment and different hardware platforms, for example the Gumstix [30] platform. Also, Dingo allows mixed-mode simulation using a combination of real and simulated nodes. In Dingo, nodes have the ability to obtain their sensed data from a database or graphical objects like maps; this improves the fidelity of simulations as it makes it possible to check the simulation results against the real data.

Dingo focuses on the protocols and algorithms for higher layers of network state but it does not directly support sensor networks at the physical layer. It has major drawbacks which limit its functionality. Most of these drawbacks are due to the incomplete nature of the tool. These drawbacks are: (1) The lack for Media Access Control or MAC layer, communications to be handled by point-to-point systems. (2) No collision management procedure, partly due to the absence of the MAC layer.

K. NS-3

NS-2 [31] is an object-oriented discrete event simulator targeted at networking research. It is an open source network simulator originally designed for wired, IP networks. The NS-2 simulation environment offered great flexibility in studying the characteristics of WSNs because it includes flexible extensions for WSNs. NS-2 has a number of limitations: (1) It puts some restrictions on the customisation of packet formats, energy models, MAC protocols, and the sensing hardware models, which limits its flexibility; (2), the lack of an application model makes it ineffective in environments that require interaction between applications and the network protocols. (3) It does not run real hardware code; (4) It has been built by many developers and contains several inherent known and unknown bugs. (5) It does not scale well for WSNs due to its object-oriented design; (6) Using C++ code and oTcl scripts makes it difficult to use.

To overcome the above drawbacks the improved NS-3 simulator [10] was developed. NS-3 supports simulation and emulation. It is totally written in C++, while users can use python scripts to define simulations. Hence, transferring NS-2 implementation to NS-3 require manual intervention. Besides the scalability and performance improvements, simulation nodes have the ability to support multiple radio interfaces and multiple channels. Furthermore, NS-3 supports a real-time schedule that makes it possible to interact with a real systems [10]. For example, a real network device can emit and receive NS-3 generated packets.

L. Shawn

Shawn is an open source discrete event simulator for WSNs. It is written in C++ and can be run in Linux/Unix and Windows environments. Shawn aims to simulate large- scale WSNs, where physically accurate simulations fail. The idea behind Shawn is to use abstract models to simulate the affects of a phenomenon rather than the phenomenon itself [23]. Users of Shawn can adapt the simulation to their needs by selecting the application preferred behaviour. The authors claim that Shawn provides a high abstraction level that hides a lot of the simulation details. Users are given full access to the communication graph, which allows them to observe nodes and their data [23]. However, there are some limitation in Shawn, for instance: Visualization output is not supported, MAC module is not extent, and also users need to do much programming [32].

III. Discussion

Generally, real WSNs testbeds provide a more accurate, realistic, and replicable validation mechanism for algorithms and protocols. However, the cost of deployment and maintenance of large-scale testbeds limits their applicability. Moreover, the wide variety of available sensor hardware can make it rather difficult to replicate any results produced by real testbeds. Besides, in some applications, where dangerous conditions are being studied, e.g. chemical pollution, a real testbed is an unwanted choice. Out of these restrictions came the need for simulation as a tool for validating and testing algorithms/protocols. As shown in Section II, simulation tools are widely available and used by WSNs researchers. However, most of the existing simulators are incomplete and follow different approaches to investigate different problems. The variety of existing simulation tools has led to accuracy and authenticity issues that concern even the best simulators available today. Such issues also make it even more difficult to replicate and compare evaluation results from competing simulation systems. Simulation drawbacks also include the lack of visualisation tools, GUI's, poor documentation, absence of examples, amongst others.

To solve the dilemma of having an accurate but scalable and low-cost prototyping solution, we suggest the use of mixed-mode simulation as an effective midrange solution. Mixed-mode simulation is the integration of a simulated environment and a real testbed to improve both the accuracy and scalability of testing results. In other words, the mixed-mode simulation enables the simulation of algorithms partially in software and partially in a real hardware WSN testbed. A small number of simulation tools like NS-3 and Dingo already support this mode of simulation. This simulation mode allows researchers to compare the results of running the same algorithm in both simulation and on physical sensor hardware; the comparison allows the inclusion or the modelling of more realistic conditions in the simulation environment. A flexible mixed-mode simulator should support integration of heterogeneous sensor devices. Also, the simulation-testbed interaction remains a challenging task that needs to be addressed. For instance, the authors of Dingo describe in [33] a new Python library that implements synchronous message-passing concurrency to improve coordination between many hosts.

Yet, the choice of a suitable simulator is a difficult decision. There is no 'best' simulator; each simulator has specific features that work well in certain circumstances. The selection of a simulator depends mostly on the algorithmic feature to be evaluated. High level simulators like NS-2 gives an estimation about the applications and some middleware behaviour. Mid-level simulators, e.g. OMNET, provides more information about the physical layer components that are simulated without giving too much details. Low-level simulators provide accurate bit level estimations of the hardware as well as software performance. Regardless of the simulator, any simulations will always have weaknesses either due to non-realistic assumptions or modelling errors that may be present in the algorithm itself. Therefore, developing formal methods, e.g. using graph theory [34], to verify the correctness of new algorithms and protocols is also part of the testing or evaluation research.

Table 1 summarise and compares the reviewed simulation tools.

TABLE 1. SUMMARY ABOUT REVIEWED SIMULATION TOOLS

Simulators	Programming Language	GUI	General or Specific Simulator	Open Source	Main Features	Limitations
SensorSim	C++	No	Specifically designed for WSNs	Yes	-Power and communication protocol models Sensing channel and sensor models -Scenario generation -Support for hybrid simulations	-Limited in SensorSim project realism -Consider limited resources of sensor nodes. -Simulates the complete WSN protocol stack
TOSSIM	C++	Yes	Specifically designed for WSNs	Yes	-Can be targeted to motes without modification -Nodes share the exact same code image -The developed algorithms can be tested on a target platform	-Makes several assumptions about the target hardware platform -Focusing on making some behaviour accurate while simplifying others
TOSSF	C++	Yes	Specifically designed for WSNs	Yes	-Primary focus on scalability -Support heterogeneous nodes and dynamic topology	-Long test-debug cycles
GloMoSim	C/Parsec	Yes	General	Yes	-Supports protocols designed purely for wireless networks. -Built using a layered approach. -Uses standard APIs between different simulation layers.	-Not scapable of simulating sensor networks accurately -does not support phenomena occurring outside of the simulation environmen
Qualnet	C/C++	Yes	General	Comm-ercial	-Comprehensive set of advanced wireless modules and user-friendly tools	- The annual license is expensive
OPNET	C/C++	Yes	General	Comm-ercial	-Uses a hierarchical model to define each characteristic of the system -Capability of recording a large set of user defined results	- scalability problems
EmStar	C	Yes	Specifically designed for WSNs	Yes	-Supports hybrid mode -Provides an option to interface with actual hardware while running a simulation -Compatible with two different types of node hardware	-Supports only the code for the types of nodes that it is designed to work with
SENS	C++	No	Specifically designed for WSNs	Yes	-Multiple different component implementations	-Less customisable -Only measurable phenomenon is sound
J-Sim	Java	Yes	Specifically designed for WSNs	Yes	-Ability to simulate the use of sensors for phenomena detection -Support for using the simulation code for real hardware sensors	-Comparatively complicated to use -Unnecessary overhead in the intercommunication model
Dingo	Python	Yes	Specifically designed for WSNs	Yes	-Full functionality of a programming language can be used -Option to split the visualisation from the simulation	-Does not directly support sensor networks at the physical layer -Incomplete nature of the tool
NS-3	C++	No	General	Yes	-Supports simulation and emulation -Supports a real-time schedule -Ability to support multiple radio interfaces and multiple channels	- Some restrictions on the customisation. -Lack of an application model -Does not run real hardware code -Does not scale well for WSNs
Shawn	C++	No	Specifically designed for WSNs	Yes	-Able to simulate large- scale WSNs -Ability of selecting the application preferred behaviour -Full access to the communication graph	-Does not support visualization output -MAC module is not extent -Lots of programing is required

IV. Conclusion

This paper provides a comprehensive review of simulation tools that are widely used in the field of WSNs. The aim is to help researchers choosing the most appropriate simulation tools to evaluate their work. There are a variety of simulation tools with different capabilities. However, the authors believe that they are insufficient for testing and evaluating WSNs algorithms. This is because the simulation results can be unrealistic due to the incomplete or inaccurate simulation models. An immediate measure is to develop unified models, e.g. energy, for different simulators. This allows realistic comparisons between results produced by different simulators to be made. To improve authenticity and accuracy of simulation results, it is important that researchers make their simulation code available for download by other researchers. Moreover, researchers should dedicate more space in their papers to clearly describe their simulation setup. On the other hand, large-scale real testbeds are still infeasible due to their cost and complexity. It can be easily observed that the trend in the WSNs field is to use mixed-mode simulation as an interim solution. Finally, we believe that theoretical validation of algorithms can serve as a good means for evaluating many WSNs algorithms.

References

[1] P. Levis, N. Lee, M. Welsh, and D. Culler, “TOSSIM: accurate and scalable simulation of entire TinyOS applications,” SenSys '03: Proc. of the 1st int.conference on Embedded networked sensor systems, 2003, pp. 126-137.

[2] P. Levis, et al., “TinyOS: An Operating System for Sensor Networks,” In Ambient Intelligence, 2005.

[3] IARIA, “The Fifth International Conference on Sensor Technologies and Applications, SENSORCOMM 2011,” 2006-2011; http://www.iaria.org/conferences2011/SENSORCOMM11.html.

[4] S. Kurkowski, T. Camp, and M. Colagrosso, MANET Simulation Studies: The Current State and New Simulation Tools, The Colorado School of Mines, 2005.

[5] E. Larsen, et al., “iOLSR: OLSR for WSNs Using Dynamically Adaptive Intervals,” The Fifth Int.l Conference on Sensor Technologies and Applications, 2011, pp. 18 to 23.

[6] K. Shi, Z. Deng, and X. Qin, “TinyMQ: A Content-based Publish/Subscribe Middleware for Wireless Sensor Networks,” Proc. SENSORCOMM 2011, The Fifth Int. Conference on Sensor Tech. and Applications, 2011, pp. 12 to 17.

[7] R. Behnke, J. Salzmann, P. Gorski, and D. Timmermann, “HDLS: Improved Localization via Algorithm Fusion,” The Fifth Int. Conference on Sensor Tech. and Applications, 2011.

[8] S. Feldman and M. Feldman, “Tree-Based Organization for Very Large Scale Sensor Networks,” The Fifth Int. Conference on Sensor Technologies and Applications 2011, pp. 45 to 50.

[9] F. Derogarian, J. Ferreira, and V. Tavares, “A Routing Protocol for WSN Based on the Implementation of Source Routing for Minimum Cost Forwarding Method,” Proc. SENSORCOMM 2011, The Fifth International Conference on Sensor Technologies and Applications, 2011, pp. 85 to 90.

[10] nsnam, “NS-3,” 2011; from http://www.nsnam.org/.

[11] S. Park, A. Savvides, and M.B. Srivastava, “SensorSim: a simulation framework for sensor networks,” MSWIM '00: Proceedings of the 3rd ACM int. workshop on Modeling, analysis and simulation of wireless and mobile systems, 2000.

[12] C. Ulmer, “Wireless Sensor Probe Networks-SensorSimII,” 2007.

[13] xbow, “Mica Mote,” 2007.

[14] L.F. Perrone and D.M. Nicol, “A Scalable Simulator For TinyOS Applications,” Simulation Conference, 2002. Proceedings of the Winter, vol. 1, 2002, pp. 679 - 687.

[15] A. Gahng-Seop, T.C. Andrew, V. Andras, and S. Li-Hsiang, “Supporting Service Differentiation for Real-Time and Best-Effort Traffic in Stateless Wireless Ad Hoc Networks (SWAN),” IEEE Transactions on Mobile Computing, vol. 1, no. 3, 2002, pp. 192-207; DOI 10.1109/tmc.2002.1081755.

[16] X. Zeng, R. Bagrodia, and M. Gerla, “GloMoSim: a library for parallel simulation of large-scale wireless networks,” PADS '98: Proceedings of the twelfth workshop on Parallel and distributed simulation, 1998, pp. 154-161.

[17] D. Curren, “A survey of simulation in sensor networks,” 2005.

[18] S.N. Technologies, “QualNet Simulator,” from http://www.scalable-networks.com/products/qualnet/.

[19] X. Chang, “Network simulations with OPNET,” WSC '99: Proceedings of the 31st conference on Winter simulation, 1999, pp. 307-314.

[20] L. Girod, et al., “Emstar: A software environment for developing and deploying heterogeneous sensor-actuator networks,” ACM Trans. Sen. Netw., vol. 3, 2007, pp. 13.

[21] S. Sundresh, W. Kim, and G. Agha, “SENS: A Sensor, Environment and Network Simulator,” The 37th Annual Simulation Symposium (ANSS37), 2004.

[22] A. Sobeih, et al., “J-Sim: A Simulation Environment for Wireless Sensor Networks,” ANSS '05: Proceedings of the 38th annual Symposium on Simulation, 2005, pp. 175-187.

[23] S.P. Fekete, A. Kroller, S. Fischer, and D. Pfisterer, “Shawn: The fast, highly customizable sensor network simulator,” Proc. Networked Sensing Systems, 2007. INSS '07. Fourth Int. Conference on, 2007, pp. 299-299.

[24] S. Mount, R.M. Newman, E. Gaura, and J. Kemp, “SenSor: an Algorithmic Simulator for Wireless Sensor Networks,” In Proceedings of Eurosensors 20, vol. II, 2006, pp. 400-411.

[25] hp, “iPAQs,” 2000.

[26] xbow, “Mica Mote,” 2012.

[27] S. Mount, “Dingo Wireless Sensor Networks Simulator,” 2008.

[28] H. Mohammad, K. Alexander, and G. Elena, “MuMHR: Multi-path, Multi-hop Hierarchical Routing,” SENSORCOMM '07: Proceedings of the 2007 International Conference on Sensor Technologies and Applications, 2007, pp. 140-145.

[29] W. Heinzelman, et al., “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” Proceedings of the 33rd Int. Conference on System Sciences, 2000.

[30] Gumstix.com, “Gumstix way small computing,” 2007.

[31] NS-2, “The Network Simulator,” 2007.

[32] E. Kolega, V. Vescoukis, and D. Voutos, “Assessment of network simulators for real world WSNs in forest environments,” Proc. Networking, Sensing and Control (ICNSC), 2011 IEEE Int. Conference on, pp. 427-432.

[33] M. Hammoudeh, “Modelling Clustering of Sensor Networks with Synchronised Hyperedge Replacement,” ICGT '08: Proceedings of the 4th international conference on Graph Transformations, 2008, pp. 490-492.

[34] S. Mount, M. Hammoudeh, S. Wilson, and R. Newman, “CSP as a Domain-Specific Language Embedded in Python and Jython,” Proc. Comm. Process Architectures IOS Press, 2009.

Elaboration of Cognitive Decision Making Methods in the Context of Symbiotic Networking

Milos Rovcanin, Eli De Poorter, Opher Yaron, Ingrid Moerman, David Plets, Wout Joseph, Luc Martens
Ghent University - IBBT, Department of Information Technology (INTEC), IBCN-WiCa
Gaston Crommenlaan 8, Bus 201, 9050 Ghent, Belgium
milos.rovcanin@intec.ugent.be, {*firstname.lastname*}@intec.ugent.be

***Abstract*—Recently, the concept of 'cognitive networking' has been introduced, in which reconfigurable radio networks rely on self-awareness and artificial intelligence to optimize their network performance. These cognitive networks are able to perceive current network conditions and then plan, learn and act according to end-to-end goals. This paper elaborates on different methods (network solutions) that can be used by cognitive networks for deciding on how to optimize the performance of a large number of co-located devices with different characteristics and network requirements. To this end, a negotiation based networking methodology ('symbiotic networking') is used that supports efficient network cooperation between heterogeneous devices in order to optimize their network performance. In this paper, the advantages and disadvantages of different reasoning techniques that can be used during the decision making phase are discussed.**

***Index Terms*—symbiotic cognitive networks; reasoning methods; machine learning; game theory;**

I. Introduction

Wireless networks are becoming increasingly complex, heterogeneous and dynamic, which motivate the evolution of the concept of cognitive networks. As it is described in [1]: ”A wireless cognitive network is a network with a cognitive process that can perceive current network conditions and then plan, decide and act on those conditions”. The network can learn from these adaptations and use them to make future decisions, all while taking into account end-to-end goals.

Cognition process (Fig.1), in this case, is related to machine learning [2] as any algorithm that: ”improves its performance through experience gained over a period of time without the complete information about the environment in which it operates”. This process can be divided into four stages:

(i) *Gather observations* of the important aspects (properties) of the network or, in other words, gather knowledge. Inside complex systems, with a large number of nodes, it is more likely that the cognitive process is performed with an incomplete knowledge about the system status. (ii) *Plan actions* according to the network policies and the knowledge that was gathered. Reasoning is used to decide which scenario best fits the end-to-end goals. After the decision is made, (iii) actions are performed accordingly. The (iv) cognitive feedback loop measures the success of the chosen solution relative to the defined objectives. This way, when similar circumstances happen in the future, the cognitive decision maker will have an idea what decisions are preferred and which ones should be avoided.

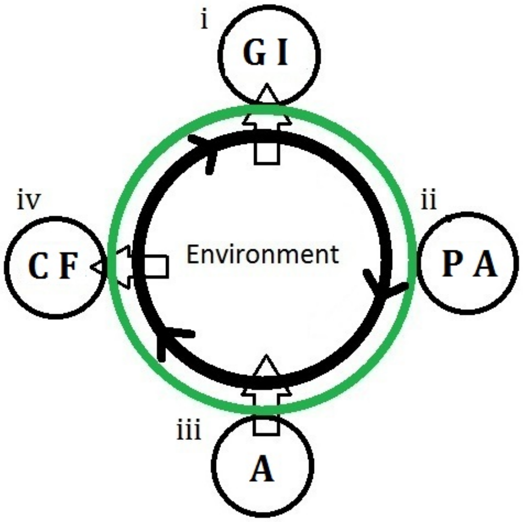


Fig. 1. A four stage cognition cycle: (1) data gathering (DG), (2) planning (PA), (3) acting (A), (4) collecting feedback (CF)

The fundamental difference between a cognitive network and a cognitive radio are the end-to-end goals. They give a cognitive network its network-wide scope, separating it from other adaptation approaches, which usually have only a local, single element scope [3].

Cognitive networking solutions can be applied to improve network performance in situations in which different networks cooperate in an ad-hoc and dynamic way. For example, co-located IEEE802.11 WiFi, IEEE802.15.4 sensor and 802.15.1 Bluetooth networks are typically configured independently from each other: their settings take into account only the behaviour of their own communication technology and they typically ignore their influence on each other. By ensuring that different co-located networks are aware of each other, they can modify their configuration so that the performance of all individual networks improves. The cognition loop can be used as the way to further improve the newly formed symbiotic network performance. Some of the examples are: improving reliability, decreasing energy consumption, lowering exposure etc.

The remainder of the paper is organised as follows: The SymbioNets use case, a concept of symbiotic wireless sensor networks cooperation, is presented in Section II. Section III brings the overview of the most commonly used cognitive methods in the context of wireless (sensor) networking. In Section IV we compare suitability of the above mentioned approaches, from several different aspects, for the symbiotic networking. Conclusion is given in Section V.

II. The SymbioNets use case

The need for cognitive networking is also stated in the SymbioNets project [4]. In this project, different networks engage in cooperation by activating specific symbiotic network services (Fig.2), when activating these services result in better network performance for all involved networks. These symbiotic network services are not crucial for the correct operation of the individual networks, but instead influence the behaviour of the communication.

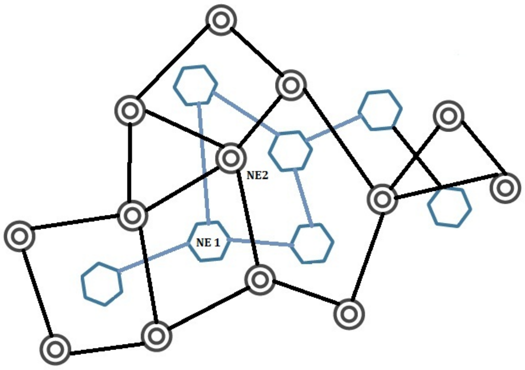


Fig. 2. Coexisting networks suitable for symbiotic cooperation with their negotiation representatives (NE). An example would be: temperature monitoring network (black nodes) and a security network (blue hexagons) inside a facility

By activating a symbiotic network service, the reliability, energy consumption, exposure, etc. of one or more networks is influenced. In the SymbioNets project, a number of symbiotic services is dynamically activated or deactivated based on the network requirements. Some example symbiotic network services are the following:

- A network can offer Internet access to other networks
- An 'interference avoidance' algorithm can be activated to reduce interference
- A 'packet sharing service' allows cooperating networks to interpret and route packets from each other

The full SymbioNets cognitive cycle works as follows. Co-located communities of devices exchange their service profiles so that each one of them has an idea about the network preferences of co-located devices. These preferences describe behavioural aspects of the network (e.g., 'limit the battery consumption') or express the need for additional functionality (e.g., 'get Internet access'). There is a negotiation entity (NE) in every community which collects all service profiles of neighbouring communities and decides which symbiotic network services should be activated or deactivated. The process of negotiation is rerun every time a new or an updated profile is received. Service negotiation messages are exchanged between negotiation entities to reach a common decision. Then, the decision is disseminated to every node in the community.

One critical aspect of the SymbioNets approach is that the networks need to decide which network services to activate based not only on environment observations, but also in networks consisting of strongly heterogeneous devices with diverging communication technologies, diverging application requirements and diverging computational resources. In the current implementation, a linear programming algorithm, that is implemented on the negotiation node [4], is used to automatically calculate the optimal set of network services to be enabled on every node. However, this solution assumes that benefits and costs of enabling each service are predefined and fixed. In a dynamic environment this assumption usually does not hold. Instead, a more advanced cognitive decision method is required. In this paper, we describe a number of cognitive decision methods that can be used and explain the concepts on which they rely. Most importantly, we point out their major advantages and drawbacks if applied in heterogeneous, symbiotic environments. These cognitive methods are evaluated on several criteria:

- *Complexity*. Are they also suitable for use on devices with low processing power, such as sensor nodes?
- *Support for heterogeneity*. Suitable for an environment where different communication technologies and different types of application requirements coexist.
- *Dealing with "'malicious"' behaviour* in the network. For example, what if certain devices try to 'cheat' by falsely reporting that they activated certain services?
- *Support for distributed solutions*. Suitability for networks where devices have to make decisions based on local information, rather than having a full network overview.

III. Cognitive methods

This section gives an overview of relevant decision making methods and elaborates on their advantages and suitability for heterogeneous, symbiotic networks. We discuss the following decision methods: mathematical methods (including linear programming), interactive network simulators, game theory and machine learning.

A. Mathematical methods

In small networks, it may possible to calculate the influence of a specific network configuration on the network performance. As an example, consider *queuing theory*, that can be used to derive performance metrics such as the average waiting time in queues of the system, the expected number of packets in queues, packet drop probabilities, etc. However, queuing theory is not capable of predicting end-to-end network qualities in large-scale and complex networks that use multiple network protocols.

Other mathematical models were developed to calculate the performance of higher-layer network protocols. For example, in [5] a mathematical model is created to optimize the stability and fairness of rate control algorithms for the Internet. However, these formulas require in-depth knowledge of the innerworking of the specific network protocols.

Finally, in [4], a linear program is used to calculate which, among a list of potential network functions, should be activated to best suit the network requirements of several co-located networks. For example, if energy efficiency of a smartphone device is an issue and no low-latency applications are being used, the linear program can activate a packet aggregation protocol. For its calculations, the linear program assumes that the influence of network protocols is known in advance.

All these mathematical methods have the same disadvantages. (i) They do not accurately model the whole networks, or use abstractions of certain parts of the network. (ii) They are not suitable for modeling complex cross-layer and cross-network interactions. (iii) And finally, they assume perfect knowledge about the network. On the other hand, mathematical formulas often have a low processing overhead and they can accurately predict a number of key performance metrics.

B. Network simulators

As an alternative, it is possible to use a *network simulator* in the decision making process. The cognitive engine can recreate (to the best of its abilities) the monitored network conditions in a network simulator, and use the simulation to predict the influence of optimization decisions. Existing network simulators such as ns2/ns3, OPNET, Netsim, etc. have become increasingly accurate and are capable of taking into account many cross-layer and even physical layer interactions. A similar approach is the use of *planning tools* for decision making. Planning tools are typically used before deployment of a network to predict the behaviour of (wireless) networks in different environments [6]. They are, for example, used to calculate the optimum number of Wi-Fi access points and communication settings to obtain a certain network performance.

A cognitive reasoning engine can be created by recreating the network set-up in a network simulator or planning tool, and running multiple scenarios with different settings. This way, the decision engine can estimate and select the best network configuration.

The main disadvantages with this approach are the following. (i) Network simulators are typically not designed for distributed calculations. (ii) Detailed information about the network is required, such as the location of each device, the exact network configuration and settings and the application requirements. (iii) Current network simulators do not take into account the influence of different co-located communication technologies that use the same frequency band. Finally, (iv) network simulators often require heavy processing power.

C. Game theory

Game theory models the behaviour of a system as a game, played by at least two rational players. Rationality, in this context, means balancing costs against benefits to arrive at an action that maximizes personal payout. Game theory has been used in logic, economics, psychology, political science, biology etc. In the domain of computer networks, it offers tools that can be used in modelling an interaction among individual nodes in a wireless network. Game theory enables us to determine the existence, uniqueness and convergence to a steady state operating point.

Formally, a game is given by $G = (N, A, u_i)$, where $N = 1, 2, ..., n$ is the set of players, A_i is the action set for player i. u_i is the set of utility functions that each player wishes to maximize, where $u_i : A_i \rightarrow \Re$. For every player, i, the utility function is a function of the action chosen, a_i and the actions chosen by all the players in the game other than player i, denoted as a_{-i}. Together a_i and a_{-i} make up the action tuple a. An action tuple is a unique choice of actions by each player.

Steady-state conditions, the Nash equilibrium, can be identified using this model. The Nash equilibrium represents a state where an individual node cannot increase the values of its utility function by changing actions, assuming other players remain constant in their strategies. Pareto efficiency, on the other hand, describes how acceptable is the achieved steady state from a global point of view. A Pareto optimal state is the one where no player can benefit from any action without making someone else worse off.

Problems such as node's power consumption, contention for a communication medium and routing in cognitive wireless networks have already been presented in the form of a game [7].

For our specific use case, described in Section II, the negotiation process should include a process of defining pricing policies and action sets, taking into account the wireless technology each community is based on. The disseminated list of enabled services defines the starting point in a game. Nodes aim to increase their utility functions regarding each given incentive and a properly designed pricing policy will lead to an increase of the Pareto efficiency. By sending feedback to the negotiation node, service policies can be updated to trigger recalculation and renegotiation. Since every network preference that needs to be optimized in the community demands a distinctive game, it is likely that a number of these different games will involve common parameters. Certain trade-offs can be foreseen. For example: a packet sharing and a power saving game can be played simultaneously. While the first one demands a node to forward more packets thus spend more energy, the second one will try to minimize the power consumption as much as possible. There has to be a trade-off between spending less energy per node and ensuring better network coverage and shorter routing paths.

One difficulty of using the game theory in SymbioNets, is how to correctly interpret low-level actions into high-level goals. Additionally, the set of available actions to the players needs to be carefully defined and verified as well as the states the system can go through. Finally, computational complexity will increase significantly as the number of nodes grows.

D. Machine learning

Machine learning is a form of artificial intelligence in which devices learn using inductive inference (Fig.3) [8] [9]. One specific field, *Reinforcement learning*, is particularly suitable for usage in the context of cognitive networks [10]. It involves the notion of learning through trial-and-error interaction with a dynamic environment.

This concept can be described as follows: a decision making node has a set of possible actions $A = a_1, a_2, ..., a_N$ at each step. Based on observations of the environment, the next step is chosen. It is desirable to have a process with a finite number of states. When the action is taken, the environment makes a transition to another state according to a certain probability distribution $P(s'|s, a)$ and so on. The main goal of a decision maker is to maximize its action-value function (Bellman's equation) :

$$Q(s, a) = r(s, a) + \gamma \sum_{s'} P(s'|s, a) max Q(s', a')$$

The Q function assigns a value to each state/action pair. This value is increased every time a decision maker takes action a at state s. It also takes into account the highest available Q value at each connected state. The factor γ makes sure the reward for making the same decision in the future decreases. $r(s, a)$ represents an immediate reward, the one that is awarded at the initial transition from the state s.

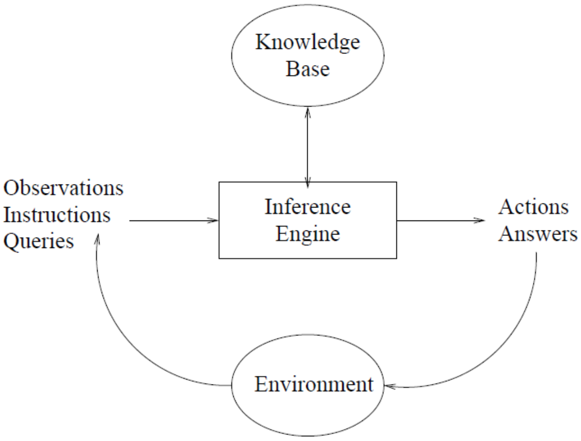


Fig. 3. The standard model of a learning agent

In [11] a number of reinforcement learning (RL) technique applications in wireless ad-hoc and wireless sensor networks are presented. Reinforcement learning is described as very well suited for distributed problems, like routing. It has ”medium” requirements for memory and computation at the individual nodes, is easy to implement, highly flexible to topology changes, but it needs some time to converge.

Q-learning algorithm [12] is probably the most frequently used technique of reinforcement learning in wireless ad-hoc networks. It does not need any model of the environment and can be used for on-line learning of the value function of some RL task, referred to as the Q-function. It is relatively easy to implement and has a good balance of optimality to memory and energy requirements. In wireless sensor networks, it was mostly used for optimizing the routing algorithms [13].

LSPI (Least Square Policy Iteration) is a model-free algorithm, which calculates the Q value of every state as a linear combination of so called ”weighted basis functions” [14].

$$Q(s, a, w) = \sum_k \phi_i(s, a) w_i$$

Each one of k basis functions $\phi_i(s, a)$ represents a certain information about each state-action pair (e.g., residual energy of s', link quality between s and s' etc.). Weights w_i are parameters of the linear equation that are learned by gathering samples $\langle s, a, r, s' \rangle$. Every sample describes the reward r received upon executing action a in state s, ending in state s'. The main advantages of this algorithms are: it converges more quickly than the Q-learning and it does not require carefully tuning initial learning parameters [15].

Additionally, in [16] the Collaborative Reinforcement Learning technique (CRL) is introduced as a model-based technique with collaborating RL agents. This algorithm is applicable in heterogeneous networks, where agents typically poses different capabilities. CRL allows newly discovered agents to negotiate the establishment of causally connected states with their neighbours by exchanging device capability information.

In the context of the SymbioNets use case, the negotiation process, described in Section II, is used to check the compatibility of neighbouring community's communication technologies. It aims at defining the initial values of all the configuration parameters regarding each enabled service in the community. These values are used as starting points in the cognition process. A periodic feedback (values of a predefined set of parameters) are sent to a negotiation entity. After investigating the progress that has been made, the negotiation node decides whether to renegotiate the service policy or not.

The process of learning should be applied to every established network preference. Since the best possible action at some state is never known a priori, the decision making node needs to try different actions and sequences of actions so it could learn from its experiences. The process of maximizing action-value functions assigned to each preference enables the node to discover an optimal action-policy. This policy maps input values (gathered from the environment) into actions. As mentioned in the previous section, multiple processes are performed in parallel, including some common parameters which might lead to conflicting situations. It is necessary to determine correct trade-offs. This analysis can be done during the negotiation phase.

One of the main issues with reinforcement learning is that the number of possible environment states is large. To calculate the reward of a certain action, a node has to, at least, take a good guess what state the system is going to move into by preforming an action. The other problem is how to gather all the necessary observations for multi-objective learning. Should there be a predefined set of data that is to be gathered at each step or should it depend on objective? Another pitfall is that, since the number of nodes will be fluctuating, it will be hard to establish a solid and non-changing policy since the influence of each action could change as the network topology changes.

IV. Choosing the optimal reasoning engine

This section compares the suitability of different reasoning engines for the symbiotic networking paradigm that was presented in Section II.

A. Supporting heterogeneity

Mathematical formulas typically assume that the network is very homogeneous. Heterogeneous networks might have diverging network requirements, different communication protocols and even use different communication technologies. Since new formulas have to be created for each different network situation, the use of mathematical formulas in not feasible for heterogeneous networks.

Current *network simulators* already have support for heterogeneity. They typically include a wide set of standardized network protocols and technologies, and as such can be used to estimate how different network settings influence performance. However, especially at the physical layer, a large number of unexpected cross-layer and cross-network interactions are typically simplified.

With regards to *game theory*, agents in heterogeneous networks can play a game, but with different perception of what goes on in the network, because the utility functions differ significantly - different power supply, radio transmission costs etc. This leads to different cost/benefit reasoning among nodes in different communities and has to be taken into account while negotiating optimal service and pricing policies.

Reinforcement learning algorithms support heterogeneity but every newly discovered agent has to negotiate the establishment of causally connected states with their neighbours by exchanging device capability information.

B. Complexity

The complexity of *mathematical formulas* depends strongly on the effect that is modelled (link local, end-to-end, ...) and the number of parameters that are taken into account. However, it is safe to say that highly complex networks can not fully be modelled using only mathematical formulas. In addition, solving complex mathematical formulas can be difficult on resource-constrained devices such as sensor nodes or smart phones.

The computational complexity of *network simulators* strongly depends on the accuracy of the simulator. Simulators that use approximations of communication effects can have low computational complexity, whereas extremely accurate simulators might require several days to calculate the behaviour of even small networks. However, even simplified network simulators are too complex to be used in resource-constrained networks: sensor networks or smart phones might want to off-load the cognitive reasoning process to a remote server that is less resource constrained.

In *game theory* approach, computation complexity increases as the number of players grows. Every individual player has to take into account actions of all other players. The existence of a steady state point must be proven before utilizing any specific algorithm. In the approach with the centralized authority (a negotiation node), the rules of a game can be changed during the play in respect of the feedback a negotiation entity is given from the network. Coordinated games assume exchange of information between nodes which can cost a lot in power consumption and memory space that is being used.

For *reinforcement learning*, model free algorithms need fresh data every once in a while, but the computational power and memory utilization are much lower than in model based algorithms. However, a model based algorithm can guarantee good results in an environment where acquiring measurements is highly expensive. With this approach, learning is faster, as much more use can be made of each experience. It can be used to solve multiple optimization problems. Many algorithms have already been implemented in WSNs, which implies that they can be adjusted to cope with severe power limitations. Complexity, however, does differ from algorithm to algorithm.

C. Dealing with ”‘malicious”’ behaviour

In order to increase their own performance, devices can report false values, or fail to activate the network configuration settings they promised. A cognitive engine should be able to detect this kind of behaviour, and optionally punish these devices, or refuse to further cooperate with them.

When the predicted outcome of *mathematical formulas* differs from the measured performance, this can be an indication of ‘cheating’ nodes. However, it is almost impossible to detect the difference between incorrect predictions and malicious behaviour.

When the observed performance differs from the expected performance, *network simulators* can run through a large number of simulations to identify which settings produce similar behaviour. However, this process is likely very expensive computationally.

In *game theory*, a properly designed pricing policy aims to increase the Pareto efficiency of a game by making sure that every asocial behaviour is ’punished’. A node is considered to be malicious (selfish) if it tries to increase its own benefit without taking into account the social aspect of the game (making Pareto efficiency worse off).

When using *reinforcement learning*, Higher learning rates and not well defined rewarding policy will enforce particular states very quickly. In a dynamic network this can lead to suboptimal performance. One should trade-off between converging faster, but to a possibly mediocre operating point and converging at a slower rate while taking into account network changes and possible erroneous behaviour.

D. Suitability for distributed implementation

Mathematical methods typically require complete knowledge of the network configuration and as such can not be used in a distributed way. The same is true for *network simulators*.

Game theory approach is naturally intended to operate in a distributed manner. Every node plays a game with one objective: to increase its payoff as much as possible. As mentioned before, to avoid selfish behaviour, the pricing policy must be properly defined. The existence of a supervising node, which gathers feedback information of all the players in a community and changes the rules and pricing policies on the fly, can be helpful, but fully centralized approach is inapplicable since it demands great computational power and full knowledge of the network.

Centralized *reinforcement learning algorithms* are inapplicable in resource-constrained wireless sensor networks since they assume a complete knowledge of the network's topology. If not impossible, in most cases this will be tremendously expensive to obtain. On the other hand, there already exist solutions for distributed reinforcement learning - Distributed Q-learning is a good example. MDQL is the solution for problems of multi-objective incentives in distributed manner [17].

V. Conclusion

This paper surveys different reasoning approaches and discusses how they can be applied to SymbioNets. Four different decision approaches are discussed. (i) *Mathematical approaches* require a low computational overhead, but are not well-suited to model complex cross-network and cross-layer influences. (ii) *Network simulators* can be used to determine optimal network settings by simulating a large number of network configurations. However, simulations have a large computational overhead and require perfect network knowledge at a central location. (iii) *Game theory* is well-suited for distributed negotiation implementations. However, each device needs an individual, custom designed cost and utility function. Finally, (iv) *machine learning approaches* do not require any knowledge about the innerworking of the network protocols. However, they can take a long time to reach a steady-state optimal network situation. As such, it is clear that no 'best' cognitive decision approach exists. Instead, the choice of approach depends on which SymbioNets criteria is deemed most important: *complexity*, *support for heterogeneity*, able to deal with *"'malicious"' behavior* and/or *support for distributed solutions*. Further work will focus on how to combine these different approaches to overcome their individual disadvantages.

Acknowledgment

This research is funded by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) through the IWT SymbioNets project and by the Interdisciplinary Institute for Broadband Technology through the QoCON project.

References

[1] R. W. Thomas, L. A. DaSilva and A. B. MacKenzie, "Cognitive networks", Proc. IEEE DySPAN 2005, pp.352-60

[2] M. A. L. Thathachar and O. S. Sastry, "Networks of Learning Automata", Kluwer, 2004

[3] R. W. Thomas, D. H. Friend, L. A. DaSilva and A. B. MacKenzie, "Cognitive networks: Adaptation and Learning to Achieve End-to-End Performance Objectives", IEEE Communications magazine, Dec. 2006

[4] E. De Poorter, P. Becue, M. Rovcanin, I. Moerman and P. Demeester, "A negotiation-based networking methodology to enable cooperation across heterogeneous co-located networks", Ad Hoc Networks, Available online 8 December 2011, ISSN 1570-8705, 10.1016/j.adhoc.2011.11.007.

[5] F. Kelly, "Mathematical modelling of the Internet", Mathematics Unlimited - 2001 and Beyond, Springer-Verlag, Berlin, 2001. 685-702.

[6] D. Plets, W. Joseph, K. Vanhecke, E. Tanghe, L. Martens, Coverage Prediction and Optimization Algorithms for Indoor Environments, EURASIP Journal on Wireless Communications and Networking Special Issue on,Radio Propagation, Channel Modeling, and Wireless,Channel Simulation Tools for Heterogeneous, Networking Evaluation, accepted 2011.

[7] Jiahua Wu, "A Survey of Game Theory in Wireless Networking Design"

[8] T. G. Dietterich , Learning and reasoning, May 26, 2003

[9] A. Forster, "Machine Learning Techniques Applied to Wireless Ad-Hoc Networks: Guide and Survey", 3rd International Conference on Intelligent Sensors Sensor Networks and Information (ISSNIP), Melbourne, Australia, 3-6 Dec. 2007.

[10] T. G. Dietterich, and O. Langley, (2007) Machine Learning for Cognitive Networks:Technology Assessment and Research Challenges in Cognitive Networks: Towards Self Aware Networks, John Wiley and Sons, Ltd, Chichester,UK. doi: 10.1002/9780470515143.ch5

[11] L. P. Kaelbligh, M. L. Littman, A. W.Moore, "Reinforcement learning: A Survey", Journal of Artificial Intelligence Research 4 (1996) 237-285

[12] C. Watkins, "Learning from delayed rewards,"Ph.D disseration, Cambridge university, Cambridge, England, 1989

[13] J. A. Boyan and M. Litman, "Packet routing in dynamically changing networks: A reinforcement learning approach," *Advances in Neural Information Processing Systems*, vol.6, 1994.

[14] M. Lagoudakis and R. Parr. Model-free least-squares policy iteration. In Proc. of NIPS, 2001.

[15] P. Wang, T. Wang, "Adaptive Routing for Sensor Networks using Reinforcement Learning", The Sixth IEEE International Conference on Computer and Information Technology, 2006. CIT '06.

[16] J. Dowling, E. Curran, R. Cunningham and W. Cahill, "Using feedback in collaborative reinforcement learning to adaptively optimize manet routing", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 35, no. 3, pp. 360-372, 2005.

[17] C. E. Mariano, E. Morales, "A New Distributed Reinforcement Learning Algorithm for Multiple Objective Optimization Problems"

Solving Hidden Terminal Problem in Cognitive Networks Using Cloud Technologies

Yenumula B. Reddy
Grambling State University, USA
ybreddy@gram.edu

Abstract—**Hidden terminal problem is one of the well-known problems when a node is visible from the access point but cannot communicate through it. The problem occurs in ad hoc networks as well as cognitive networks. The clear solution is not discussed in cognitive networks. In this paper, we proposed a method to solve the hidden terminal problem through cloud computing. The idea is that the cloud can store the status of cognitive network, compute, reorganize, and make available the current state of cognitive networks for future decisions. Further, we discussed the role of cloud computing in hidden terminal problems and a solution using the blackboard technology. The simulations were presented to cognitive radio network cloud and discussed the packet transfer and handshaking. The proposed cognitive network model using cloud technologies eliminates the overheads of request to send and clear to send. Further, implementation of cognitive networks through cloud technologies minimizes the problem of sudden entry of primary user.**

Keywords: Hidden Terminal; Cognitive Networks; Cloud computing; spectrum mobility; spectrum holes.

I. Introduction

The software defined radio and cognitive radio brought significant attention in telecommunications technology [1, 2]. The cognitive radio has the capability of changing its parameters depending upon the environment that it operates. Therefore, the cognitive capability and reconfigurability makes the cognitive radio (CR) detects the licensed user and identify the unused spectrum (spectrum holes or white spectrum). In recent years, the underutilization of licensed spectrum and introduction of digital broadcasting TV, the FCC (federal communications commission) regulations allowed access to TV white spaces [3]. The recent FCC regulations encouraged international organizations to define cognitive radio standards on TV white spaces.

The CR is a promising technique to detect and utilize the spectrum holes (unused spectrum) efficiently [4]. The process includes spectrum sensing, spectrum management, spectrum sharing, and spectrum mobility. In spectrum sensing, the CR must detect primary (licensed) user through sensing methods including matched filter detection, energy detection, and feature detection. The spectrum management includes characterization, selection, and reconfiguration of the spectrum. The management further includes channel selection, modulation selection, bandwidth setting, observation time setting, transmission time setting, and power setting. The spectrum sharing helps in preventing spectrum overlapping when multiple cognitive radios are involved in intranet. The spectrum mobility deals with the handoff (transfer of connection to another unused spectrum band) in CR networks. The handoff involves the connection lost due to mobility and the quality of service. Further, the hidden terminal problem (HTP) happens in spectrum mobility. The HTP can be solved using the cloud with cognitive radio networks.

A cognitive radio network (CRN) is a group of cognitive radios connected to verify the concepts, algorithms, and protocols. The CRN improves the performance of spectrum utilization and helps smooth the transmission of packets. The tools for total functionality require a significant amount of computing resources in a real time environment. Cloud computing is an alternative to minimize the computing resources and improve the performance. The cloud works like another intelligent agent equipped with all CRN states, policies, and concurrent actions.

In wireless communications, CRN uses knowledge representation, machine learning, genetic algorithms, and game theory techniques for efficient resource (spectrum) utilization. The CRN detects the current network conditions, plans the successive actions, and act on those decisions. Further, the CRN system learns from the history of decisions, actions, and uses the knowledge to improve the decisions. Since CRN uses the previous knowledge and improves the current decisions, game models are very useful in the prediction of network conditions.

The cognitive radio mostly uses the first two OSI model layers (Open Systems Interconnection), whereas the CRN covers all layers. The basic functions learning, reasoning and adapting to current network conditions are required to create end-to-end optimum performance. The CRN makes the decisions to meet the network needs as a whole rather than individual network component. The CRN improves the robustness, usability, security, and human interaction for operation and communication.

The remaining paper discusses the hidden terminal problem and cognitive networks, the role of the cloud to solve the hidden terminal problem, recent developments, problem formulation, simulations, conclusions, and future research.

II. Hidden Terminal Problem and Cognitive Networks

The hidden terminal problem occurs when a node is visible from the access point but cannot communicate to the nodes within communication distance. A star network is the best example. One of the well-known hidden node problems in wireless networks is a three node problem as shown in Figure 1. For example, a user tries to transmit the information from node A to node C. The node A cannot hear the transmission from node C by sensing the medium. Similarly, node C cannot sense the transmission from node A; therefore, they try to communicate and collide at node B. The third example is provided through Figure 2 (waiting forever). The node A is communicating to node D through node B and node C (Figure 2). In the beginning, the node A assumes that channel 2 is free. Once the packet reaches node B it understood that node C is in communication with node D for channel 2. That is node C sends the request for communication (RTS: request to send) to node B for channel 2. The node B is waiting to get the signal clear to send (CTS) from node C for channel 2. Both nodes are waiting for channel 2 at node B and cannot move further. This is a waiting forever problem. The RTS/CTS mechanisms implemented in the MAC (medium access control) protocol helps to eliminate the hidden terminal problem with time overhead.

The RTS/CTS problem further introduces the exposed terminal problem (Figure 1) in wireless communications. For example, if node B is transmitting to node A and node C tries to transmit to node D. The node C and node B are in the transmission range of each other. Therefore, after carrier sense, transmission of node C interferes with the transmission of node B. Further, node D can still receive the transmission of node C without interference, if the nodes C and B are synchronized with the same packet size and data rate (IEEE 802.11 RTS/CTS) [8]. If the nodes are not synchronized, the neighboring node may not hear the CTS during transmission and exposed terminal problem may occur.

The unsuccessful transmissions are proportional to number of hidden or exposed terminals from the transmitting terminal. The RTS/CTS implemented in IEEE 802.11 solves the problem with the conditions explained above. A similar problem occurs if the presence of the primary signal cannot be detected by cognitive radios due to interference or noise of neighboring cognitive radios (hidden node). The RTS/CTS may not help in such cases. The cognitive radios can solve the problem in a cooperative spectrum sensing. Cognitive radios sense the spectrum holes at each hop and utilize the opportunity of allocating the channels in an optimum manner. In a multi-hop cognitive network (MUCON), users can use different frequencies depending upon the availability of spectrum. In MUCON, a common control channel is used to control the available channels. The number of channels available at each hop depends upon the presence of the primary user (PU), because the complete path of the primary user is dedicated. Therefore, the number of channels available to the secondary user or cognitive user (SU/CU) is fixed. In the current problem, the number of channels available is variable at each hop due to collaborative activity of the cognitive network users. The unshaded part of nodes A, B, C, and D in Figure 2 shows the available channels.

The spectrum sensing is used to detect the presence of the PU in the frequency band. The unused spectrum (spectrum holes) detected by the CU helps the control channel to allocate the needed spectrum to CU on the communication path. Further, the energy detectors are used to detect the presence of PU and determine the current status of PU in the geographical area.

The design of cognitive radio (CR) in spectrum sensing poses more challenges since it requires sensing the spectral environment and flexibility to adapt transmission parameters. The design of CR must detect the weak signals and strong signals. The possible solutions include notch filtering, banks on radio frequency (RF) chip, and spatial filtering RF beam forming through adaptive antenna arrays.

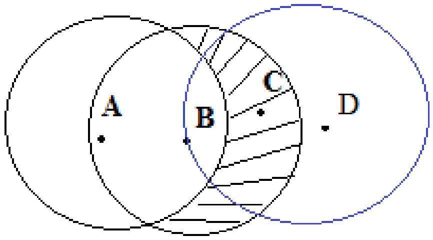


Figure 1. Hidden and exposed node problem

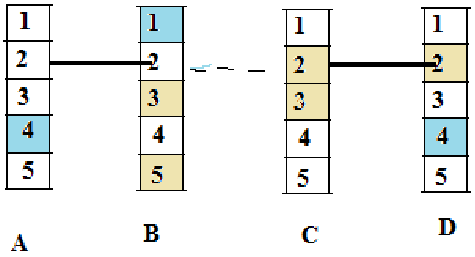


Figure 2. Hidden Node Problem

III. Role of Cloud to Solve Hidden Terminal Problem

The current challenges in cognitive radio networks are storing vast amount of information, processing for real time decisions, and handoff situation during the transfer of information packets at various CRN nodes. Storing and processing the information requires a vast amount of computational capabilities. Further, knowing the current state of channels of the nodes in advance makes the transmission of information easy. But, the change in node status (availability of channels) during the transfer of information requires additional capability. The IEEE 802.11 RTS/CTS solves the hidden terminal problem to a certain extent in wireless communications, but the same cannot be applied in cognitive networks due to the sudden entrance of the primary user. The cloud is very useful (come out of the current bottlenecks) to minimize the storage, save the status of channels and complex computations at node level. Since the channel status at each node is stored in the cloud, the HTP problems will never arise.

The current computing facilities have limited access to resources, financial limitations to purchase packages, problems of portability, and maintenance. To overcome these limitations an emerging concept called cloud computing entered into the technological market. The cloud concept simplified the user needs by eliminating the purchase and maintenance of systems and packages. The advantages are enormous, and security is a basic issue. Paying for the resources you use is the fundamental advantage. This eliminates the purchase of special equipment, packages, other related software, and maintenance. The user can store all the information and use it anywhere in the world without carrying the work related resources. Since the resources can be stored, maintained, executed, and made available on the real-time basis, the concept extended to cognitive radio networks to use spectrum efficiently without any obstructions. Further, the cloud concept eliminates the hidden terminal problem due to the storage of all CRN information at one place and the availability of the current status to all nodes.

CR (portable or non-portable) functions require a cloud due to the intensive computations and vast storage. The functions include signal detection and environment awareness; large amount of data storage, processing and sharing; learning; collect data from various CR nodes and make available to all CR nodes. To facilitate these functions a prototype needs to be built. The prototype must have capabilities of collecting, processing in real time and interacting with nodes for all updates at a central place called cloud as shown in Figure 3. The CRN nodes are connected with cloud to store, plan, update status, and transfer information from a source node to the destination node. This process helps to eliminate the RTS/CTS process at nodes. Further, it saves lot of battery, processing time and hand shaking (or handoff) for mobile devices. The overhearing will be avoided in wireless networks and eliminates the hidden node problem.

Cloud computing avoids the common control channel (CCC) in CRN. The need of a dedicated CCC problem in CRN will be avoided by connecting the nodes of CRN to cloud and allow cloud to take care of the activities of CRN including the HTP. In cloud connection, we do not need to use the time intervals for the available channels. The available channels and frequencies are registered in the cloud and cloud activities are dedicated to transferring the user information to the destination. Therefore, no special protocols or controls are required.

The CCC was discussed in [5] using an example with cognitive nodes and unawareness of the channel set on successive node. In Figure 2, A, B, C, and D are four cognitive nodes in the network. The shaded parts are the channels that occupied. The initial handshake from node A to node B communicated through the control channel is channel 2. Further, node A is unaware of the available channels in node B and vice versa. If the user from node A wants to transmit the information to user (s) at node B, the nodes A and B should negotiate with their channel set with the control structures called RTS/CTS to reserve the channel for communication (we are not discussing a dedicated channel). This problem of negotiating will be eliminated using cloud, since the available channels are registered in the cloud. Therefore, most of the communications and waiting for CTS signal will be eliminated using cloud. Further, the best channel will be selected and quality of service maintained using cloud.

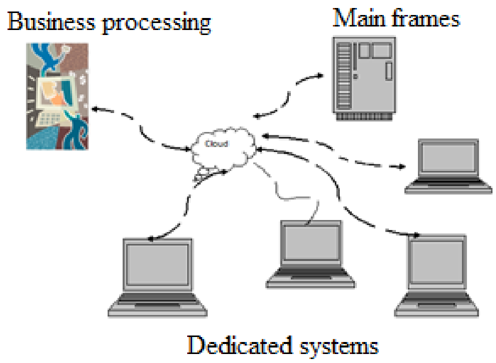


Figure 3. CRN Test-bed and Cloud

IV. REVIEW OF RECENT DEVELOPMENTS

The authors in recent literature [6 - 10] discussed the HTP in ad hoc networks. The literature includes the hidden vs. exposed terminals, solutions to HTP, minimizing the effect of deafness and HTP in wireless and ad hoc networks, and the role of RTS/CTS in IEEE 802.11 to eliminate HTP. The concepts were extended to CRN to solve HTP in CRN [5, 11]. The cloud concept helps CRN to allocate the resources efficiently at node level compared to traditional tools [12-14]. The solution to HTP is the extended opportunity of cloud.

In CRN primary user detection and transferring the packets on multi-hop cognitive radio network (MCRN) are known problems. Further, HTP is another problem that was solved using RTS/CTS in IEEE 802.11 protocol. The RTS/CTS implementations take time and computational resources. The authors in [5] introduced an alternative MAC protocol on MCRN to avoid common control channel through simulations using NS2 package. The idea is good but consumes more computational resources and time. The HTP with the mathematical model and simulations through NS2 was discussed in [9]. Biswas et al. [15] used cooperative spectrum sensing in dynamic CRN to detect primary user and collect the spectrum holes for efficient use of unused spectrum. Zang et al. [16] proposed a fast spectrum sensing algorithm with minimum cognitive radios to perform the cooperative spectrum sensing with minimum errors.

Jayasuria et al. [7] discussed the hidden vs. exposed terminals and concluded that RTS/CTS degrade the performance. The MAC protocol for multi-hop networks was studied in [17] by proposing pair-wise ID (identity detection) countdown. The hidden terminal problem using directional antennas was studied in [10]. They claimed the performance does not depend on network topology and network pattern. Further, the experimental study of Hidden terminal problem was discussed in [6, 18]. In [6], hidden terminal jamming problem presented in IEEE 802.11 for ad hoc networks was discussed. They claimed that signal differential 2dB is sufficient for the stronger transmission and effectively jam a weaker signal. In [18], the impact of hidden nodes in both infrastructure and multi-hop an ad hoc network was studied. The authors claim that RTS/CTS degrade the throughput and may not solve the hidden node problem.

The cognitive networks and using game models for resource allocation was discussed in [19-23]. The role of cognitive radios, the cognitive networks, and standardization of large scale wireless networks were the main theme of these papers. The cloud did not have any role of these networks. The recent developments and timely papers in [12 - 14] open the doors for new research using cloud for CRN. Further security is a big issue in implementing CRN in cloud. The channel allocation in CRN, status of cognitive nodes, and hand shaking in moving devices will be solved easily with cloud.

V. Solution to Hidden Terminal Through Cloud

The CRN cloud (CRNC) is a centralized global data structure consisting of a set of knowledge sources (CRN nodes) called intelligent agents. These agents are self sufficient to detect the primary signal and spectrum holes and store the current status on the cloud. Further, the nodes continuously update their information on the CRNC. The design is similar to the blackboard technology to maintain the current state of nodes.

Figure 3 shows that each node in the CRN is connected to the cloud (CRNC). Let us discuss the transfer of packets from node A to node D in the Figure 2. Each node in the CRN is connected to the cloud and the nodes update their status at cloud. The cloud software executes the status of free channels in each node connected from node A to node D and secures the free channels. The cloud software decides the channel in each connected node to transfer the packets. Further, if the primary user enters at any time, the cloud decides the alternate action of assigning the channel or stores the packets in its buffer till it finds the free channel to transfer the packets to the destination. The same facility will not be possible without cloud organization. For example, if the primary user enters (to use the channel) suddenly, the current data transfer on the channel used by secondary user must be stopped and wait for alternate channel. In CRNC, the free channels are available at each node in the cloud knowledgebase. The cloud software connects the free channel to transfer the data without delay. Therefore, delay time is less in CRNC since it maintains the current channel states and does not require RTS/CTS. Further, if a node cannot connect or is disconnected, the status is stored at cloud and the system administrator is notified. In CRNC if any node could not be connected or disconnected the information will never be lost. The information is forwarded through an alternate path or stored till the node is available (connected).

Each node in CRNC contributes towards a solution without knowing the status of other nodes in the network. The solution model contains the control structure, nodes involved in the current solution, CRNC knowledgebase, query processor, and database access as shown in Figure 4. The query processor takes the request from the CRNC node, verifies the status of each node in the path (source to destination) from database, uses the facts and rules from knowledgebase, and selects best possible path. This process eliminates the RTS/CTS at each node as well as hidden node problem. Therefore, the CRNC process saves time since we eliminate the RTS/CTS and noise interruption (hidden terminal problem). In CRNC, the decision is taken at a global level (in the cloud) and eliminates the problems at each node including processing time, waiting time for clearance of path, interference of nodes in communication distance, and request for a channel to send packets to successive nodes. Two cases arise in the current situation. The case 1 deals with mobile devices and Case 2 with fixed devices.

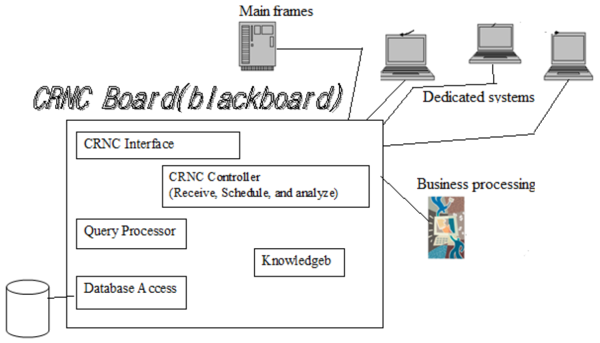


Figure 4. CRNC Architecture

Case 1 deals with mobile devices communications. Once the device opens the communication, the device is registered in the cloud along with the destination device. As the source and destination devices change their current status, the CRNC changes its connection status on real time, since handshaking is needed at any time to any one or both the devices.

Case 2 deals with stable devices. Once the device enters in contact or is ready to communicate, it enters the requests in cloud with a source and destination address. The cloud triggers the respective nodes to update their current status.

One solution is that the information will be sent to cloud from the source node and a destination node copies from cloud. Therefore, we can eliminate even transfer of packets through each node, since the packets are in the cloud and destination address is known. The loss of packets will be minimized when connecting through cloud. Further, the destination will be triggered as soon as the packets are ready to send from the source. The destination node copies from the cloud. This is possible because, any change in the cloud database (blackboard) is triggered to all related CRNC nodes. Therefore, the hidden terminal problem, overhearing, and processing will be eliminated automatically. The purpose will be served if we can achieve the control strategy and real time performance.

The second solution is that the connections required to transfer the information will be calculated and assigned through the cloud database. Once the channels are assigned at each node, the node status will be updated at cloud and the packets will be transferred. As soon as the packets are transferred from each node the channels will be freed and cloud status will be updated.

VI. SIMULATIONS

CRNC scheme eliminates the following requirements which are needed in CRN [5].

- We do not need maintain two radios to be equipped with each node for control signals and receiving/transferring.
- If a new node enters in the network, it registers in cloud only. It does not notify its neighbors.
- If a primary user enters at any node or node status changes the node need not inform to its neighbors.
- The node need not send the communication to its neighbor about its intension to communicate.

Using the above assumptions, sample simulations were completed using MATLAB language. Initially, we assumed five nodes connected in CRNC. The statuses of all nodes are available in the cloud for process. The channels are assigned randomly (0 means busy). Two cases were discussed.

In case 1, if a user in node A wanted to transfer the information to a user in node D, the node status will be verified and updated on the CRNC board. Once the request comes to cloud from node A to transfer packets to the destination node D, the cloud controller triggers node A and node D and lock the needed channels to transfer the information. The channel update (at each node) will be done after completion of the task. Further, if we want to transfer the packets through each connected node in the path, we need to follow a different procedure.

The available channels in all nodes in the path will be verified and assigned the needed channel (s) in each node and mark them as busy. In this case, the first available channels are 2 in node A, 1 in node B, 1 in node C, 1 in node D, and 1 in node E. These channels will be locked and information packets will be transmitted. The channels will be freed after the completion of the task.

In the case 2, the user in node A is making conversation with the user in node E. But A is moving towards B. The CRNC knows that user in node A is close to node B and handshaking is required. The CRNC scheduler searches for a free channel in B. As soon as the user enters in node B's boundary the handshaking will take place and channel in node B will be assigned. The channels will be free after completion of the task.

The CRNC design eliminates the RTS/CTS problem as well as hidden node problem because the status of nodes and allocation of channels was done at CRNC board. Since the decisions were taken at CRNC, the overhearing and packet loss were eliminated automatically. Further, processing in each node, waiting time for allocation of channels, and reservation of channels (RTS/CTS) and other overheads will be eliminated with the implementation of CRNC.

MATLAB OUTPUT

```
Case 1: Fixed terminals
send packet A to D
cloud status
trigger A and D
Available Channels in node A        0   2   3   0   5
Available Channels in node B        1   2   0   4   5
Available Channels in node C        1   2   3   0   0
Available Channels in node D        1   2   3   0   0
Available Channels in node E        1   2   0   4   0
Assigned Channel in node D (first available channel)   1
Channel in node D copies and exits

Case 2: Moving Terminals
Channel in node A moving and calling through the channel in
node E
The updated CRNC cloud status
Available Channels in node A        0   2   3   0   5
Available Channels in node B        1   2   0   4   5
Available Channels in node C         1   2   3   0   0
Available Channels in node D         1   2   3   0   0
Available Channels in node E         1   2   0   4   0

assigned Channel in node A   i =    2
assigned Channel in node D   i =    1
Hand shaking channel in A    j =    2
Hand shaking with channel in B          k =    1

Task completed
cloud status
Available Channels in node A        0   2   3   0   5
Available Channels in node B        1   2   0   4   5
Available Channels in node C         1   2   3   0   0
Available Channels in node D         1   2   3   0   0
Available Channels in node E         1   2   0   4   0
```

VII. CONCLUSIONS AND FUTURE RESEARCH

In the proposed CRNC structure, the interface is connected to CRN nodes and CRNC board (blackboard). The controller receives messages from CRN nodes, schedules messages, and conducts appropriate actions. The knowledgebase consists of a set of production rules and inference engine to operate those rules. The scratch space (work space) stores the current state of messages for processing.

The hidden node problem and dynamic spectrum allocation are very important in cognitive radio networks. The hidden node problem will be eliminated by using cloud, since CRNC board has current status of CRN nodes. Further, the changes will be triggered automatically using the blackboard structure. The structure of CRNC provides the use of two mobile devices, one mobile device and one fixed device, both fixed devices and/or multiple devices with fixed and mobile devices.

The cost factor related to computing and communications, storage is not provided in this paper.

These are application dependent. Further, power optimization at node level depends upon the quality of services and time varying state of wireless communications [24, 25]. The cloud helps better power savings [25].

The future work involves the security issues for CRNC, and possible solutions.

ACKNOWLEDGEMENT

The research work was supported by the Minority Leaders Program through Grant number GRAM 11-567-02-C2. The author wishes to express appreciation to Dr. Connie Walton, Provost and Vice President, Academic Affairs, Grambling State University for her continuous support.

REFERENCES

[1] Mitola, J. and Maguire, G. Q., “Cognitive Radio: Making software radios more personal”, IEEE personal Communications, vol. 6, no. 4, pp. 13-18, 1999.

[2] Mitola, J., "Cognitive Radio – An Integrated Agent Architecture for Software Defined Radio", Ph.D. Dissertation, Royal Institute of Technology, Kista, Sweden, May 8, 2000

[3] FCC, Second report and order and memorandum opinion and order, FCC 08-260, 2008 (last access July 1012).

[4] Akyildiz, I. Lee, W. and Chowdhury K., “CRAHNs: Cognitive Radio ad hoc Networks”, Ad Hoc Networks, vol. 7, pp. 810-836, 2009.

[5] Kondareddy, Y. R. and Agrawal, P., “Synchronized MAC Protocol for Multi-hop Cognitive Radio networks”, IEEE International Conference on Communications (ICC’08), pp. 3198 – 3202, 2008.

[6] Ware, C. Wysocki, T. and Chicharo, J., “On the Hidden Terminal jamming Problem in IEEE 802.11 Mobile Ad hoc Networks”, IEEE International Conference on Communications (ICC), Helsinki, Finland, 2001, pp. 261-265.

[7] Jayasuriya, A. Perreau, S. Dadej, A. and Gordon, S., “Hidden Vs. Exposed Terminal Problem in Ad hoc Networks”, Proceedings of the Australian Telecommunications, Networks and Architecture Conference (ATNAC 2004), December 2004, Sydney, Australia.

[8] Fullmer, C.L. and Garcia-Luna-Aceves, J.J., “Solutions to Hidden Terminal Problems in Wireless Networks”, Proceedings of the SIGCOMM’97 conference on Applications, technologies, architectures, and protocols for computer communication, 1997, pp. 39-49.

[9] Jeong, J. Kim, H. Lee, S. and Shin, J., “An Analysis of Hidden Node Problem in IEEE 802.11 Multihop Networks”, Sixth International Conference on Networked Computing and Advanced Information Management (NCM), Seoul, S. Korea, 2010, pp. 282-285.

[10] Gossain, M. P. Cordeiro, C. and Agrawal, D. P., “Minimizing the Effect of Deafness and Hidden Terminal Problem in Wireless ad hoc Networks using Derectional Antennas”, Wireless Communications & Mobile Computing - Wireless Ad Hoc Networks: Technologies and Challenges, vol. 6, Issue 7, November 2006, John Wiley and Sons Ltd. Chichester, UK, pp. 917-931.

[11] Reddy, Y. B., “Spectrum Detection in Cognitive Networks by Minimizing Hidden Terminal Problem”, ITNG 2012, April 16-8, Las Vegas.

[12] Ge, F. Lin, H. Khajeh, A. Chang,C. J. and Eltawil, A.M., “Cognitive Radio Rides on the Cloud”, Military Communications Conference, MILCOM 2010, October 2010, 1448 – 1453

[13] Chen, Z. Zang, C. Lin, F., Yu, J. and Lie, X., “Towards a Large-Scale Cognitive Radio Network: Testbed, Intensive Computing, Frequency Agility, and Security”, International Conference on Computing, Networking and Communications (ICNC), 2012, Feb. 2 2012, pp. 556 - 562

[14] Ko, C. Hang, D. H. and Wu., S., “Cooperative Spectrum Sensing in TV White Spaces: When Cognitive Radio Meets Cloud”, IEEE INFOCOM 2011, pp. 683-688.

[15] Biswas, A. R., Aysal, T.C., Kandeepan, S., Kliazovich, D., and Piesiewicz, R., “Cooperative Shared Spetrum Sensing for Dynamic Cognitive Radio Networks”, IEEE International Conference on Communications, Dresdan, Germany, June 2009, pp. 1-5.

[16] Zhang, W. Mallik, R. K. and Letaief, K. B., “Cooperative Spectrum Sensing Optimization in Cognitive Radio Networks”, IEEE International conference on Communications, Beijing, China, 2008, pp. 3411-3415.

[17] You, T. Hassanein, H. and Yeh, C., “PIDC – Towards an Ideal MAC Protocol for Multi-hop Wireless LANs”, International conference on Wireless Networks, Communications and Mobile Computing, June 2005, pp. 13-16.

[18] Ng, P. C. Liew, S. C. Sha, K. A. and To, W. T., “Experimental Study of Hidden-node Problem in IEEE 802.11 Wireless Networks”, ACM SIGCOMM, August 2011, Toronto, Canada.

[19] Thomas, R. W. DaSilva, L. A. and MacKenzie, A. B., "Cognitive networks", Proceedings of the First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, Baltimore, MD, USA, November 8–11, 2005.

[20] Fortuna, C. and Mohorcic, M."Trends in the development of communication networks: Cognitive networks", Computer Networks, 2009.

[21] Song, L., "Cognitive Networks: Standardizing the Large Scale Wireless Systems", 5th IEEE Consumer Communications and Networking Conference (CCNC), 2008, pp. 988-992.

[22] Balamuralidhar, P. and Prasad, R., "A context driven architecture for cognitive nodes", Wireless Personal Communications 45 (2008), pp. 423–434.

[23] Song, L. and Hatzinakos, D. "Cognitive networking of large scale wireless systems". International Journal of Communication Networks and Distributed Systems 2 (4): 2009, pp. 452-475.

[24] Khajeh, A. Cheng, S. Y. Eltawil, A. and Kurdahi, F., “Power management for cognitive radio platforms,” in IEEE Global Telecommunications Conference, GLOBECOM 2007, pp. 4066 -4070.

[25] Amin, K. Kim, M. Dutt, N. Eltawil, A. M. and Kurdahi, F. J., “Cross-Layer Co-Exploration For Mobile Multimedia,” in proceedings of IEEE Workshop on Embedded Systems for Real-time Multimedia (ESTIMedia’08) in part of ESWEEK, pp. 13-18.

Cluster-based Energy-efficient Composite Event Detection for Wireless Sensor Networks

Irfana Memon, Traian Muntean
ERISCS Research Group
Aix-Marseille Université
France

Emails: {irfana.memon, traian.muntean}@univ-amu.fr

***Abstract*—Wireless sensor networks (WSNs), well known communicating architectures today, are often used to detect the occurrence of some environmental events, such as pollution monitoring, forest fires detection, location and tracking, etc. In order to reduce irrelevant alarms, multiple attributes are used in the event detection process. In WSNs, communication is often by far more expensive and difficult to control than local computation within nodes. Therefore, it becomes critical to reduce the amount of data exchange within a WSN, in order to optimize the use of power and energy resources within nodes. Energy optimization is thus one of the most important aspects of the WSN design. There are already literature and projects dealing with the detection of composite events using data aggregation at intermediate nodes. In this paper, a cluster-based energy-efficient composite event detection (CEC) for wireless sensor networks scheme is proposed, which performs local computation at sensor nodes and local data aggregation at level of each cluster heads in order to reduce the communication overhead. Simulation results show that jointly, considering both local computation at sensor nodes level and local data aggregation at intermediate nodes will further reduce the total energy consumption and thus prolong the network lifetime.**

Keywords-Wireless sensor networks (WSNs); data aggregation; local computation; composite event detection.

I. Introduction

A wireless sensor network (WSN) consists of a large number of sensor nodes which are distributed in a given space for measuring environmental parameters, such as temperature, light, sound, humidity, and so on [1]-[3]. Many applications have already been envisioned and described for WSNs in a wide range of areas, such as environment monitoring [4], health care applications [5], military surveillance applications [6], positioning and tracking [7], etc. Depending on the application domain, it may be necessary for sensor nodes within the WSN to react quickly or with critical timing constrains to detected events [8]. Moreover, the data collected by the WSN must be fresh when the corrective action is taken.

One of the critical tasks in designing a WSN is to monitor, detect and report various useful occurrence of events in a timely and reliable fashion. An event can be defined as an exceptional change in the environmental parameters. Events can be simple (atomic) or composite [9]. An atomic event can be detected merely based on the observation of one attribute, for example high temperature, if the temperature is higher than a specified threshold, an atomic event is detected. A composite event is the combination of different atomic events. A detailed description of composite events is given in Section III. An event alarming application needs an answer to a question which can be derived by a set of predicates. For example, in fire alarming applications, users are not interested in knowing the exact reading of attributes (temperature, smoke) of monitored area, but they want an exact and valid answer to the question: is there fire in the monitored area? In this case, we assume that an event has some significant characteristics that can be used as threshold to distinguish between normal and abnormal environment parameter. Event detection sensor networks require periodic data update (fresh data) from the network. Sending data periodically to a remote base station may incurs high communication overhead, and high energy consumption for event-driven applications. One of the key problems in event-driven applications is energy efficient data extraction, (i.e. how can a base station obtain the event report with a low energy consumption). Hill et al. [10] have shown that a sensor node spends approximately the same amount of energy for sending a single bit of data as it does to execute 800 instructions. Thus, in order to decrease energy consumption and thus increase network lifetime, the amount of data exchanged should be minimized.

Data aggregation techniques are very effective in reducing communication overhead (i.e., the data sensed by the sensors are combined at intermediate nodes before sending to a remote base station, BS). A number of data aggregation algorithms have been proposed in the literature [11]-[16]. Cluster-based topologies help to deal with in-network data aggregation (i.e., sensors are grouped in clusters and data aggregation is done locally within clusters). Some of the advantages to be achieved from clustering in WSNs are the reduction in energy for message transmission and constructing a virtual backbone for data routing purpose [17]. Many clustering algorithms have been proposed for WSNs, such as LEACH [18], LRS [19], PEGASIS [20], and HEED [21]. In this paper, we are not investigating clustering algorithms. We are considering that a wireless sensor network is divided into multiple clusters using an existing clustering algorithm. Each cluster has a special node (called cluster head (CH) or aggregator node) that collects data from all other nodes in the cluster and then performs some computations on the collected data. All CHs form the network backbone. Instead of transmitting each packet from sensor nodes to a remote base station, all CHs collect data, aggregate and send a single aggregated data packet to the base station along the network backbone. For intra-cluster communications (the communications between cluster members and the cluster head), an efficient routing mechanism is required. A number of routing protocols are reported in the literature [22]-[26]. Routing within the cluster can be realized through a directed diffusion algorithm.

Several schemes have been proposed for event detection for WSNs. In some studies, events are detected using a fuzzy logic rule-based system [27]-[28]. Several rules constitute a fuzzy rule-based system [29], however, keeping the rule-base might require a significant memory on sensor nodes. The number of rules grows exponentially to the number of variables, such as for n variables each of which can take m values, the number of rules in the rule-base is m^n. Furthermore, use of spatial and temporal features during decision process increases the number of rules. Since sensor nodes have limited memory, keeping a complete rule-base on every node might not be achievable. Moreover, constantly traversing a large rule-base might considerably slow down the event detection.

On the other hand, in several schemes, events are detected using a user defined threshold [30]-[32]. In such schemes, event alarm is arisen when sensor reading is lower or higher than predefined threshold value. However, existing threshold based schemes introduce high communication overhead. More details on related work are provided in Section II.

In this paper, we propose a threshold-based approach for composite event detection in WSNs. The main advantages of this proposed approach is simplicity, energy efficiency, and can be applied more easily. Indeed, the sensor node need to evaluate a simple boolean expression which is perfectly in line with considering requirement of WSNs of keeping computational complexity low. Our proposed Cluster-based Energy efficient Composite event detection (CEC) algorithm is a two-level event detection scheme. At the first level, sensor nodes perform local computation to detect atomic events and transmit a report message to the CH, when an atomic event has been observed. Moreover, the second level is carried out at intermediate nodes (i.e., CHs), which perform local data aggregation of received atomic event reports from the sensors in the cluster and take decision for a composite event. The main contributions of our proposed scheme are:

- (1) All sensor nodes (i.e., CHs and cluster members) send packets in binary format instead of raw data; achieves low overhead on data packet.
- (2) Each cluster member performs local computations on the sensed data, and sends a report message to the cluster head when an atomic event has been observed.
- (3) CHs perform local data aggregation for the received data from their cluster members and evaluate the composite event. Then, the CHs send particular synthesis of composite event occurrence to the base station, when a composite event has been observed.
- (4) Finally to reduce false alarm rate to the base station, composite event definition consists of multiple attributes [31].

Since events are often combination of more than one attribute , use of multiple attributes can help increasing event detection rate and reducing false alarm rate [31]. The event that is a combination of multiple attributes is a composite event. For example, the composite event fire is a combination of multiple attributes, i.e., the occurrence of fire should satisfy some conditions such as temperature $>$ 100 ^{0}C AND smoke $>$ 100mg/L, rather than a simple condition temperature $>$ 100 ^{0}C alone. Thus, it reduces false alarm rate to the base station.

In this paper, optimizing energy saving with joint local computation on sensors and local data aggregation on cluster heads is considered. Simulation results compare the proposed scheme with the non-local computation schemes. It is shown that significant energy saving can be achieved via the proposed scheme. Ongoing work further reviews performances and compare with other existing schemes; this will be presented in a companion paper.

The rest of the paper is organized as follows. In Section II, we describe related works. Section III provides detail of the Cluster-based Energy efficient Composite event detection (CEC) scheme. Section IV presents simulation results, and Section V concludes the paper and discusses future works.

II. Related works

Recently, problems of event detection for WSNs have drawn a lot of attention. Liang and Wang [27] propose to use fuzzy logic with double sliding window for event detection. However, the authors do not study the effect of fuzzy logic approach.

Marin-Perianu and Havinga [28] have proposed fuzzy logic based approach for event detection. In this approach, fuzzy logic is used to combine personal and neighbors' observations for an event detection. However, the approach does not use any temporal semantics and do not analyze the number of false alarms.

Kumar et al. [30] have proposed a framework for detecting both simple and composite events with distributed collaboration of sensor nodes. This framework consists of two protocols called simple event detection and composite event detection. The proposed protocols are based on publish-subscribe communication paradigm. Each protocol works in two phases: (i) initialization phase, and (ii) collection phase. In the initialization phase, an application subscribes events of interest (atomic events or composite events) to the sensor network, and constructs an event based tree (EBT) based on these events. In the collection phase, data in the form of predicate is collected along this tree and sent to the subscriber. The subscriber will decide of the occurrence of events. In order to save energy, the data are collected using aggregation. In this scheme, all the sensor nodes periodically send their data to the subscriber. A conclusion is then made at the subscriber to decide whether a predefined event has happened based on the received data. However, sending data periodically from sensors to the subscriber incurs communication overhead which causes large energy consumption. In addition, at a subscriber, the received data need to be further analyzed to obtain a conclusion which delays the alarm to be timely announced.

Vu et al. [31] examined the Timely Energy-efficient k-watching Event Detection Problem. The objective is to assure that an area is monitored by at least k sensors. The proposed protocol divides the set of sensors into a number of non-disjoint subsets called detection sets such that each detection set ensures the k-watching property. Each detection set defines a tree, constructed using the Breadth First Search (BFS) algorithm starting from a gateway and data is then collected along this tree. Any sensor node with richer energy resource can be gateway. To ensures k-watching property, the protocol maintains a counter 'cl' to records the number of currently required sensors for composite event detection in the actual detection set. In this scheme, all the sensor nodes send their sensed data to a gateway node, and a decision on whether a predefined event occurs based on the received data is made by gateway node. However, sending sensed data from sensors to the gateway node incurs high communication overhead which causes large energy consumption. In addition, an other drawback of this approach is that it requires global information to construct the detection sets.

To overcome the problem of global information in [31], Marta et al. [32] have proposed an energy-efficient composite event detection in wireless sensor network. The algorithm sets sensor's schedule using a localized connected dominating set based approach. In the Initialize phase a sensor node will decide whether it will be active or sleeping in the next round. The algorithm decides the sensor scheduling at the beginning of each round. Each node relies on local information from its h-hop neighborhood. The main drawback of this approach is that each node requires h-hop neighborhood information. For constructing h-hop neighborhood, each node broadcasts h-HELLO messages, thus incurring high communication overhead.

III. Composite Event Detection

A composite event, e.g., fire is a combination of multiple attributes, i.e., the occurrence of fire should satisfy some conditions such as temperature $>$ 100 ^{0}C AND smoke $>$ 100mg/L AND humidity $<$ 50. In particular, this paper focuses on forest fire detection. We implement threshold-based method for forest fire detection in WSNs. In forest fire detection method, sensor nodes collect measurement data such as , temperature, smoke, and relative humidity. All these factors are key factors for determining the forest fire [33]-[34]. The Forest-Fires Surveillance System (FFSS) [33] was developed to prevent forest fires in the South Korean Mountains and to have an early fire-alarm in real time. The system senses environment state such as temperature, humidity, smoke and determines forest-fires. It is quite difficult to maintain threshold values for temperature, smoke, and humidity. High temperature, low humidity are the key factors for forest fire [35]-[38] and burning of wood gives off large amount of smoke. We evaluate our approach by setting threshold value for temperature, smoke and humidity equals to 50, 50, and 20 respectively.

All nodes have sensors, temperature, smoke, and humidity. But due to lack of energy there might be a situation where sensor can fail, e.g., Node 1 has temperature, smoke and humidity sensors. But, node 2 has smoke and humidity sensors because temperature sensor on node 2 has failed. In such case, composite event detection must be occurs with sharing sensing capabilities of sensor nodes. Keeping this in mind, our approach perform composite event detection at cluster head. In this section, we present a Cluster-based Energy efficient Composite event detection (CEC) protocol for WSNs in detail.

A. Notations, Messages structure, and Preliminary definitions

*1) **Notations**:* The following notations are used throughout the paper for different type of messages used in our simulations.

- Thr_Temperature, Thr_Smoke, Thr_Humidity: Thresholds for temperature, smoke, and humidity respectively.
- C_Temperature, C_Smoke, C_Humidity: Current sensed temperature, smoke, and humidity respectively.
- AE: Atomic Event
- CE: Composite Event

*2) **Messages structure**:* A message has several fields. The first field of a message is the type of the message which can be one of the following: AE_Report (Atomic Event Report), and CE_Report (Composite Event Report).

- **AE_Report message has the following format:** {AE_Report, S_ID, (X_{S_ID}, Y_{S_ID}), Dest_ID, (X_{Dest_ID}, Y_{Dest_ID}), Temperature, Smoke, Humidity}
  Where, S_ID field stores the ID of the sensor node

that sends the message (sender node), (X_{S_ID}, Y_{S_ID}) field stores the location of the sender node in the monitored area, Dest_ID field stores the ID of the destination, $(X_{Dest\_ID}, Y_{Dest\_ID})$ field stores the location of the destination, Temperature field holds the report of atomic event Temperature, Smoke field holds the report of atomic event Smoke, and Humidity field holds the report of atomic event Humidity. This message is used by ordinary nodes (cluster members) to give information on the occurrence of an atomic event to the CH in the cluster. The packet size is 132 bytes.

- **CE_Report message has the following format:** {CE_Report, S_ID, (X_{S_ID}, Y_{S_ID}), Dest_ID, $(X_{Dest\_ID}, Y_{Dest\_ID})$, CE}
  Where, S_ID field stores the ID of the sensor node that sends the message (sender node), (X_{S_ID}, Y_{S_ID}) field stores the location of the sender node in the monitored area, Dest_ID field stores the ID of the destination, $(X_{Dest\_ID}, Y_{Dest\_ID})$ field stores the location of the destination, and CE field holds the report of composite event. This message is used by CHs to give notification to the BS that the occurrence of a composite event has been detected. The packet size is 124 bytes.

*3) **Preliminary definitions**:* An atomic event condition and composite event condition are formalized as follows.

Definition 1 Atomic Event detection: An atomic event can be determined based on single attribute. The atomic event detection is carried out by comparing sensed data of attributes with their predefined threshold values. An atomic event condition is a Boolean formula and is denoted by AE which evaluates to TRUE or FALSE.

Definition 2 Composite Event detection: A composite event involves multiple attributes. A composite event condition is also a Boolean formula and is denoted by CE which occurs when all attributes that forms a composite event occurs.

B. Cluster-based Energy efficient Composite event detection (CEC) protocol

Our proposed CEC protocol contains also two phases: (1) Initialization phase and (2) data collection and composite event detection phase.

(1) Initialization phase: In initialization phase, clusters are formed after network deployment using an existing clustering algorithm. Clusters are assumed to have their own cluster head and all CHs collectively form a backbone in the network (shown as Figure 1.). CHs will send their report to the base station along the backbone. For simplicity of our discussion, we assume that each CH knows the topology and other ordinary nodes (cluster members) in each cluster know their CH. Note that the knowledge needed by CHs and cluster members can be obtained when the clustered network is built.

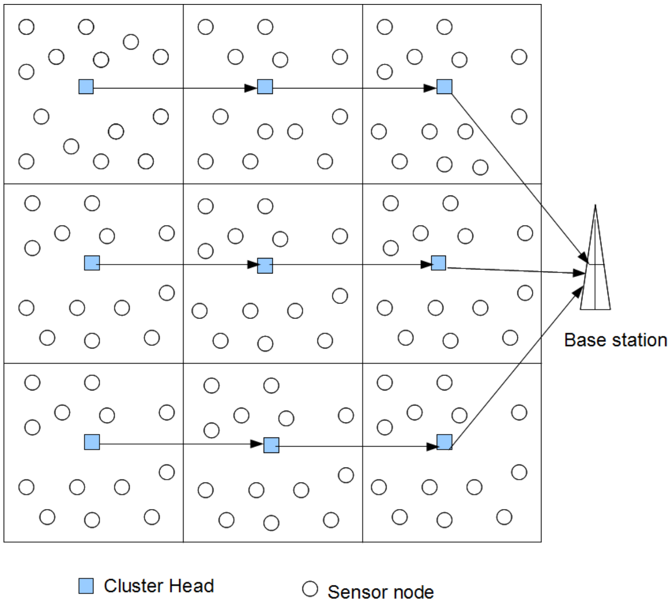


Figure 1. Clustered network

(2) Data collection and Composite Event detection phase: Each node periodically monitors a set of distributed attributes A = { A_1,...,A_n}, and generates a discrete data value vector at every time instance (at every second). Each attribute, A_i, may be an environmental property being sensed by the node (i.e., temperature, smoke, humidity). In this paper, sensor data is generated via a pseudorandom number generator. The node compare the sensed with their corresponding predefined threshold values, and compute atomic event 'AE' using the definition 1. When AE becomes "TRUE" (i.e., AE = 1), a cluster member sends <AE_Report> message to the CH in the cluster. We assume that CHs have timer. Timer is used to ensure that the composite event detection is based on fresh date received from cluster members. On receiving <AE_Report> message, CH performs local data aggregation to estimate composite event occurrence based on the collected data within the "timer" using the definition 2. When CE becomes "TRUE" (i.e., CE = 1), a cluster head sends <CE_Report> message to the BS along the network backbone.

IV. Simulation and Results

A. Simulator

WSNet simulator [39] is used as a simulation platform. WSNet is wireless network event-driven simulator, it has been developed in CITI laboratory of INSA Lyon. It is largely similar to other event-driven simulators such as ns2, JiST, GloMoSim, GTNetS, omnet++ though it differentiates itself with various functionalities, a precise radio medium simulation and the simulated node internals. Node, environment and radio medium blocks are developed in independent dynamic libraries. Moreover, the addition of new models does not require to modify the core of WSNet and can be done easily.

B. Simulation Results

We evaluate our approach composite event detection *with Local Computation* on the cluster members in the cluster through simulation, and we compare it to composite event detection *without Local Computation* on the cluster members in the cluster. The parameters involved in this comparison are number of messages transmitted in the cluster, and amount of the total remained energy in the cluster. To do simulation and evaluation, 100 sensor nodes are located randomly within a cluster of 50*50 m^2. The initial energy of nodes is taken 1 Joule. We consider that each node consumes 0.003 Joules to send a packet. Atomic event attributes are sensed periodically and in each period atomic event detection process is executed. All of the simulations were run 50 times, and the average results are plotted in the graphs. Table 1 lists the simulation parameters.

Table I
SUMMERY OF THE PARAMETERS USED IN THE SIMULATION EXPERIMENTS

Parameter	Value
Simulation time	10s,20s,30s,40s,50s,60s,70s,80s, 90s,100s
Cluster size (m x n)	50 x 50 m^2
Number of Nodes in the cluster	100 nodes
Nodes distribution	Nodes are randomly distributed
Performance parameters	Communication overhead within the cluster, sum of remaining energy in the cluster
Antenna type	Omnidirectional
MAC Layer	802.11
Initial node energy	1 Joule
Energy for transmitting one packet	0.003 Joules
Radio range	30

Figure 1 shows comparison of communication overhead in the cluster by varying simulation time for two cases. *NoLocalComputation* is the case of all existing protocols for composite event detection, in which all ordinary sensors monitor atomic event attributes and send it to the CH periodically. *CEC Scheme* is the case in which local computation for an atomic event detection on the ordinary sensors (cluster members) is applied. When a cluster member detects an atomic event, then it sends a message to the CH. Figure 1 shows that our proposed scheme has low communication overhead in the cluster when compared to the existing scheme with *NoLocalComputation*.
Figure 2 shows comparison of the total remaining energy of cluster member in the cluster by varying simulation time in *CEC Scheme* to *NoLocalComputation* case.

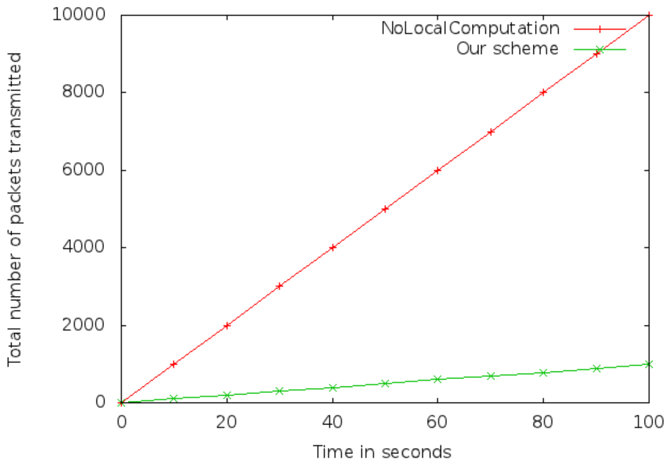


Figure 2. Total number of packets transmitted in one cluster

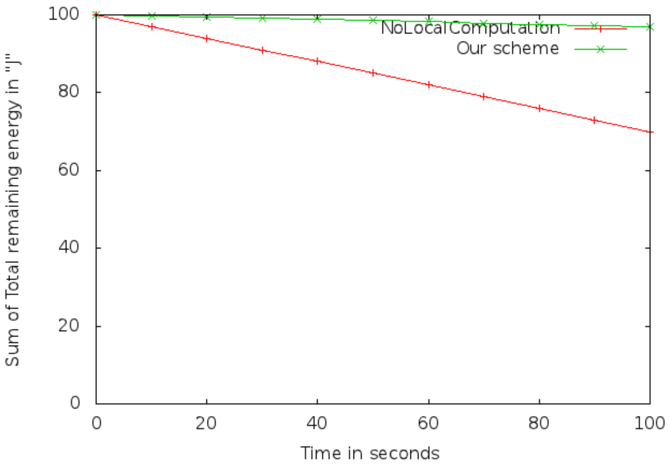


Figure 3. Total Remaining Energy in one cluster

V. CONCLUSION AND FUTURE WORK

As mentioned already, communication consumes high energy when compared to computation in WSNs. In order to reduce energy use within sensor nodes and thus to increase lifetime of wireless sensor networks, composite event detection with local data aggregation has been used in this work. It is supported by local computation in the clusters. Our proposed scheme is proven to be more efficient, since each sensor is required to perform local computation to detect the atomic event and send report to cluster head in the case when atomic event occurs. The total remaining energy in the cluster was also determined, which is considered as the metric to prove energy efficiency with our proposed protocol. This has been conducted by simulation. The simulation results show that using local computation in the cluster, communication overhead is decreased by our scheme and thus more energy is saved than the existing schemes with *NoLocalComputation* which will lead therefore to an increase the network lifetime.

In our ongoing work, we will examine the performance data, accuracy, robustness and reliability. To prevent false alarms, a computationally cheap and efficient filtering system is required for forest fire detection. Keeping into mind the critical resource constraint nature of sensor network, we use Bayesian classifier and Gaussian Mixture Model classifier to filter false or irrelevant event reports. The companion paper provides information about these classifiers and reasons why they are suitable for wireless sensor networks. We will compare our proposed approach with other existing schemes. Furthermore, filtering process has an impact on reporting delays. We will also investigate trade-off between irrelevant alarms tolerance and reporting latency.

References

[1] C. Buratti, A. Conti, D. Dardari, and R. Verdone, “An Overview on Wireless Sensor Networks Technology and Evolution”, Sensors, pp. 6869-6896, [retrieved: July, 2009].

[2] I. F. Akyildiz, W. Su, and E. Cayirci, “Wireless sensor networks: A survey”, Computer Networks, pp. 393-422, [retrieved: December, 2001].

[3] N. Meratnia, B. J. V. D. Zwaag, H. W. V. Dijk, D. J. A. Bijwaard, and P. J. M. Havinga, “Sensor Networks in the Low Lands”, Sensors, pp. 8504-8525, [retrieved: July, 2010].

[4] J. B. Ong, Z. You, J. Mills-Beale, E. L. Tan, B. D. Pereles, and K. G. Ong, “A wireless, passive embedded sensor for real-time monitoring of water content in civil engineering materials”, IEEE Sensors, pp. 2053-2058, [retrieved: November, 2008].

[5] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis, and M. Welsh, “Wireless Sensor Networks for Healthcare”, Proceedings of the IEEE, pp. 1947-1960, [retrieved: November, 2010].

[6] L. Lamont, M. Toulgoat, M. Deziel, and G. Patterson, “Tiered Wireless Sensor Network Architecture for Military Surveillance Applications”, SENSORCOMM, pp. 2053-2058, [retrieved: August, 2011].

[7] Q. Hao, D. J. Brady, B. D. Guenther, J. B. Burchett, M. Shankar, and S. Feller, “Human tracking with wireless distributed pyroelectric sensors”, IEEE Sensors, pp. 1683-1696, [retrieved: December, 2006].

[8] L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, A. Tirumala, Q. Cao, T. He, J. A. Stankovic, T. Abdelzaher, and B. H. Krogh, “Lightweight Detection and Classification for Wireless Sensor Networks in Realistic Environments”, SenSys, pp. 205-217, [retrieved: November, 2005].

[9] S. Li, S. H. Son, and J. A. Stankovic, “Event Detection Services Using Data Service Middleware in Distributed Sensor Networks”, In Proceedings of the 2nd Internnational Workshop on Information Processing in Sensor Networks, pp. 502-517, [retrieved: April, 2003].

[10] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System architecture directions for networked sensors”, In the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 93-104, [retrieved: November, 2000].

[11] K. Maraiya, K. Kant, and N. Gupta, “Wireless Sensor Network: A Review on Data Aggregation”, International Journal of Scientific and Engineering Research, [retrieved: April, 2011].

[12] S. Madden, R. Szewczyk, M. Franklin, and D. Culler, “Supporting aggregate queries over ad-hoc sensor networks”, In Workshop on Mobile Computing and Systems Applications (WMCSA), pp. 49-58, [retrieved: August, 2002].

[13] D. Wagner, “Resilient aggregation in sensor networks”, In Proceedings of the 2nd ACM workshop on Security of adhoc and sensor networks, pp. 78-87, [retrieved: October, 2004].

[14] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, “Impact of Network Density on Data aggregation in Wireless Sensor Networks”, In Proceedings of the 22nd International Conference on Distributed Computing Systems, pp. 575-578, [retrieved: November, 2002].

[15] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks”, In Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom’2000), pp. 56-67, [retrieved: August, 2000].

[16] S. Madden, M. J. Franklin, and J. M. Hellerstein, “TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks”, In Proceedings of the Fifth Symposium on Operating Systems Design and implementation, [retrieved: December, 2002].

[17] O. Dagdeviren and K. Erciyes, “A Distributed Backbone Formation Algorithm for Mobile Ad hoc Networks”, Lecture Notes in Computer Science, pp. 219-230, [retrieved: December, 2006].

[18] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocols for Wireless Microsensor Networks”, In Proceedings of the 33rd Hawaii International Conference on System Sciences, [retrieved: January, 2000].

[19] S. Lindsey, C. S. Raghavendra, and K. Sivalingam, “Data Gathering in Sensor Networks using the Energy Delay Metric”, In Proceedings of the 15th International Parallel and Distributed Processing Symposium, pp. 2001-2008, [retrieved: April, 2001].

[20] S. Lindsey and C. S. Raghavenda, “PEGASIS: power efficient gathering in sensor information systems”, In Proceedings of the IEEE Aerospace Conference on IEEE Aerospace and Electronic Systems Society, pp. 1125-1130, [retrieved: March, 2002].

[21] O. Younis and S. Fahmy, “HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks”, IEEE On Transactions on Mobile Computing, pp. 366-379, [retrieved: October, 2004].

[22] T. Muntean, J. Garmendia, and S. Rivas, “A model for routing in networks of mobile agents Parallel and Distributed Computing Systems”, In Proceedings of the Parallel and Distributed Computing and Systems, [retrieved: November, 2000].

[23] J. Garmendia, S. Rivas, L. Mugwaneza, and T. Muntean, “Pi-Routage: une technique d’acheminement des communications pour processus mobiles”, In Proceedings of the RENPAR’12, pp. 179-184, [retrieved: June, 2000].

[24] T. Muntean, “Diffusing Mobile Processes”, in Concurrent Information Processing and Computing, NATO Science Series III: Computer and System Sciences, pp. 111-130, [retrieved: May, 2005].

[25] V. Moraru and T. Muntean, “A Model for Secure Broadcasting Mobile Systems”, ICTEI’08 - Chisinau, [retrieved: May, 2008].

[26] K. Atighehchi, T. Muntean, S. Parlanti, R. Rolland, and L. Vallet, “A cryptographic keys transfer protocol for secure communicating systems”, In Proceedings of the 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, pp. 339-343, [retrieved: September, 2010].

[27] Q. Liang and L. Wang, “Event detection in wireless sensor networks using fuzzy logic system”, In Computational Intelligence for Homeland Security and Personal Safety, Institute of Electrical and Electronics Engineers (IEEE), pp. 52-55, [retrieved: August, 2005].

[28] M. Marin-Perianu and P. Havinga, “D-FLER: a distributed fuzzy logic engine for rule-based wireless sensor networks”, In Proceedings of the 4th International Conference on Ubiquitous Computing Systems, LNCS, pp. 86-101, [retrieved: August, 2007].

[29] L. A. Zadeh, “Outline of a new approach to the analysis of complex systems and decision processes”, IEEE Transations on Systems Man and Cybernetics, pp. 28-44, [retrieved: January, 1973].

[30] A. V. U. P. Kumar, A. M. Reddy, and D. janakiram, “Distributed collaboration for event detection in wireless sensor networks”, In Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing, pp. 1-8, [retrieved: November, 2005].

[31] C. T. Vu, R. A. Beyah, and Y. Li, “Composite event detection in wireless sensor network”, In Proceedings of the IEEE International Performance, Computing, and Communications Conference, pp. 264-271, [retrieved: April, 2007].

[32] M. Marta, Y. Yang, and M. Cardei, “Energy-Efficient Composite Event Detection in Wireless Sensor Networks”, In Proceedings of the 4th International Conference on Wireless Algorithms, Systems, and Applications, Lecture Notes in Computer Science, pp. 94-103, [retrieved: August, 2009].

[33] B. Son, Y. Her, and J. Kim, “A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea mountains”, International Journal of Computer Science and Network Security (IJCSNS), pp. 124-130, [retrieved: September, 2006].

[34] http://www.firescience.gov/projects/briefs/03-1-3-02_FSBrief41.pdf [retrieved: February, 2009].

[35] N. Corsi and A. Gemelli, “An Innovative Approach to Forest-Fire Detection and Monitoring: The EU-FIRE Project”, In Proceedings of the 14th International Conference on Automatic Fire Detection, [retrieved: September, 2009].

[36] I. P. Anderson, I. D. Imanda, and Muhnandar, “Forest fire prevention and control project”, http://www.fire.uni-freiburg.de/se_asia/projects/ffpcp/FFPCP-02-Interpretation-NOAA-Hot-Spot-Data.pdf, [retrieved: March, 1999].

[37] F. Siegert and A. A. Hoffmann, “The 1998 Forest Fires in East Kalimantan (Indonesia): A Quantitative Evaluation Using High Resolution, Multitemporal ERS-2 SAR Images and NOAA-AVHRR Hotspot Data”, Remote sensing of environment, pp. 64-77, [retrieved: April, 2000].

[38] http://www.erh.noaa.gov/car/firewxopsplan.txt [retrieved: May, 2008].

[39] WSNet simulator, available: http://wsnet.gforge.inria.fr/

Towards Intelligent Sensor Evolution: A Holonic-Based System Architecture

Vincenzo Di Lecce
DEE
Politecnico di Bari
Bari, Italy
v.dilecce@aeflab.net

Marco Calabrese, Claudio Martines
Holsys soc. coop. and Politecnico di Bari
Taranto, Italy
{m.calabrese, c.martines}@holsys.com
{m.calabrese, c.martines}@aeflab.net

***Abstract*— Rapid developments of smart sensor technologies envisage a new era where information handling and knowledge sharing will play a crucial role. Traditional sensors were conceived as simple hardware transducers of physical quantities into measurable signals, eventually requiring an analogue/digital conversion to make data available for software applications. IEEE-1451 family of standards has added to mere transduction some architectural prescriptions to mainly address the issue of connection transparency, a desirable property virtually making any sensor a plug-and-play device. Our percept is that next generation smart sensor-based systems will have to face another challenge: the need to endow devices with the ability to process application-level bits of knowledge to best accomplish their informative goals. As a result, unexpected proactive and dialogue-oriented behaviors will have to be taken into account, thus reducing the gap between what we commonly refer to as smart sensors and intelligent agents. In order to support this view, a semantic-driven sensor-based system architecture is introduced and an example proof-of-concept case study is commented.**

Keywords - smart sensors, intelligent sensor, information processing, holonic modeling

I. INTRODUCTION

In the latest years, the impressive growth of pervasive and ubiquitous devices has entailed a profound evolution in the field of measurement systems. In fact, the mere transduction of physical quantities into analog or digital signals, do not live up to the complexity of modern society any more.

A number of commercial products endowed with "intelligent" functionalities are progressively replacing their old-generation competitors as it happens, for example, in mobile phones [1] or in industrial automotive applications [2].

The reasons fostering the adoption of smart technological solutions on a global scale are both technical and economical and can be enlisted as in [3]:

- reduction of the data communication with the main application processors for some preset functions with a specific value;
- lower system power consumption since some data is filtered and not all of the action needs to be done by the main processor;
- easier integration due to standard digital interface and pre-defined functions, avoiding developing all applications from raw data.

As a result, a large amount of legacy, incompatible, and often proprietary industrial solutions has sprung over years, sometimes producing specifications after upgraded to the rank of international standards (e.g., the ISO 11992-1 CAN bus [4] in the automotive industry).

When viewed through the lens of (artificial intelligence) AI, sensor "smartness" appears to stay in-between merely transduction and complex post-processing, with the boundary left undetermined. Yet, it is difficult to state with certainty to what extent IEEE-1451 compliant sensors can be considered as intelligent agents since they share aspects related to the measurement field with others related to information processing.

In any case, information processing brings forth the need of a reference model for the correct understanding of the obtained measures. Hence, especially in distributed settings like wireless sensor networks (WSN), the share of local ontologies [5] necessary to refine and control the global observed phenomenon, become a key point to address.

Our percept is that next generation smart sensor-based systems will have to face this challenge: endowing devices with the intelligence necessary to communicate with peers and humans to best accomplish their informative goals. The paper is devoted to provide a prospective view on this aspect. The remainder of the paper is as follows. Section II overviews the recent trends in the evolution of smart sensor technologies towards solutions with an increasing level of intelligence embedded; Section III introduces the employed sensor-based communication model; Section IV investigates a possible case study; Section V reports on early implementations and future perspectives; Section VI concludes the work.

II. RELATED WORK

As addressed in [6], there is a language gap between practitioners in the two fields of measurements and AI.

In our view, an oversimplified classification may be considering the first group as focusing on the statistical aspects [7] and the second on the semantics of the measured data [5]. In this perspective, lessening the distance between the two groups would mean narrowing the gap between the measurements of physical phenomena and their interpretation models.

At the very core of any sensor, there is a transduction action permitting the transformation of measurands into electrical/optical outputs. These are more suitable for both data analysis and the activation of high-level processes for data presentation [8].

Once arrived at the electrical/optical stage, an analogue/digital (A/D) converter determines the passage from the physical world to the computational world. In fact, from a computational intelligence (CI) perspective, A/D or D/A conversions can be revisited and generalized in the framework of fuzzy sets and granular computing [8]. In other terms, after the A/D conversion process, digitalized measurement values become containers of information granules that need to be given sense by means of some computational task.

According to this view, smart sensors can be considered as a first attempt to bridge the gap between measurements and information processing since they are purposely conceived as hardware/software transducers able to bring the measured physical signal to an application target level.

IEEE has played a pivotal role in the smart sensors standardization process through the IEEE-1451 family of transducer interface standards [9]. In particular, the IEEE-1451 addresses mainly the significant engineering aspect of connection transparency. The aim is to aid transducer manufacturers in developing smart devices that can be interfaced to networks, systems, and instruments in a plug-and-play fashion.

However, smart sensors are not conceived to offer support in high-level information processing needs such as, for example, the possibility to host self-correction on board, performing data integration and fusion, managing local alarms to reduce the network and the host load. Henceforth, a new family of intelligent sensor capable to deal with the increasing complexity of modern applications is required to go beyond architectural prescriptions defined by the IEEE-1451 standards.

As of the latest couple of years, a new class of devices referred to as 'intelligent sensor hubs' is attracting the focus of the market and the academia. These can be viewed as sensor platforms endowed with a microcontroller unit that pre-process and aggregate external sensor data. An example of this new sensor generation is the MMA9550L motion sensing platform from Freescale company, housing a 14-bit 3-axis accelerometer together with a 32-bit CPU, I2C, SPI and other GPIOs. The low-power and small size enable applications in mobile phones, portables devices and also medical and industrial applications.

The enhancement of sensor platforms with a microcontroller unit derives from the need to overcome the limits of traditional smart sensor technologies, which cannot be customized to any specific application context since the embedded logic is fixed. However, the bare availability of a microprocessor does not suffice to provide an intelligent framework alone. Sensory data, in fact, have to be processed in accordance to an ontology shared with the potential end-users of the information processing task.

Typical limitations related to application-level tasks such as (to cite a few) effective customization, data fusion and self-calibration require to employ some kind of 'software intelligence' currently not addressed by the available standards. Consequently, it seems that the concept of sensor is broadening in the direction of AI [10] and intelligent agents [11]. For this reason, there has been a growing debate in the last couple of years on the appropriateness of using the term "smart sensor" when referring to functionalities typical of intelligent information processing [12].

Intelligent information processing brings sensory data at a higher level where the problems of suitable data interpretation models. At this stage, research on sensors inevitably floods into AI: at this level, transductions continue to occur but at the ontological level thus posing the relevant problem of sense disambiguation [13].

It is useful to stress that sense disambiguation is considered one of the most relevant and difficult problems in AI [14]. Semantics is prone to ambiguity because there can be multiple interpretation models (ontologies) and a reference knowledge [15] crafted by domain experts may not be always available. To correctly interpret sensory data, disambiguation strategies have to be pursued and some recent works begin to focus on that with reference to sensor-based applications [16] [17].

When the positive outcome of the disambiguation procedure in charge of an agent is hindered by the unavailability of sufficient local knowledge, one solution is requesting for help to an external source. A dialogue is then instantiated between the querying agent and one or more respondents (whatever humans or machines), thus producing a 'conversational' [18] behavior.

Communication protocols and architectures have been extensively studied in the field of multi-agent systems (MAS) [19] [20] and human-machine interaction (HMI) [21] [22]. Basing on these studies, a semantic-driven dialogue-oriented system architecture to use in sensor networks for knowledge sharing and information processing is introduced in this paper. Capitalizing on the experience gained in the fields of smart sensors research and CI, we are confident that the proposed architecture provides a preliminary step in the direction of next-generation intelligent sensor-based systems.

III. PROPOSED SYSTEM

In [23], a modeling approach for processing information at multiple granularity levels was presented. This approach grounds on the concept of "holon", introduced by Arthur Koestler in late 60s [24]. Such concept is at the very core of our proposal; for this reason, it is useful recalling briefly the basic properties of holons and holonic systems.

A. *Holons and Holonic Systems Architecture*

In CI, holons and holonic systems can be considered as a recent evolution of agents and MAS [25]. In particular, holon is a recursive agent [26] with peculiar computational aspects such as self-modularity and self-organization.

Self-nested hierarchies of holons are properly called holarchies: with respect to MAS, they account for a more general concept of agent organization comprising multiple nested granularity levels. At the base of a holarchy, atomic holons are found, i.e., holons that cannot be further decomposed according to the problem context. Fig. 1 depicts an agent-based representation of a holarchy.

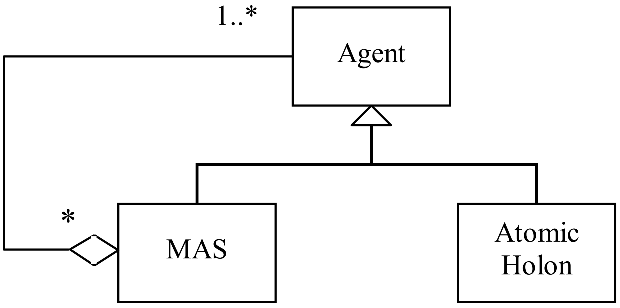


Figure 1. Architectural agent-based representation of a holonic system (slightly adapted from [27]).

It is noteworthy that holons and holarchies, due to holon intrinsic recursiveness, can be considered as two faces of the same coin viewed at two adjacent granularity levels [28]. Consequently, holon encapsulates in a single concept that of system of arbitrary complexity thus overcoming the part/whole and abstraction/enrichment dichotomy [29] that traditionally impedes Reductionist approaches [30]. As a result, the holonic view is highly suitable for modeling complex problems under a multi-granularity levels holistic [31] perspective.

B. Holonic Intelligent Sensor Information Processing model

The same recursive concept of holon encompassing multi-level agent-based architectures can be used to setup an information processing framework for dealing with data interpretation at an arbitrary semantic granularity.

In [13], a holonic model for processing sensor data at multiple granularity levels was introduced. Each level accounts for a different scale of information granules (in the Zadeh's sense [32]) that can be represented by means of linguistic descriptions (whatever fuzzy [33] or categorical), i.e., ultimately by words [34].

The information processing model can be implemented as a composition of two layered parts:

- the holonic transduction layer (HTL) and
- the holonic ontological layer (HOL)

The HTL is the computational layer. It consists in the implementation of the transduction functions necessary to functionally map a measured input signal into an output digital value.

The HOL is the agent-based layer. HOL is engineered as to respond to user query by claiming information from the subordinate HTL or from other agents listening to the bus. In this sense, it encompasses the features of interface and broker agents in hierarchical MAS architectures [35]. It is useful noticing that the querying user can be another intelligent sensor or a human as well. In this last case, system dialogue takes the form of human-computer interaction [22].

HOL works in strict cooperation with HTL. For example, HOL may account for the concepts of 'ppm' and 'ammonia' and answers the query about 'what is the concentration of ammonia' by calling the transduction function Ppm(*ammonia*), which outputs the part-per-million concentration of NH_3 at a given sampling time *k*.

It could happen that the queried intelligent sensor is not able to compute ammonia directly with an acceptable accuracy, because of its high cross-sensitivity. In this case, intelligent sensor equipped with low-cost ammonia detector could ask other sensors to share their data to infer on the presence and accurate concentration of ammonia basing on some computational inference mechanism as the ones presented in [36][37].

Fig. 2 depicts a hypothetical setup made of the five intelligent sensors in Table 1, each one endowed with a basic traditional sensor. By means of the computational techniques presented in [36] and [37], an intelligent sensor can answer the query about the concentration of ammonia by instantiating a dialogue with the other peers connected to the bus.

TABLE I. INTELLIGENT SENSORS EMPLOYED FOR THE HYPOTHETICAL SETUP DISPLAYED IN FIG. 2.

Intelligent Sensor id	***Basic sensor employed in HTL***	***Principal monitored Quantities in HOL***	***Basic sensor cross-sensitivity***
IS1	LM335	temperature	low
IS2	TGS2602	air contaminants	high
IS3	TGS2180	humidity	low
IS4	MQ136	sulfur dioxide	high
IS5	MQ131	ozone	high

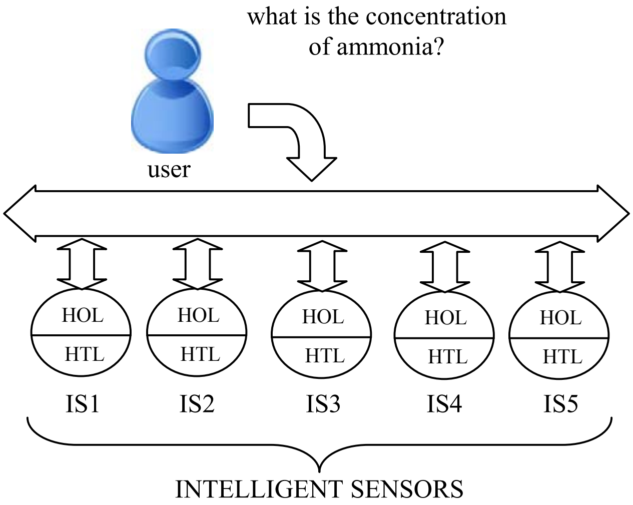


Figure 2. Holonic intelligent sensor example setup. The five intelligent sensors of Table 1 have to work in cooperation to minimize the effect of high-cross-sensitivity and properly respond to user query. Ammonia is not directly measured by any of the employed HTLs, however, it can be inferred by applying CI techniques, i.e., by instantiating a dialogue among intelligent sensors.

C. *System Knowledge Organization and Properties*

As shown by early works on sensor-equipped mobile robots in 80s [38], the problem of organizing and traversing different granularity levels according to both enrichment or abstraction criteria is a central issue from the system engineer's point of view.

In more recent years, the aspect of knowledge granularity has been intensively studied in the framework of granular computing (GrC) [39] and now seems to offspring in the field of holonic systems [28].

According to Pedrycz's view [40], GrC as opposed to numeric computing (which is data-oriented), is knowledge-oriented and accounts for a new way of dealing with information processing in a unified way. Since knowledge is basically made of information granules, information granulation operates on the granule scale thus defining a sort of pyramid of information processing where low levels account for ground data and higher level for symbolic abstraction.

In the holonic semantic model presented in [41], all the holons at the same level (representing one or more sub-holarchies) share the same ontology. The holons at the lowest level receive data from the real world using a set of sensors. Furthermore, these holons can handle various actuators to operate in the real world. On the other hand, all the holons at the higher levels receive data from the holons at the neighboring lower level. In sum, holarchy represents system knowledge across different granularity levels, spanning from the sensors/actuators level to the context-dependent problem conceptualization.

D. *A Holonic Smart Sensor-Based System Proposal*

By coupling the information processing model with the holonic architecture previously reported, a system is obtained where architectural and semantic recursion is operated by means of communication acts among system agents (the interested reader can refer to [42] for a formal specification of agents' communication acts). As a result, a dialogue-oriented system using intelligent sensors as intelligent agents can be modeled.

The employed ontology sharing mechanism is entirely based on communication acts among holons. Communication is a fundamental issue at least for reasons of two orders:

- It allows for representing bits of local information according to a distributed multi-level model of the observed system (refer to [13] for an example holarchy managing multi-level knowledge organization in low-cost gas sensor setups);
- It permits the extraction of inference rules from the environment where the holarchy dwells by means of recently developed computational techniques [43].

The proposed system is portrayed in Fig. 3. The querying user depicted in Fig. 2 becomes the agent managing the HOL while the above mentioned HTL becomes a recursive encapsulation of (holonic) intelligent sensors representing the knowledge objects to use in the information processing task.

As the figure shows, the system is built around the dialogue between HOL and HTL at any possible granularity level. At the maximum possible abstraction level there is, generally, a human user (acting in charge of the HOL) who queries the subordinate HTL (representing the automated part of the system).

The novelty of the model is that traditional user/system dichotomy has disappeared in favor of a holistic view: the user (human or machine) is now part of the model.

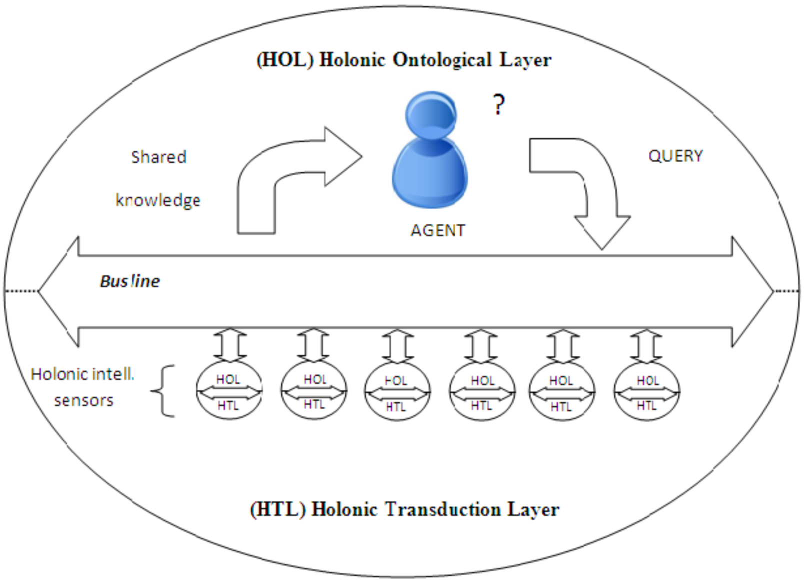


Figure 3. Conceptual model of a holonic intelligent sensor-based system. Since the model is recursive, the image accounts for a generic granular representation level.

IV. Example Case Study

In this Section, a possible application of the proposed intelligent sensor-based system model is discussed. Proof-of-concept case study is the indoor monitoring of various air quality parameters by means of low-cost oxide-based resistive sensors. Due to their cheap manufacturing process, these sensors are prone to show imprecise and inaccurate responses being sensitive to multiple contaminants at once [44]. Furthermore, their output changes significantly with temperature and humidity.

Suppose to consider three low-cost sensors: TGS2602 (air contaminants), MQ131 (ozone and mono-nitrogen oxides) and MQ136 (sulfur dioxide). As shown in [16], this triplet is sufficient to discriminate among CO, NH_3 and SO_2 presence, if appropriate sensor response disambiguation techniques are applied. Additionally, LM335 and TGS2180 can be used for temperature and humidity calibration.

A. General holonic architecture

The reference communication bus architecture is the one presented in Fig. 2. Five holons account for one measured parameter each. One additional holon instead acts as representative for the human user interested in interfacing the system.

In general, it happens that a relevant pattern to monitor is the result of a composition of distributed events (i.e., parameter values involving at least two holons at the same time). In this case, one holon queries the other on the supposedly occurring pattern. If the queried holon verifies that its local parameters are consistent with the warning raised by its paired unit, a proper action is triggered. In general, the more complex the pattern, the more the rounds of communication acts to check pattern verification.

In the following, two example use cases are commented to provide an overview of the possible communication acts exchanged among the hosts connected to the bus. The two examples have been chosen to depict the flexibility of the holonic-based architecture where communication acts can be triggered both from high-level holons (e.g., human user) or low-level holons (local sensors).

B. Use case 1: temperature and humidity self-calibration

In this use case, the action is triggered by the human user (super-holon) asking for the instantaneous concentration of ammonia. User performs her request by means of the PC connected to the LAN bus.

The queried holon (the one with MQ131 connected), according to its datasheet, is highly dependent on temperature and humidity. For this reason, to accurately fulfill its request, it calls the two holons of temperature and relative humidity to receive their data. Once that this information has been obtained, the MQ131 holon is able to calibrate its output and responds to user query with a text string like this:

```
%ANSWER TO QUERY FROM 192.168.8.137/Human%
    Ammonia is   8 ppm
    measured @   8.44 AM UTC+1
    with         temperature = 19°C
                 relative humidity = 65%
    in           AeFLab computer science laboratory
    accuracy     high
```

If, for example, the temperature holon were off-line, the MQ131 holon would continue providing a response, but with lower accuracy. In this latter case, the output string could be like the following:

```
%ANSWER TO QUERY FROM 192.168.8.137/Human%
    Ammonia is   8 +/-5 ppm
    measured @   8.44 AM UTC+1
    with         relative humidity = 65%
    in           AeFLab computer science laboratory
    accuracy     low  - -> could not find temperature
                          information
```

C. Use case 2:detecting a gas

In this use case, the action is triggered by one holon locally detecting a transition from one state to another. The holon looks up its local knowledge base (KB). KB stores all patterns the system consider as important to monitor. For example, as shown in [16], the following pattern accounts for NH_3 presence:

MQ131< θ AND TGS2602> θ AND MQ136< θ

being θ an empirically tuned threshold value.

If MQ136 holon begins to sense local values such that the logical state MQ136 < θ is true, then it raises a query to TGS2602 and MQ131 holons to know their current logical states. If both the two queried holons confirm the NH_3 pattern, then a warning is raised to the PC holon to inform high-level holon (human user).

V. Implementations and Future Works

A test-bed implementation of the previously described holonic architecture is currently under way.

All holons are supposed to communicate their data by means of an Ethernet bus and inform one other about their services (e.g., measuring ammonia or CO) by means of a shared repository accessible via a Web-service interface.

Holons connected to the bus are hosted by the following devices:

- For the HTL, low-cost Linux-based RISC devices for continuous data acquisition and local parameter monitoring have been used. With reference to the overall system holarchy, all these units are responsible for hardware/software transductions by converting physical-level signals into digital parameters.

- For the HOL: one PC hosting complex data processing and visualization. This unit is responsible for extracting coherent IF THEN patterns from the sampled data sent by the HTL. After such process, the HTL is informed of the extracted patterns and can use them either for triggering user-defined actions (such as local alarms) or for knowledge exploration.

Each Linux-based RISC device is equipped with one of the three gas sensors reported in Fig. 2, plus a humidity and temperature sensor to host self-calibration on board. Software on-board has been written in C language to perform data acquisition, compression, storage and publishing. On each device, an ever running task constantly analyze real-time data arriving from all input channels to check if one of the monitored patterns verifies (or is about to verify). In this case, an action is triggered to the high-level holon (for example, a mail is sent to the system administrator if the temperature surpasses 25°C).

The PC station has been equipped with a Java-based data analysis framework realized by the Holsys company. In particular, the framework called H-GIS (Holonic-Granularity Inference System) implements the computational technique presented in [36][43] allowing one to extract inference rules from sampled data.

At the moment, employed acquisition devices write the sampled data into excel files, which are submitted in batch to H-GIS for knowledge extraction. In this setup, H-GIS is the true only agent of the architecture capable of exposing a high-level interface to an external user. In the long run, we aim at endowing each Linux-based RISC device with the same 'intelligence' of the PC hosting the inference system.

One pending issue is to find a suitable communication protocol to use for distributed information sharing in holonic-based systems. According to this objective, our schedule is to evaluate the opportunity of using standard ontology language (such as the Semantic Sensor Network ontology [46]) and rule description language in agent message exchange on bus. Figure 4 reports on the experimental system architecture.

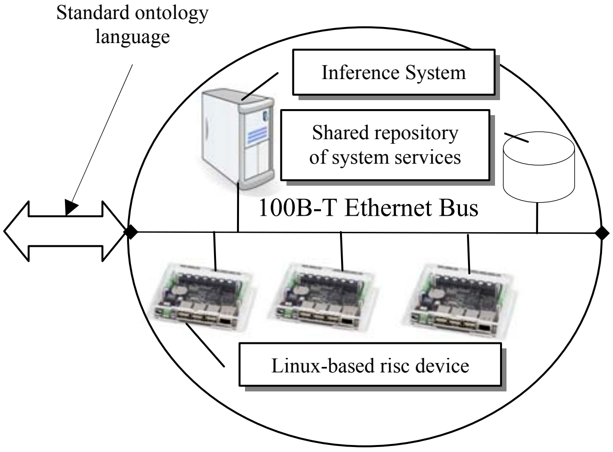

Standard ontology language
Inference System
Shared repository of system services
100B-T Ethernet Bus
Linux-based risc device

Figure 4. Test-bed architecture for holon hosting.

VI. Conclusion

When operational contexts are complex, cluttered and ambiguous, in addition to the relevant features of connection transparency and pre-processing functions, it is useful to endow smart devices also with the ability to interpret data thus showing an intelligent behavior towards the application layer and the human user.

With this objective, a semantic-driven dialogue-oriented system architecture to use in sensor networks for knowledge sharing and information processing has been introduced. The architecture has its backbone in a bus dwelling all holon communication.

Due to intrinsic holon properties, information can be organized according to a multi-level representation. This means that the same request can be answered differently (dependently on the operational context) since it ingenerates communication acts aimed at responding with the maximum possible accuracy level permitted by the current knowledge available. As a result, a communication among holons is instantiated at multiple semantic granularity levels.

One significant aspect that has been skipped here is the reverse engineering effort that has to be carried out in order to identify the possible utterances and consequently the meaningful context-sensitive queries. Furthermore, a proper communication protocol among holons has to be found to leverage real-world implementations. Future works will concentrate on such language-oriented aspect that still need to be formalized and properly tested.

References

[1] N.D. Lane, E. Miluzzo, Lu Hong, D. Peebles, T. Choudhury and A.T. Campbell, "A survey of mobile phone sensing", IEEE Communications Magazine, Vol. 48, Issue 9, Sep. 2010, pp. 140-150.

[2] M.H. Salah, T.H. Mitchell, J.R. Wagner and D.M. Dawson, "A Smart Multiple-Loop Automotive Cooling System—Model, Control, and Experimental Study", IEEE/ASME Transactions on Mechatronics, Vol. 15, Issue 1, Feb. 2010, pp. 117-124.

[3] S. Gervais-Ducouret, "Next Smart Sensors Generation", in Proceedings of the IEEE Sensors Applications Symposium (SAS), 22-24 Feb. 2011, pp. 193-196.

[4] International Organization for Standardization (ISO) 11992-1, "Road vehicles — Interchange of digital information on electrical connections between towing and towed vehicles", International Standard, Second edition, 2003-04-15.

[5] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing", International Journal of Human and Computer Studies, Vol. 43, pp. 907–928, 1995.

[6] R. Taymanov and K. Sapozhnikova, "Problems of Terminology in the Field of Measuring Instruments with Elements of Artificial Intelligence", Sensors & Transducers, Vol. 102, Issue 3, March 2009, pp.51-61.

[7] J. R. Taylor, "An introduction to error analysis: the study of uncertainties in physical measurements", University Science Books, 1997.

[8] A. D'Amico and C. Di Natale, "A contribution on some basic definitions of sensors properties", IEEE Sensors Journal, Vol. 1, Issue 3, Oct. 2001, pp. 183 – 190.

[9] E.Y. Song and K.B. Lee, "Sensor Network based on IEEE 1451.0 and IEEE p1451.2-RS232", Proceedings of the IEEE Instrumentation and Measurement Technology Conference Proceedings, (IMTC 2008). May 2008, pp. 1728 – 1733.

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

[11] A. Rogers, D.D. Corkill and N.R. Jennings, “Agent Technologies for Sensor Networks”, IEEE Intelligent Systems, Vol. 24, Issue 2, March-April 2009, pp. 13-17.

[12] S. Y. Yurish, “Sensors: Smart vs. Intelligent”, Vol. 114, Sensors & Transducers Journal, March 2010, pp. I-VI.

[13] V. Di Lecce and M. Calabrese, “Smart Sensors: A Holonic Perspective”, Published by Springer in Lecture Notes in Bioinformatics, sub-series Bio-Inspired Computing and Applications: 7th International Conference on Intelligent Computing, ICIC 2011, Zhengzhou, China, August 11-14 2011, to appear.

[14] R. Navigli, “Word Sense Disambiguation: a Survey”, ACM Computing Surveys, 2009, 41(2): 1-6.

[15] V. Di Lecce and M. Calabrese, “Taxonomies and Ontologies in Web Semantic Applications: the New Emerging Semantic Lexicon-Based Model”, IEEE International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC'08), pp. 277-283, 10-12 Dec. 2008, Vienna.

[16] V. Di Lecce and M. Calabrese, “Discriminating Gaseous Emission Patterns in Low-Cost Sensor Setups”, Proceedings of the IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA 2011), Sep 19-21, 2011, Ottawa, Canada, pp. 1-6.

[17] V. Di Lecce, M. Calabrese and Rita 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.

[18] F. Quek, “A conversational paradigm for multimodal human interaction,” Proc. of the 30th Applied Imagery Pattern Recognition Workshop, Oct 2001, pp. 80–86.

[19] M. Wooldridge (2009), An Introduction to Multi Agent Systems, 2nd ed., John Wiley & Sons.

[20] K. Sycara (1998), MultiAgent Systems, AI Magazine 19(2): 79-92.

[21] B. A. Myers, “A brief history of human–computer interaction technology,” Interactions Vol. 5(2) pp. 44–54, 1998, ACM Press.

[22] V. Di Lecce, M. Calabrese, D. Soldo, and A. Quarto, "Dialogue-Oriented Interface for Linguistic Human-Computer Interaction: a Chat-based Application", Proc. of the 2010 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems (VECIMS 2010), Taranto, Italy, 6-8 September 2010, pp. 103-108.

[23] M. Calabrese, “Hierarchical-Granularity Holonic Modelling”, PhD Thesis, University of Milan, March 2011.

[24] A. Koestler, The Ghost in the Machine, (1st Ed.) Hutchinson, London (1967).

[25] F. Amigoni, A. Brandolini, V. Caglioti, V. Di Lecce, A. Guerriero, M. Lazzaroni, F. Lombardi, R. Ottoboni, E. Pasero, V. Piuri, O. Scotti, D. Somenzi, Agencies for perception in environmental monitoring, in: IEEE Transactions on instrumentation and measurement. - 55:4(2006 Aug), pp. 1038-1050.

[26] A. Giret and V. Botti (2004), “Holons and agents”, Journal of Intelligent Manufacturing, 15: 645-659.

[27] H. V. D. Parunak and J. Odell (2002), “Representing Social Structures in UML”, International Workshop on agent-oriented software engineering, Vol. 2222, pp. 1-16.

[28] M. Calabrese, V. Piuri and V. Di Lecce, “Holonic Systems as Software Paradigms for Industrial Automation and Environmental Monitoring”, keynote speech paper for the IEEE Symposium Series on Computational Intelligence - SSCI 2011.

[29] B.T. Clegg and D. Shaw, “Using process-oriented holonic (PrOH) modelling to increase understanding of information systems”, Information Systems Journal, 2008, 18: 447-477.

[30] F. Pichler (2000), “Modelling Complex Systems by Multi-Agent Holarchies”, Lecture Notes in Computer Science, Computer Aided Systems Theory - EUROCAST’99, Springer Berlin / Heidelberg, pp. 154-168.

[31] M. Ulieru and R. Doursat, “Emergent Engineering: A Radical Paradigm Shift”, International Journal of Autonomous and Adaptive Communications Systems, Vol. 4, Issue 1, Dec. 2011, pp. 4-38.

[32] L.A. Zadeh (1979), “Fuzzy sets and information granularity, in: Advances in Fuzzy Set Theory and Applications”, Gupta, N., Ragade, R. and Yager, R. (Eds.), Amsterdam: North-Holland, pp.: 3-18.

[33] L.A. Zadeh (1996), “Fuzzy logic = computing with words”, IEEE Transactions on Fuzzy Systems, 4(2): 103-111.

[34] J.M. Mendel (2007), Computing with Words: Zadeh, Turing, Popper and Occam, IEEE Computational Intelligence Magazine, 2(4): 10-17.

[35] V. Di Lecce, C. Pasquale and V. Piuri, “A Basic Ontology for Multy Agent System Communication in an Environmental Monitoring System”, IEEE/CIMSA 2004 – Proceedings of International Symposium on Computational Intelliginece for Measurement Systems and Applications, Boston, MA, USA, pp. 45-50, July 14-16, 2004.

[36] V. Di Lecce and, M. Calabrese, “Describing non-selective gas sensors behaviour via logical rules”, Proceedings of the Fifth International IEEE/ACM Conference on Sensor Technologies and Applications SENSORCOMM 2011, August 21-27, 2011 - French Riviera, Nice/Saint Laurent du Var, France, pp. 6-11.

[37] V. Di Lecce, R. Dario and J. Uva, “A Wireless Electronic Nose for Emergency Indoor Monitoring”, 5th International Conference on Sensor Technologies and Applications SENSORCOMM 2011, August 21-27, 2011 - French Riviera, Nice/Saint Laurent du Var, France, pp. 274-279.

[38] R.Brooks, “A robust layered control system for a mobile robot”, IEEE Journal of Robotics and Automation, Vol. 2, Issue 1, 1986, pp. 14-23.

[39] W. Pedrycz (2001), “Granular computing: an introduction”, Proc. of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference, pp. 1349 – 1354.

[40] A. Bargiela and W. Pedrycz (2003), Granular computing: an introduction, Kluwer Academic Publisher, Boston, Dodrecht, London.

[41] M. Calabrese, A. Amato, V. Di Lecce and Vincenzo Piuri (2010), Hierarchical-granularity holonic modelling, Journal of Ambient Intelligence and Humanized Computing , Springer, Berlin Heidelberg,, 1(3): 199-209.

[42] M. Verdicchio and M. Colombetti, Communication languages for multiagent systems, Computational Intelligence, 25 (2), 2009, 136–159.

[43] 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), 6-8 September 2010, pp. 102-106.

[44] K. Ihokura and J. Watson, The Stannic Oxide Gas Sensor, Principles and Applications. Boca Raton, FL: CRC Press, 1994.

[45] RFC 4511 Standard, Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map.

[46] D. O'Byrne, R. Brennan, D. O'Sullivan, “Implementing the draft W3C semantic sensor network ontology”, 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010, pp. 196-201.

The Profile-based Data Processing Method Using Wireless Sensor-actuator Networks

TaeWook Heo, Jong-Arm Jun, Nae-Soo Kim
IoT Architecture Research Team
Electronics & Telecommunications Research Institute
Daejeon, South Korea
e-mail: {htw398, jajun, nskim} @etri.re.kr

Hoon Choi
Dept. of Computer Sci. & Eng.
Chungnam National University
Daejeon, South Korea
hc@cnu.ac.kr

***Abstract*—Wireless sensor and actuator networks (WSANs) bring many gains to smart building systems. When a control system is unified by a WSAN, and particularly if the network size is wide, a distributed communication and a control method are necessary. But, multi-hop communication and packet sizes among sensors and actuators cause challenges in making such systems. This paper proposes and evaluates a new profile based data processing scheme for a smart building system with WSANs. Experimental results show that the proposed method effectively achieves the reduction of packet numbers and sizes with self-controlled sensors and actuators. We also discuss how to dynamically extend the service of a WSAN with only profile distributions and updates.**

Keywords-Distributed networks; profile based data processing; profile distribution; wireless sensors-actuator networks; (WSANs).

I. INTRODUCTION

Wireless sensor networks (WSNs) are made by small-sized, low-cost, and wireless communications enabled sensors, which have been installed to form various monitoring systems, e.g. a building environment monitoring system [1].

Wireless Sensor Networks (WSNs) that are used by a centralized way gather the sensing information and perform commands through a central server [2]. Therefore, this method has several disadvantages, e.g., poor response time, heavy network traffic, and centralized bottleneck. However, distributed wireless sensors and actuators close to each other distance and the actuators have a decision making algorithm, these have several strengths, e.g., the data traffic balance and reduction, the battery usage reduction, and the network lifetime increase [3-4].

This paper proposes a new profile based data processing method. There are separated the procedures which have the combinations of events and event processing conditions. And adding an actuator to change the service can be performed dynamically.

For example, when the actuator with ventilation service joins the network, the actuator operation is generated by a combination of events for the ventilation task. Also, the actuator with fire alarm service joins the network, the actuator operation is provided by a union of events for the fire monitoring work.

Now, the server provides events conditions and the combination of events for the frequent changes of the service. Additionally, the response time is fast and it is controlled effectively despite disconnection of the central server.

The problems of the existing systems are the applications-oriented approach. Whenever there are changes in the service they should provide and develop the appropriate applications. These points have been raised as a problem. The development of a service and configuration of it which fit the application are more resources intensive.

Therefore, in this paper we introduce to handle easily commands which execute, change, and update the services. Also the protocol for data processing is introduced. Also, the distributed WSAN are more energy efficient than the centralized WSAN.

Section II discusses the related work. Section III proposes the profile-based data processing scheme, and the Section IV discusses the experimental results. Section V shows the conclusion and future work.

II. RELATED WORK

Smart building systems are used for controlling the environment of smart buildings, such as temperature, humidity, illumination, by means of ventilating, lighting, fire monitoring, and so on [5].

Recently, centralized and distributed methodologies have been studied by various researchers. Among them, a distributed control method is designed and compared with a centralized control method in [6-7].

The traditional methods for modelling and controlling building environment system may become impractical when the control-system loops are closed by the WSANs, in which unreliable and incomplete data and network attributes, such as network traffic, should be paid enough notice [8].

At the sensor nodes in a centralized method, the data are collected from the gateway using WSAN are sent to application servers that are located outside. Therefore, the event collected by the application server is detected and actuators that can handle the event are received to the control command.

In this way, because of all data-intensive to the applications server, there are the prolonged data-path weaknesses.

Also, the shortcomings are that the nodes in the neighbourhood of the application server consume more energy than the other nodes.

In this study, to solve these shortcomings, the self-control method between the sensor nodes are generated through the various sensors and actuators that can handle the data are able to process the events through the autonomous connection.

In WSAN technical adaptation of the various sensing data to be sent to the server, WSAN lifetime (time to provide services) will have the advantage of extending by reducing the power consumption of nodes to deliver a control command from any node in the path.

III. WSAN PROFILE BASED DATA PROCESSING METHOD

A. Centralized WSAN and distributed WSAN

For the centralized WSAN, denote N as the number of nodes, R_b the channel bit rate. Then we define a factor, α_A, taking account of the overhead presented by all protocol stack layers. We consider a WSAN where nodes are requested to send their samples (composed of D bytes each) taken from the monitored space every T_R seconds. Here, we can write:

$$N \le R_b \alpha_A T_R \big/ (8D) \qquad (1)$$

This equation offers an approximate estimation of the number of nodes that can be part of a single-sink single-hop WSAN.

For the distributed WSAN, there are multi-sink multi-hop networks. Let us denote by h_m the average number of hops per data sample taken from the field.

We can assume that each sink (denoting as N_s their overall number in the network) can serve up to N nodes with N limited by expressions. Therefore, we can write:

$$N \le N_S R_b \alpha_A T_R \big/ (8Dh_m) \qquad (2)$$

To give a numerical example, assume R_b=250Kbits/s, TR = 10ms, α_A = 0.1, D = 3; then, if there are N_S = 5 sinks in the network, the maximum number of nodes is approximately 50. But, for the single pan, N cannot exceed 10[9].

The proposed technology using the profile is able to change the service at runtime and to modify the events conditions dynamically. Therefore, there are profile distributions, profile executions, and profile updates.

B. Profile distributions

First, the profile distributions are about how to deliver the boundary event conditions and the combination of sensor events from the server to the node.

In general how to distribute event condition and data for each node is sent to individual nodes.

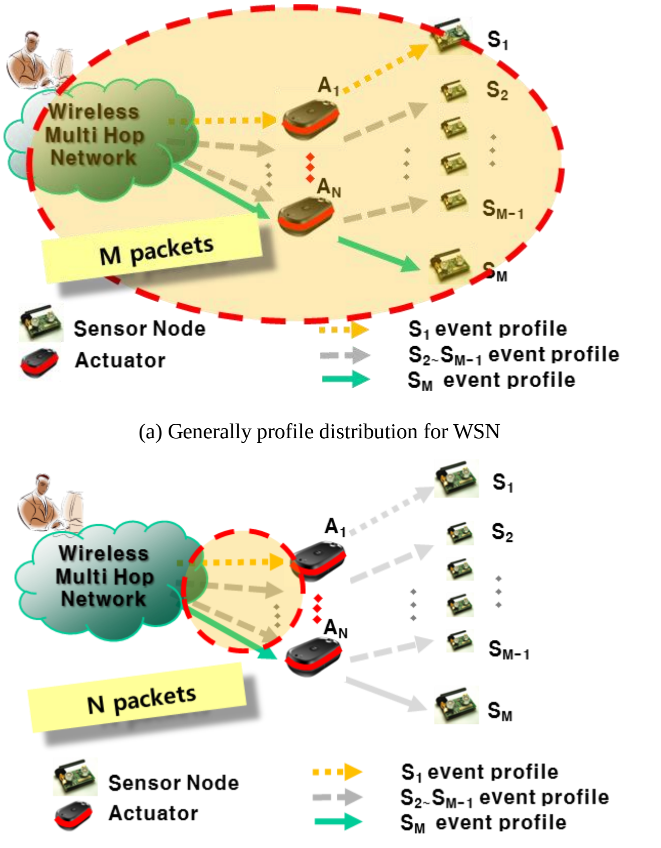
In the process of sending M packets it is about how to send an event to sensor nodes. For example, the condition of sensor node is sent to the sensor node. You can set the event condition that “If the temperature of the sensor reading is greater than 20 degrees, make reports”.

First, Figure 1 (a) shows that each packet sent to the sensor node through the route node M times individually if event profile is sent to the node and the M event profiles through the actuator are sent to the sensor nodes if it sends, or not.

The packet-flow which is sent through the actuator is the most common. In this case, the numbers of the packets sent from network coordinator to the actuator are M.

However, the proposed method sending two messages is separated. First, the step 1 is that the N packets are sent from the network coordinator to the actuator. The step 2 is that the actuator is sending the message to the respective sensor node again. In the two steps, if you place them the same number of packet sent from the actuator to the sensor nodes, the number of packets that are transmitted to the N packets from the M packets.

Typically, the number of sensor node is more than the number of actuator node. Actuator node has the power and performs data processing and routing function.



(a) Generally profile distribution for WSN

(b) The proposed method

Figure 1. Comparison with general WSN method and the proposed method

C. Profile executions

In this paper, we propose that our framework is the logical combination and the separation of the event conditions.

Through the logical combination and union of the event conditions, the specific service can satisfy the various sensor conditions. In this session, we explain the profile based data processing method and the network information for WSAN.

In this paper, the data processing methodology is similar to the node middleware. And our profiles consist of the actuation profile and sensor event profile.

Figure 2 shows the profile based event processing sequence diagram.

The profile based event processing methods are as follows.

1. If the actuator will join the network, it sends the ReportJoin message to the coordinator.

2. The BSI(Building service interface) checks the attribute data of the actuator.

3. The BSI sends to the actuator the actuation event profile packet and the sensor event profile packet.

4. The actuator sends to the sensor nodes in the group the sensor event profile.

5. The sensor node sets the sensor event condition if it receives the profile packet.

6. The sensor node creates the event if the condition of event is met.

7. The sensor node sends the generated event to the actuator.

8. The actuator checks the event generated from the sensor node Actuation and in case the condition of actuator event profile is met, it will perform the actuation control.

9. The actuator sends the actuation event to the server.

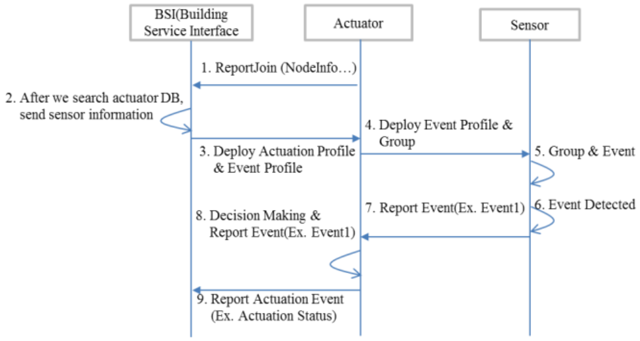

BSI(Building Service Interface
Actuator
Sensor
1. ReportJoin (NodeInfo…)
2. After we search actuator DB, send sensor information
4. Deploy Event Profile & Group
3. Deploy Actuation Profile & Event Profile
5. Group & Event
7. Report Event(Ex. Event1)
6. Event Detected
8. Decision Making & Report Event(Ex. Event1)
9. Report Actuation Event (Ex. Actuation Status)

Figure 2. Profile based Event Processing Diagram

1) *Actuator Profile:*

The actuator profile is the combination of the event conditions. For the decision making in the actuator, this profile includes the actuator command that can be performed.

In Table I, there are the actuator profile structures. They are making up a combination of events.

TABLE I. ACTUATION PROFILE FRAME STRUCTURE

Name		**Description**	**Type**
Command ID		This field is command identification number (0x02)	U8int
Pay-load	Actuation Profile ID	Actuation profile identification number	U32int
	Group ID	Group identification number	U8int
	Grouping Type	Sets the bitmap to generate a group (bitmap: 0-sensor type, 1-loc, etc.)	U8int
	Sensor-Actuator Type	Sets the detailed information of sensor or actuator	U16int
	Default Control Value	Actuator's default control value (e.g. Outlet: Off(0), On(1), and so on)	U8int
	Actuation Control Value	when the combination of events is true, the actuator do the actuation control (e.g., outlet: Off(0), On(1), and so on)	U8int
	Number of Conditions	The number of event conditions	U8int
	Logical Value	Sets the logical condition value	U8int
	Condition Value	Sets the event identification as the condition value	U32bit
	Event Trigger Rate	When an event connects several nodes, we provide the threshold parameter for each node.	U8int

For example, there are three event conditions. The number of condition is 3, their condition values are {0, E1}, {2, E2}, and {2, E3}.

2) *Sensor Event Profile*

The sensor event profile structure is as follows:

For example, the event structure is A1<Sensor value<A2.

The sensor event profile describes the conditions under which an event occurs. If the condition value of the profile is met the value of the sensor, the event occurs. Table II shows sensor event profile frame structure and Table III shows condition types for the sensor event profile.

TABLE II. SENSOR EVENT PROFILE FRAME STRUCTURE

Name		**Description**	**Type**
Command ID		This field is command identification number (0x01)	U8int
Pay-load	Group ID	Group identification number	U8int
	Actuator Address	Actuator address for adapting sensor event profile	U16int
	NumberofEventID	The number of events	U8int
	EventID	Event identification for sensor event profile.	U32int
	EventType	EventType is 0 for sensing EventType is 1 for actuation	U8int
	Sensor-Actuator Type	Detailed identification for sensor and actuator	U16int
	Condition Type	Conditions between events (between, less than, greater than, outer, and so on)	U8int
	Event Condition Value 1	First parameter for condition value 1(A1)	U32int
	Event Condition Value 2	Second parameter for condition value 2(A2)	U32int
	Number of Sensor Node	The number of sensor node belonging to the group	U8int
	SensorNode Address	The address of sensor node (N-number Array)	U16int

TABLE III. CONDITION TYPE

Condition	0x00	A1<x<A2
	0x01	A1<x<=A2
	0x10	A1<=x<A2
	0x11	A1<=x<=A2
	0x0f	A1<x
	0x1f	A1<=x
	0xf0	x<A2
	0xf1	x<=A2
	0x88	x< A1 or x>A2
	0x98	x< =A1 or x>A2
	0x89	x< A1 or x>=A2
	0x99	x< =A1 or x>=A2
	Etc.	Reserved

D. Profile updates

Profile updates as the profile distribution are to update the combination of actuation events and the relation of the sensor event threshold conditions.

In some cases, you may cancel your profile and change it. Therefore, the profile modification with the new conditions has the advantage of dynamically changing the actuator status.

IV. DISCUSSION AND RESULTS

The profile based data processing scheme for verification the technology was applied to the actual five story building. The experimental environments used in the building are as follows:

The number of the actuators is 3, the number of the sensor nodes is 9, and the number of route nodes is 3 [10].

The emulation data is generated for applying this technology. Randomly, it generates for performing actuation.

For the real network, we used 5-story building test-bed and because the situation does not occur, we used the following emulation data.

A. Environment and Emulation data

- Precondition: a sampling interval for 24 hours is 15 minutes.

- Light sensor: The value of luminance set 1 time a day, its range is from 100 lux to 500lux.

- Temperature sensor: The value of temperature set six times a day, its range is from 10 to 55 degrees.

- CO sensor: The value of CO sensor sets to 4 times a day, its range is from 0 to 80.

- CO_2 sensor: The value of CO_2 sensor sets to 4 times a day, its range is from 0 to 2250.

In the N nodes, the packet size is as follows:

Hop counts from the coordinator to each node can be computed in the multi-hop environments. Also, there limited from each node to the actuator in wireless sensor and actuator networks.

The equation (3) represents the relationship between hop counts and the number of nodes. In the N sensor nodes, the total packet sizes from the sensor nodes to the coordinator are the greater than a minimum N and less than a maximum product of hop counts and N.

$$N \le \sum_{i=1}^{N} Hop(i) \le N \times Hop_{\max} \qquad (3)$$

where,
Hop(i): the hop count of the ith node from pan coordinator.
N: total number of nodes.
Hop$_{max}$: the greater Hop count in the Hop(i).

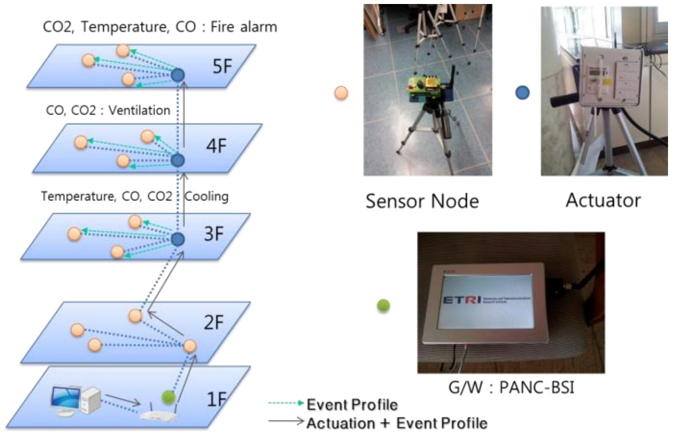

CO2, Temperature, CO : Fire alarm
5F
CO, CO2 : Ventilation
4F
Temperature, CO, CO2 : Cooling
3F
2F
1F
Sensor Node
Actuator
ETRI
G/W : PANC-BSI
Event Profile
Actuation + Event Profile

Figure 3. Event condition and test environment

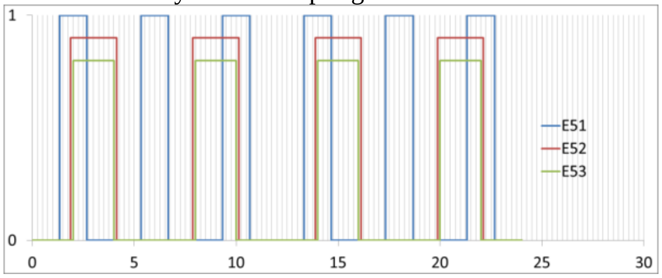
B. Packet numbers and packet sizes

In this experiment, a general centralized way and our distributed way using profile are a comparison of the predicted data.

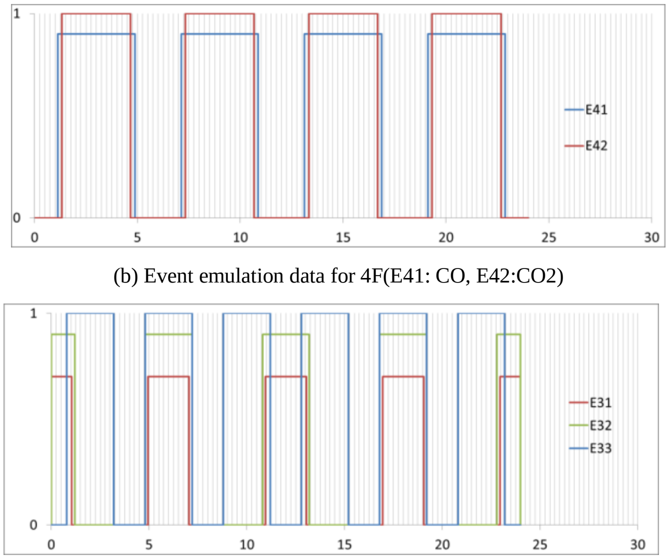
TABLE IV. EVENT PACKET ESTIMATION

Event packet numbers(5F based)		WSN	Proposed method
Sensor event packet numbers (Fire Alarm: Temperature > 40℃ and CO > 50ppm and CO_2 > 1500ppm)	Temperature sensor: 6 Hop	12*6= 72	12
	CO Sensor: 6 Hop	8*6= 48	8
	CO2 Sensor: 6 Hop	8*6= 48	8
Actuation control packet numbers	Actuator: 5Hop	8*5 = 40	8*5 = 40
	Total	208	68

Figure 4 shows the experimental value for the event emulation. The event value of each floor (3F, 4F, and 5F) is emulation data by the 24 sampling data.



(a) Event emulation data for 5F (E51: Temperature, E52: CO, E53:CO2)

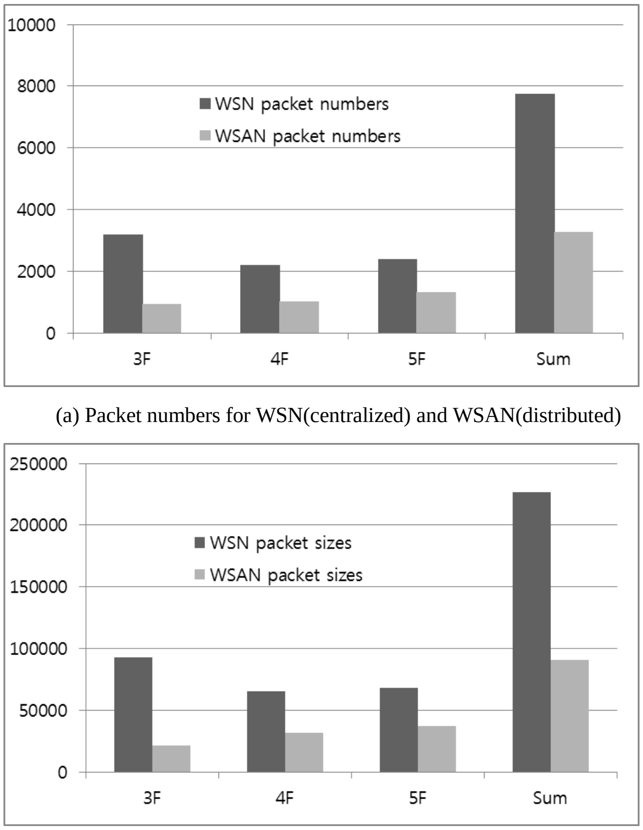


(b) Event emulation data for 4F(E41: CO, E42:CO2)

(c) Event emulation data for 3F(E31: Temperature, E32: CO, E33:CO2)

Figure 4. Event emulation data for 3F, 4F and 5F.

Figure 5 shows the experimental result for the event simulation. The packet number and packet sized of the proposed method are reduced about 50% than those of centralized method.



(a) Packet numbers for WSN(centralized) and WSAN(distributed)

(b) Packet sizes for WSN(centralized) and WSAN(distributed)

Figure 5. Test Result (data comparison central WSN and distributed WSAN for 3F, 4F, 5F a)packet numbers, b)packet sizes)

V. Conclusion and Future work

In this paper, the packet numbers and packet sizes were compared in the term of the centralized control and the distributed control. And the proposed profile-based distributed data processing technology showed good performance. Therefore we had the benefit of changing the profile easily when the service was run and the profile was distributed. It was good extensibility for the sensor network frameworks.

The profile based data processing techniques to provide a general framework in an effective way. In addition, the events that meet the criteria even more efficient in how data is processed.

The Obstacles in this paper to apply the profile to be limited service on the test adaptation and the actual test-bed is difficult.

Future work includes the accurate sensing data processing scheme such as event handling for the spatial conditions should be considered. And, the voting method is considered how to decide the truth of the event in case same multiple sensor. The way to adapt the aforementioned spatial correlation will increase the accuracy of the data.

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.

References

[1] X. Cao, J. Chen, Y. Yang, and Y. Sun, "Development of an integrated wireless sensor network micro-environmental monitoring system,*" ISA Trans.*, vol. 47, no 3, pp. 247-255, Jul. 2008.

[2] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: Research challenges," *Ad Hoc Netw*., vol. 2, no.4, pp. 351-367, Oct. 2004.

[3] H. Ramamurthy, B. S. Prabhu, and R. Gadh, "Wireless industrial monitoring and control using a smart sensor platform," *IEEE Sensors J*., vol. 7, no. 5, pp. 611–618, May 2007.

[4] Y.-J. Wen and A. M. Agogino, "Wireless networked lighting systems for optimizing energy savings and user satisfaction," *in Proc. IEEE Wireless Hive Netw. Conf*., Aug. 2008, pp. 1–7.

[5] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Trans. Ind. Electron*., vol. 56, no. 10, pp. 4258–4265, Oct. 2009.

[6] X. Cao, J. Chen, Y. Xiao, and Y. Sun, "Building Environment Control with Wireless Sensor and Actuator Networks: Centralized Versus Distributed, " *IEEE Tran. Ind. Electon.*, vol. 57, no. 11, pp. 3596-3605, Nov. 2010.

[7] J. Chen, X. Cao, P. Cheng, Y. Xiao, and Y. Sun, "Distributed Collaborative Control for Industrial Automation With Wireless Sensor and Actuator Networks," *IEEE Tran. Ind. Electon*., vol. 57, no. 12, pp. 4219-4230, Dec. 2010.

[8] A. I. Dounis and C. Caraiscos, "Advanced control systems engineering for energy and comfort management in a building environment – A review," *Renew. Sustain. Energy Rev.*, vol. 13, no. 6/7, pp. 1246-1261, Aug/Sep. 2009.

[9] R. Verdone, D. Dardari, D. Mazzini, and A. Conti, Wireless Sensor and Actuator Networks Technologies, Analysis and Design. Academic Press: London, UK, 2008.

[10] IEEE-TG15.4, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE standard for Information Technology, 2003.

A Reliable and Real-Time AggregationAaware Data Dissemination in a Chain-Based Wireless Sensor Network

Zahra Taghikhaki, Nirvana Meratnia, Paul J.M. Havinga
Pervasive Systems, University of Twente
Enschede, Netherlands
(z.taghikhaki, n.meratnia, p.j.m.havinga)@utwente.nl

***Abstract*—Time-critical applications of Wireless Sensor Networks (WSNs) demand timely data delivery for fast identification of out-of-ordinary situations and fast and reliable delivery of notification and warning messages. Due to the low reliable links in WSNs, achieving real-time guarantees and providing reliable data is quite challenging. To ensure data reliability, traditionally various retransmission mechanisms have been used, which in turn introduce extra delay. In this paper, we propose READ, i.e., a reliable and real-time aggregation-aware data dissemination to ensure reliable and fast data delivery in a chain-based WSN. We also investigate the relatively unexplored topic of impact analysis of Time To Live (TTL) and link reliability parameters on network performance in terms of attained hit ratio for three different approaches, i.e., READ, QoS-ACA, and the stop-and-wait (S-W) ARQ to assess the appropriateness of each method facing different conditions. The simulation results show READ performs better in terms of hit ratio compared with QoS-ACA and S-W ARQ when link reliability is low and packet's TTL is short. Although not being the primary goal of READ, energy consumption of the protocol is also much lower than the other two approaches.**

Keywords- Chain-based wireless sensor network, reliable/real-time data dissemination.

I. INTRODUCTION

Wireless sensor networks are one of the most promising technologies for applications such as structural health monitoring. Monitoring operational performance of large civil engineering (infra)structures such as bridges, tunnels, highways, and water pipes require deployment of long linear arrays of sensor nodes. As the length of these (infra)structures is often much greater than their width, their topologies resemble a long chain. Long linear chain-type sensor networks have often a large number of hop counts and to operate for a long time, they usually need to work on a low duty cycle. The large number of hop counts challenges existing data dissemination protocols already designed for wireless sensor networks, while the low duty cycle introduces extra delays.

Time-critical applications such as disaster management and structural health monitoring highly depend on the availability of real-time data as in these applications data is neither useful nor valuable if it is received after its Time To Live (TTL). Outdated data is not only be useless but also harmless as it may have negative impacts on the decisions made by providing invalid information. Moreover, transmitting expired data depletes the energy of relaying nodes inappropriately. Due to their poor link quality, providing real-time guarantees and data reliability in WSNs is quite challenging. Link quality can be easily affected, among others, by weather, temporary obstacles, and mobility. Most of existing real-time algorithms applied in other networks than WSNs assume network is reliable and packets are not lost because of low link quality. Therefore, they cannot be directly applied to WSNs. The higher the packet loss due to low quality links, the lower the performance of a real-time wireless sensor network. One of the mechanisms to provide data reliability is through introduction of redundant data by transmitting the same data multiple times, which results in high energy consumption. It is somewhat clear that ensuring reliability may not always go hand in hand with ensuring network lifetime. Depending on the application at hand, one can also argue that energy efficiency, real-timeness, and data reliability are not always equally important. Data reliability and real-timeness become significant for applications dealing with identification of out-of-ordinary situations as well as warning and notification systems, while continuous monitoring applications demand long network lifetime and can tolerate latency and data unreliability to some extent [1] by using local techniques such as filtering and anomaly detection. In the latter applications, data aggregation is considered as a significant primitive, which not only helps save energy and bandwidth by communicating less data but also provides meaningful information to the end-users.

The main problem addressed in this paper is the design of an aggregation-aware data dissemination protocol for a chain-based WSN suffering from low reliable communication links while satisfying the delay and reliability requirements of the packet. Unlike existing techniques, our proposed protocol combines real-time and reliability guarantees for each packet and increases hit ratio (the percentage of the packets received by the base station before their deadline expire). To deal with the energy consumption and to enrich data, we utilize data aggregation on the intermediate nodes as far as it does not influence packet deadline. We also investigate the relatively unexplored relationship between the TTL and link reliability parameters and their impact on the hit ratio for three different approaches, i.e., READ, QoS-ACA [2] and an ARQ approach, to assess the appropriateness of each method facing different conditions.

The rest of this paper is organized as follows. First we briefly discuss state of the art and preliminaries of this study. Then a detailed description of our proposed approach will be

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-207-3

provided, which will be followed by performance evaluation. Finally we draw some conclusions and future works.

II. Related work

Several data aggregation protocols have been proposed for WSNs in the past. However only a very few of them consider both reliability and timeliness and aim to ensure them simultaneously. Real-time guarantees are usually provided through either real-time scheduling or real-time routing. SPEED [3] is a well-known protocol addressing soft real-time guarantee in WSNs in such a way that packet deadline is mapped to a velocity requirement. The node with a velocity higher than the specified requirement is more likely to be chosen as the upstream node. MMSPEED [4], is an enhanced version of SPEED aims to meet reliability and timeliness requirements together while utilizing multipath routing to handle reliability such that number of path is in direct proportion to the required reliability. Timeliness is supported by combining the SPEED idea with packet prioritization, which is done on the basis of the required speed for each packet. R2TP [5] proposes a reliable and real-time data dissemination, in which reliability is satisfied by sending several copies of one packet through multipath such that sum of the reliability of the considered path is equal or higher than the requested reliability. This packet is dropped by the intermediate nodes if the elapsed time of a given node is greater than the delivery time requirement. Otherwise, it forwards that packet through multi paths using the given node's table, which stores the delay of different paths. Soyturk et al. [6] present a reliable data acquisition approach for time-critical application of WSNs. Reliability is provided similar to techniques of [4][5] leveraging multipath approach while real-time concern is supported by packet prioritization. This technique therefore deals with the priority scheduling to handle queuing delay, which is of the main causes of making end-to-end latency. Almost all of the aforementioned reliable approaches support reliability by sending several copies of a packet through different paths. To the best of our knowledge, there is no well-explored work to address these two quality of service (QoS) parameters together in a chain-based WSN, in which only one path can be established between source and destination nodes. Moreover, since approaches of [4][5] are proposed for data dissemination rather than data aggregation, they must employ other methods to filter out redundant data in case of availability of duplicate sensitive aggregation functions like sum or average. QoS-ACA [2] aims to fast, reliably, and energy efficiently aggregate data in a chain-based WSN and send the aggregated value to a base station. To ensure reliability, it leverages the benefits of retransmission without using any acknowledgement (Ack). It utilizes the optimum number of retransmission to ensure the required reliability. It considers the residual and required energy of each sensor node and the distance between node and the base station as two main criteria to select a node as an aggregator. However, it does not guarantee delivery of a packet to the base station within its deadline.

III. Preliminaries

The preliminaries of this study is presented here.

A. *Quality of Service Parameters*

An increasing number of WSN applications require real-timeness as their QoS parameter. Applications may have one of the following four notions of time:

– Time-unrestricted: which indicates no dedicated deadline exists and application at hand is not time critical.
– Soft Real Time (SRT): based on which the usefulness of a packet received after its deadline decreases, which in turn results in a graceful degradation of the performance. SRT-based approaches aim to reduce deadline miss ratio of the packets and are common in WSN because of the unpredictability nature of these networks.
– Firm Real Time (FRT): on which, the usefulness of a packet received after its deadline is Zero. FRT methods can tolerate infrequent deadline misses.
– Hard Real Time (HRT): HRT applications highly rely on receipt of all packets before their deadline ends.

Another QoS parameter requirement of many WSNs applications is reliability. One commonly used approach to ensure reliable data delivery in a failure prone environment is sending several copies of one packet from a single source node towards the destination node. To know whether data is received by the destination, one of the following techniques is used:

– Sending an acknowledgement: in this technique if the acknowledgement packet is lost due to link/network failure, source node continues sending copies of the received data, which leads to high energy dissipation.
– Sending multiple copies without sending any acknowledgement: although this approach reduces the acknowledgement overhead, it requires a solution to ensure data reach to the destination after sending *n* copies of a packet.

B. *Duty-cycling*

Efficient energy consumption has one of the highest priorities in WSNs to ensure long network lifetime. As one of the most energy-expenditure operations is transmitting data, each sensor node must turn its radio off and goes to asleep state most of the time to obtain significant energy saving. In a duty-cycle-based power management scheme, each sensor node goes to sleep and wakes up periodically. The proportion of the time that each sensor node spent in sleep mode has direct impact on the data delivery delay, packet loss, and throughput. The shorter the duty cycle, the lower event detection probability and the longer detection delay. In a scheduling scheme, a sensor node is allowed to switch between three operation modes:

– Sleep mode: which results in low power consumption. In this state the radio of a node is turned off but the sensors may be operational.
– Active mode: which itself includes two operational states: receiving state (RX), and transmitting state (TX).

- Idle state: in which radio is ready to receive or transmit data. According to the conditions the radio is changed to the appropriate active state.

Figure 1 presents the state diagram illustrating the main states of the radio and the ways state transitions occur. Once the sleeping time(Ts) is over, the radio must undergo a transition to idle state. On the other hand, the radio of a node must be switched to off as soon as the active time (TA) is finished. It is worth noting that these four states have different levels of energy consumption, which differ from one radio model to another.

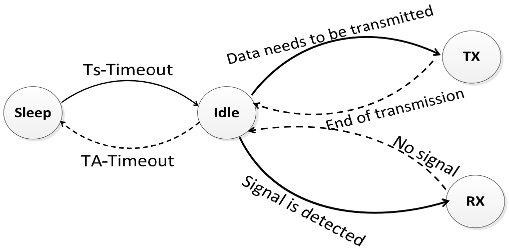


Figure 1. State diagram for radio states

C. Network Model

We make the following assumptions regarding the WSN. The WSN consists of *N* sensor nodes deployed in a linear topology and two base stations are located at two sides of the chain. We have described in [2] a mechanism using which a chain leader can be selected through which sensor data is forwarded to the base station. In case of not being a chain leader, sensor nodes can only communicate with their direct neighbors. The location of sensor nodes and the base stations are fixed and are known a priori. We have chosen for this network model as this is the case in many structural health monitoring applications. In these application, sensor nodes are placed at known and fixed locations (for instance, at critical locations) in a long linear array topology and send their data periodically or upon detection of abnormal situations via relaying nodes to a base station. It should be noted that we assume the packet loss probability of each link is almost fixed and does not change much. This is justified by the fact that we aim to find the relationship between TTL and link reliability with the network performance. We are aware that link reliability changes frequently in practice. In our ongoing work, therefore, we enhance READ by considering dynamic changes of links reliability. As far as this paper is concerned finding the relation between TTL and link reliability with the network performance requires a fixed link reliability to be assumed.

Every sensor node in a chain must send its data to its upstream neighbor which is selected in the chain construction phase. Intermediate nodes along the path to the chain leader aggregate the data received from the downstream nodes with their own data and forward the local aggregated value towards the chain leader. The chain leader, also called the aggregator, must perform final aggregation on the data received from two sides of the chain and then forward the result to the base station directly.

To motivate the need to address both data reliability and real-timeness in our protocol, let us consider the network illustrated in Figure 2, which consists of six sensor nodes such that one of them is selected as the chain-leader and a packet, whose TTL is 10s, should be forwarded from S_0 towards the leader. Let us assume that time required to deliver a packet from S_0 to the leader is 3s and from the leader to the base station is 1s. Clearly, this packet will be received by the base station after 4s. This implies that 6s from its TTL is remained, which can be exploited to achieve higher network performance. We can spend this time for either (i) increasing aggregation degree of the leader or (ii) improving transmission reliability of the network. If the network has high reliable links and it is almost guaranteed that the packet is received by destination through the first transmission, it is better to spend this remaining time for the aggregation process and to increase aggregation degree of the leader. In this case, leader can put the received packet on hold and perform aggregation on other packets which are on the way and will be received within limited time duration of the waiting packet. The remaining TTL time can also be used to improve transmission reliability by utilizing a retransmission mechanism and sending several copies of the given packet. This is particularly useful when network suffers from packet loss.

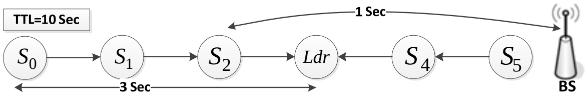


Figure 2. An example of a chain based network

D. Policies regarding Reliability and Real-Timeness

To cope with unreliability of the links, this paper leverages the benefit of retransmission approaches without using acknowledgement in order to support reliable transmission. Therefore, similar to QoS-ACA, we are going to estimate the optimal number of retransmissions for each link. Each sensor node sends multiple copies of the same packet to its upstream neighbor in order to improve transmission reliability. Since receiving a packet after its deadline is not only useless but also depletes energy. It is highly preferable to drop such packets to prevent wasting energy of the intermediate nodes relaying the packet. Since the retransmission mechanism used in QoS-ACA imposes extra delay, we modify it to meet a given latency requirement by retransmitting as far as packet's deadline is not expired. A key question here is how to assign the remaining TTL of a given packet to relaying nodes for their retransmission or in another word for how long a packet can be delayed on the intermediate nodes so that the reliability gain and on-time end to end delivery ratio can still be maximized. We answer this question by allocating the available packet's TTL proportionately to the packet loss probability of the links along the forwarding path to judiciously and fairly use the packet's TTL on intermediate nodes in such a way that reliability gain and on-time end to end delivery ratio is maximized.

IV. DETAILED DESCRIPTION OF READ PROTOCOL

Our algorithm starts with chain construction using PEGASIS algorithm proposed in [7]. In a given chain, one node must be selected as the leader in order to do the final aggregation and to send the aggregated result to the base station. Two QoS parameters, i.e., reliability and energy consumption as well as two assigned weights, are considered to make different criteria for electing a leader. To this end, we introduce the following formula:

$$B^T(S'_j) = (B^R(S'_j))^{W_R} \times (B^E(S'_j))^{W_E} \qquad (1)$$

$$B^E(S'_j) = \left(\frac{RsdEg(S'_j)}{IniEg(S'_j) \times RqEg(S'_j)}\right) \qquad (2)$$

$$B^R(S'_j) = \frac{1}{N-1} \times \sum_{i=0}^{N} EER(S_i, S'_j) \qquad (3)$$

$$EER(S_i, S_{CL}) = \begin{cases} \prod_{k=i}^{CL-1} HHR(S_k, S_{k+1}) & CL > i \\ \prod_{k=CL}^{i-1} HHR(S_k, S_{k+1}) & CL < i \end{cases} \qquad (4)$$

Where S' represents a set of sensor nodes, which are able to directly communicate with one of the base stations and CL represents the candidate leader. The hop-by-hop reliability (HHR) between two sensor nodes are obtained using $HHR(S_i, S_{i+1}) = 1 - p_{pktloss}(S_i, S_{i+1})$. By having the hop-by-hop reliabilities, base station must evaluate the appropriateness of each member of S' to be an aggregator. To this end, base station first calculates the end-to-end reliability from each sensor node to the designated leader by employing (4). At the second step, base station finds the benefit of each candidate leader in terms of reliability (B^R) by averaging sum of the end-to-end reliability of each sensor node to the designated leader using (3). This selection ensures the maximum reliability that this chain can provide. Base station also finds the benefit of each candidate leader in terms of prolonging lifetime (B^E) using (2) where $RsdEg(S'_i)$ denotes residual energy of S'_i, $IniEg(S'_i)$ is initial energy of S'_i and $RqEg(S'_i)$ denotes the required energy of S'_i if being selected as the leader. After finding all the benefit values in a chain, base station selects the sensor node, which provides the maximum benefit as the leader for a given chain using (1). The higher the benefit value of (1), the higher the probability of being selected as a leader. One should note that aggregation takes place at different locations of the network as the leader selection process results in selecting an aggregator in a dynamic way based on the energy and reliability parameters. Due to application specific nature of WSN, different applications have different requirements. Therefore, assigned weights (*w*) to each QoS parameter of (1) can be changed in order to satisfy the application requirements. As we have two base stations, two chain leaders can be selected such that they can communicate with one of the base stations directly. Sensor nodes must select one of these chain leaders to send their data to. This selection is done by considering distance between sensor nodes and the chain leaders.

To find out optimal number of copies which must be sent through each link, we follow the following steps:

Each sensor node must update packet TTL employing (5) where TT (Transmission Time) denotes the time required to transmit one packet to the upstream node.

$$\begin{cases} TTL_{Sourcenode} = PacketTTL & \\ TTL_j = TTL_{j-1} - C_{j-1} \times TT & j < LID \\ TTL_j = TTL_{j+1} - C_{j+1} \times TT & j > LID \end{cases} \qquad (5)$$

Where: $0 < C_j \leq n_j$

Using (5), required time to send C copies of a packet from one node to its upstream node is subtracted from the TTL of the packet where LID represents leader ID. As we do not know which packet copy is received first, upstream node by looking at the copy number of the packet can easily recognize C. In the next step, the chain leader assigns a portion of the remaining TTL of the packet to each node by dividing the packet loss probability of the link adjacent to a given node by sum of the packet loss probabilities of the links located between the given node and leader. Equation 6 calculates optimal number of packet copies for each node to meet deadline requirement of the packet. The second term of (6), put an upper bound for the number of packet copies for each link only by looking at the packet loss rate of the given link and the reliability requested by the application.

$$n_j = \min(n'_j, \log {}^{1-RqRl}_{PL(S_j,S_{j+1})}) \qquad (6)$$

$$n'_j = \begin{cases} \dfrac{PL(S_j, S_{j+1})}{PL(S_{LID}, BS) + \sum_{i=j}^{LID-1} PL(S_i, S_{i+1})} \times \dfrac{TTL_j^{new}}{TT} & j \neq LID \\ \dfrac{PL(S_j, BS)}{PL(S_{LID}, BS) + \sum_{i=j}^{LID-1} PL(S_i, S_{i+1})} \times \dfrac{TTL_j^{new}}{TT} & j = LID \end{cases} \qquad (7)$$

Where S_{i+1} represents the upstream node of S_i in the chain and $PL(S_j, S_{j+1})$ denotes the packet loss between S_j and S_{j+1}. Equations 5 and 7 can be used if radios of all nodes are never turned off. As we also consider duty cycling in order to save energy, (7) requires significant revisions to include sleeping times which greatly influences remaining TTL of the packet. Therefore, the way we calculate the optimal number of packet copies changes. We assume that the duty cycle of the node is in such a way that if one node sends the first copy of the packet to its upstream node, it is awake at that time but it is likely the upstream node goes to sleep mode before finishing transferring all copies of a given packet. Therefore, we first should find the number of time slots in one awake time period (nS) by having transmission time (TT) of one packet and awake time period (AwT) using $nS = \frac{AwT}{TT}$. It is worth noting that having duty cycle (DC) and toggle period (TP), the AwT can be calculated easily as $AwT = TP \times DC$.

Then we need to calculate number of time slots that each packet requires (rS) to be able to transmit all its copies along the path towards the base station. As we are allowed to send (or receive) each copy of one packet in one time slot, the number of time slots corresponds to the number of packet copies. Therefore, having required time slots for a given TTL is enough to know the number of packet copies which must be transmitted to increase reliability while TTL requirement of the packet is met. To find rS, first we need to calculate the number of required awake cycle (nRc) to transmit all packet copies through different nodes using (8) while AsT represents the time the node is in sleep mode.

$$nRc = \frac{TTL}{nS \times TT + AsT} \quad (8)$$

Where: $AsT = TP \times (1 - DC)$ (9)

Each time slot for a given node represents one receipt or one transmission for that node. Leveraging (8), (9) and (10), required time slots (rS) for the given packet is calculated. Actually, source node using (10) describes the TTL of a packet in terms of time slots.

$$rT = TTL - (nS \times TT + AsT) \times nRc \quad (10)$$

$$rS = \frac{rT}{TT} + nRc \times nS \quad (11)$$

Where rT denotes remaining time of the packet after using nRc awake cycles to transmit packet copies. Then, the optimal number of sent copies for node S_j to meet deadline requirement of the packet by considering the packet loss probabilities of the upward links can be obtained by (12). The first term of the right part of (12) represents the portion (Ptn_j) of S_j from TTL remaining of the packet.

$$n_j = \min(n'_j, \log_{PL(S_j,S_{j+1})}^{1-RqRl}) \quad (12)$$

$$n'_j = \begin{cases} \dfrac{PL(S_j, S_{j+1})}{PL(S_{LID}, BS) + \sum_{i=j}^{LID-1} PL(S_i, S_{i+1})} \times lS_j & j \neq LID \\ \dfrac{PL(S_j, BS)}{PL(S_{LID}, BS) + \sum_{i=j}^{LID-1} PL(S_i, S_{i+1})} \times lS_j & j = LID \end{cases} \quad (13)$$

$$\text{Where: } \begin{cases} lS_{SourceNode} = rS, & lS_j = lS_{j-1} - C_{j-1} \\ 0 < C_{j-1} \le n_{j-1} \end{cases}$$

Here n_j represents the number of copies of a given packet which should be transmitted by the node S_j. Each sensor node upon receiving a packet must also update remaining or left time slots (lS_j) of the packet employing (13), using which required time slots to send *C* copies of a packet from one node to its upstream node is subtracted from the available time slots of the packet.

Figure 3 shows the psuedocode of READ protocol.

V. Performance Evaluation

We used Java JDK 6 to implement all algorithms and the simulation environment. We perform simulations for different TTL, link reliability and duty cycle values. Each simulation is executed 100 times. In this section we aim to compare READ, which employs retransmission mechanism without any Ack while keeping an eye on packet's TTL remaining time, with (i) QoS-ACA, which is also a retransmission mechanism without Ack while ignoring TTL parameter and (ii) S-W ARQ, which is a retransmission mechanism with Ack. Traditional acknowledgement protocols, namely, stop-and-wait (S-W), go-back-n (GBN), and selective repeat (SR) [8][9][10], try to retransmit one erroneous frame regardless of the link reliability state. We compare our method with a hop by hop S-W ARQ which is a well-known ARQ scheme. We consider hit ratio and energy consumption as performance metrics. Hit Ratio is defined as the percentage of the packets received by the base station before their deadline expire. We aim to find out the relationship between different packet loss probabilities and various TTL values with the gained network performance for these three approaches in order to know in which condition which method should be employed to provide reliability and real-time concerns simultaneously.

A. Description of scenarios

For simulation, a chain of sensor nodes is formed consisting of 51 sensor nodes randomly distributed in a linear topology. Two base stations are located one hop away from the rightmost and the leftmost nodes of the chain. The output power of our radio model (TICC2420) is programmable in eight levels (from approximately –25 to 0 dBm). Therefore, every sensor node in case of being a leader utilizes the highest power level to provide the longest transmission range, otherwise the minimum power level which is required to reach the closest neighbor is employed. We change the packet loss probability ($p_{pktloss}$) on the links from 0.01 to 0.9. In all simulations, the first main source node is the middle node of the chain (S_{25}), which must select either the left side or right side leader towards which it transmits its data. This selection is done by looking at provided delay which is in direct relation with the link reliability and distance. The second and the third source nodes are the leftmost and the rightmost nodes in the chain. The toggle period (TP) is 1500 ms and energy threshold (θ) is 0.1.

```
Initialization
1. Construct chain using PEGASIS
2. Find $S' = \{S_i \mid S_i \text{ is in Commuicati on range of } BS\}$
3. $Leader = \{S_i \mid S_i \in S' \,\&\, best\ satisfy\ Equation\,1\}$
4. Duty cycling schedule;
5. $Ptn = \{\bigcup_i ptn_i \mid ptn_i \text{ is portion of } S_i \text{ from } TTL\}$
6. $BS \xrightarrow{(Ptn, LID)} S_{LID}$
7. $S_{LID} \xrightarrow{(Ptn, LID)} S_{LID-1}, S_{LID+1}$
8. $S_i$ receives $(Ptn, LID)$
9. Repeat {
            Repeat { $S_i$ sends $(Ptn, LID)$
              } until ( $S_{i-1}$ receives $(Ptn, LID)$)
            $i = i - 1$ and go to step 9}
        until ( $\forall S_i$ receives $(Ptn , LID)$ )

READ Protocol
1. if (event detected by $S_i$ )
        $S_i$ calculates $n_i$ using equation 12
2. if (( $S_i \neq LID$ )) {
    a. Repeat {} until ( $State(S_{i+1}) == Awake$ )
    b. $numberofSentCopies_i = 0$
    c. Repeat { $S_i \xrightarrow{(Data_i)} S_{i+1}$
           $numberofSentCopies_i ++$
    d. } until (( $State(S_{i+1}) == Asleep$ ) or
           ( $numberofSentCopies_i == n_i$ ))
    e. if ((( $State(S_{i+1}) == Asleep$ )) and
           ( $numberofSentCopies_i \neq n_i$ ))
        {Repeat {} until ( $State(S_{i+1}) == Awake$ );
           Go to step 2.b }}
3. else if (( $S_i == LID$ )) {
    a. $S_{i+1} = BS$; Run step 2.b to 2.d
    b. if ( $RsdEg_{LID} < \theta \times IniEg_{LID}$ ) {
        ( $hE = \{S_i \mid S_i \in S' \,\&\, RsdEg_i > \theta \times IniEg_i\}$ )}
    c. if ( $hE == \varnothing$ ){ for (each $S_i \in S'$ )
           { $IniEg_i = RsdEg_i$ }
    d. Go to 3.b}
    e. else { BS finds another leader based on
        Equation 1}
4. if ( $S_{i+1}$ receives $Data_i$ )
5. { $AggData_{i+1} = Aggregate(AggData_i, Data_{i+1})$
    $Data_{i+1} = AggData_{i+1}$; $i = i - 1$ and go to step 2}
```

Figure 3. Pseudocode of READ

The results of two duty cycles, 0.99 (radio is almost always ON) and 0.1 (radio is almost always Off) are represented in this paper to better judge about duty cycling impacts. The other simulation parameters are listed in Table I.

TABLE I. SIMULATION PARAMETERS

No. of nodes	51
Area size	1m x 260m
Mac layer	IEEE 802.15.4
Transmit bit rate	250 kbps
Operation frequency	2.4 GHz
Packet size	128 bytes
Radio model	TI CC2420
Transmit current at 0dBm	17.4 mA
Transmit current at -25dBm	8.5 mA
Receive current	18.8 mA
Supply voltage	(1.6 – 2.0 V)
Idle current	0.426 mA
Transmission range	10-90 m
Receiver sensitivity threshold	-95 dBm

B. Performance Evaluations

1) Hit Ratio

The achieved hit ratio is plotted for these three methods when the Link Reliability (LR=1-PacketlossProbability) can be selected randomly from a set of intervals shown in Figure 4 and duty cycling is 0.1. Figure 4 illustrates attained hit ratio as the packet TTL increases from 80 to 3200 ms. It can be seen that hit ratio of READ is higher than S-W ARQ when the link reliability in the chain changes randomly between 0.1 and 1 or between 0.4 and 1. READ also outperforms QoS-ACA when TTL of the packet is small (smaller than 1500ms). It can be seen from Figure 4 (middle and bottom) that when the lower bounds of link reliability and TTL are increased, performance of S-W ARQ improves. We can conclude that if both link reliability and TTL of the packet are quite high, performance of all three techniques in terms of hit ratio is the same. Otherwise, READ outperforms S-W ARQ and outperform QoS-ACA in case of having short TTL. To have a better judgment about the exact relation between TTL of the packet and link reliability with attained hit ratio, in the following graphs the lower bound of the link reliability interval is increased from 0.1~0.97 that means the reliability of a link should be higher than the given lower bound. Figure 5 illustrates the hit ratio graphs of two different duty cycles for these three approaches. The left side graphs show impact of duty cycle 0.99 on hit ratio while the right side graphs are for duty cycle 0.1. From Figure 5, one can see that when either TTL is short or link reliability is low, READ has better hit ratio. But when TTL is long and link reliability is quite high (TTL>1500, LR>0.8), S-W ARQ outperforms READ because it has enough time to utilize acknowledgment and also because of high reliable link, packets are almost never lost. Although, the hit ratio of READ in these conditions (TTL>1500, LR>0.8) are almost 1 but it has a little fluctuation between 0.97 and 1. It is worth noting, the sharp changes seen in right side graphs of Figure 5 when TTL is about 1500ms are because of using duty cycling for sensor nodes. In case of S-W ARQ, one node requires to frequently switch between sending and receiving mode to be able to handle sending a packet in one time slot and receiving (or waiting to receive) corresponding acknowledgement in the next time slot. Therefore, half of time slots in one awake time period are used for the acknowledgement. This is not the case for READ or QoS-ACA approaches. READ and QoS-ACA utilize all time slots for sending several copies of the packet. The higher TTL, the greater number of awake cycles every node is allowed to utilize to send packet and receive acknowledgement in order to ensure reliability while packet TTL has not yet been expired. READ and QoS-ACA also undergo these sharp changes when duty cycle is too small (i.e. 0.1) as they cannot send all packet copies in one awake cycle that is about 150ms and they have to wait for another awake cycle(s) to be able to send rest copies. When TTL raised to 1500ms (which is start point of another awake cycle), the rest copies can also be sent and the hit ratio suddenly improved especially in case of high reliable links. Also, compared with QoS-ACA, READ has better hit ratio when TTL parameter is short (shorter than 500 ms). In this case, QoS-ACA sends several copies of a packet, especially when LR is low. QoS-ACA satisfies the required reliability for the given packet only for the first few hops, in which the packet has not yet been dropped due to TTL expiration.

Also, QoS-ACA outperforms RRDA when TTL is greater than 500ms and link reliability is lower than 0.3. This is due to the fact that RRDA has to be deterministic for supporting real-timeness and hence always ponders the worst case (longest delay) which means every packet may reach (if it could reach) its upstream node on the last retransmission. Therefore, downstream nodes cannot delay one packet more than the time is assigned to them. But it is also likely that the packet is received by an upstream node before *n* retransmissions. Therefore, the downstream nodes could spent this extra time for their own benefit and do more retransmission while meeting the packet deadline. QoS-ACA exploit this fact in order to increase hit ratio in case of large TTL and low link reliability.

2) *Energy Consumption*

Figure 6 provides a comparison between useful energy consumption (energy spent for packets received before their expiration) and total energy consumption for these three reliable approaches and for two duty cycles 0.99 (right graphs) and 0.1 (left graphs). It is clear that there is almost no difference between useful and total energy consumption for READ, as it drops packets which are more likely not to reach the base station on time. As illustrated in Figure 6, READ is much more energy efficient than QoS-ACA and S-W ARQ particularly in case of low reliable links and short TTL. The reason for this is that in case of low reliable links, data packets or acknowledgement packets are much more likely to get lost. In addition, in case of short TTL, intermediate nodes in QoS-ACA and S-W ARQ approaches will still relay expired packets towards the base station which comes in the expense of energy consumption.

VI. A HYBRID APPROACH

Figure 5 shows each of these three approaches outperforms the other two under some conditions. Therefore, it is more efficient to leverage the benefit from each in a hybrid approach to achieve maximum performance in terms of hit ratio. The idea behind this hybrid approach is to make a selection among these three approaches by looking at the TTL of the packet and link reliability interval related to a given chain. The performance of the hybrid approach is plotted in Figure 7. One can see, this hit ratio graph inherits advantages of the associated graphs of Figure 5.

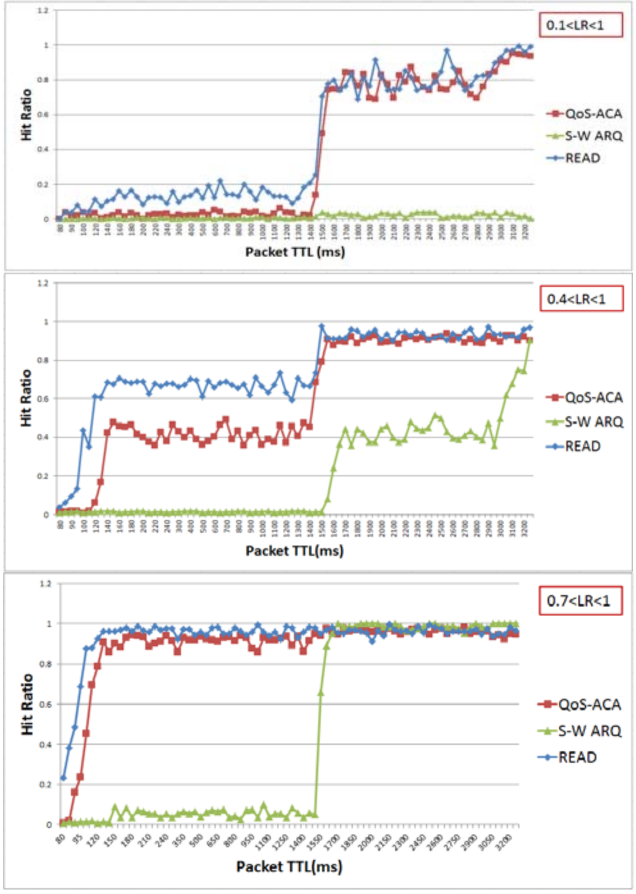


Figure 4. Hit ratio vs. Packet deadline for 0.1<LR<1(top),0.4<LR<1(middle),0.7<LR<1 (bottom)

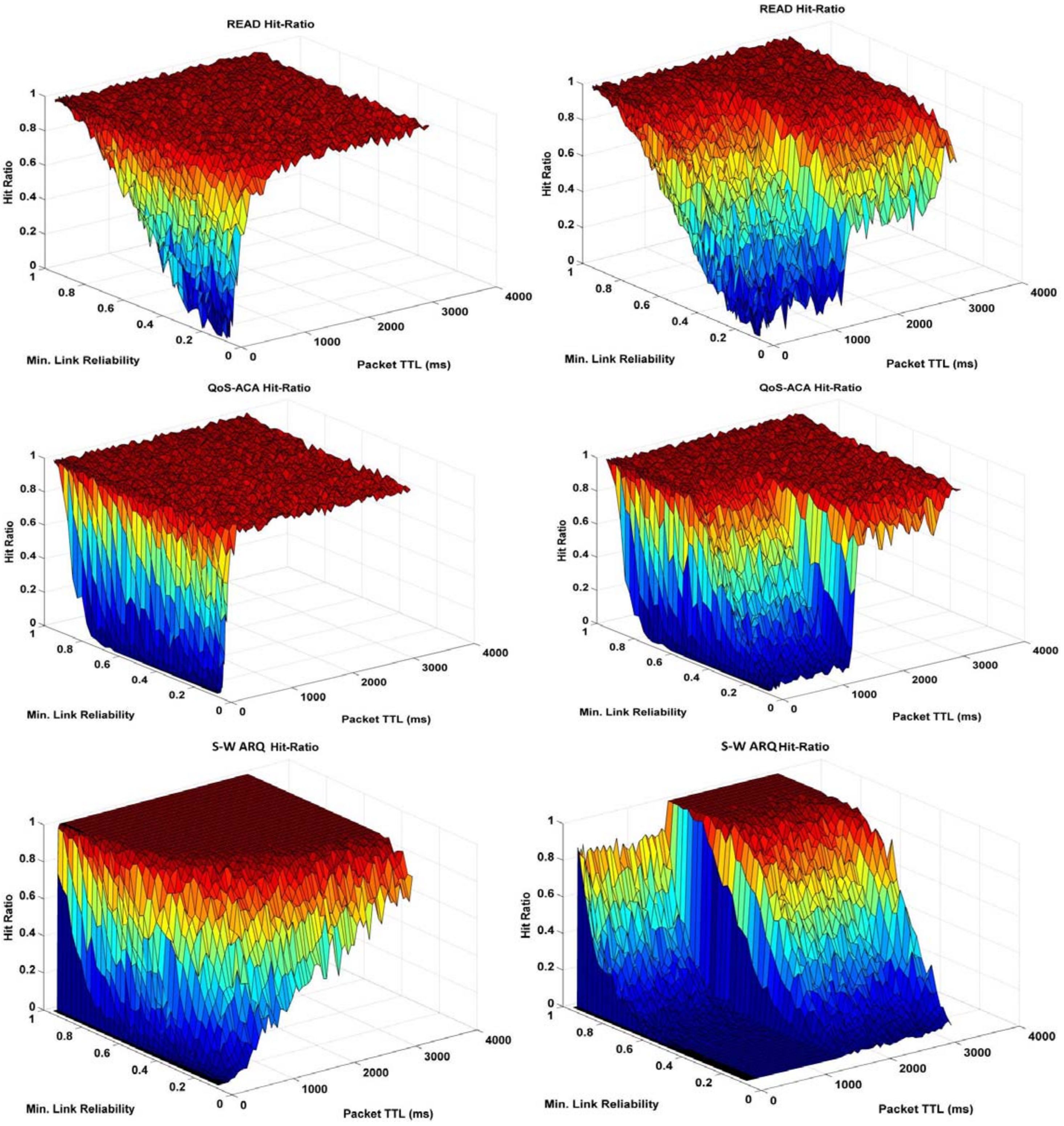


Figure 5. Hit ratio vs. Link reliability vs. Packet deadline for READ (top) QoS-ACA(middle) and S-W ARQ(bottom) while duty cycle of the left graphs is 0.99 and right graphs is 0.1

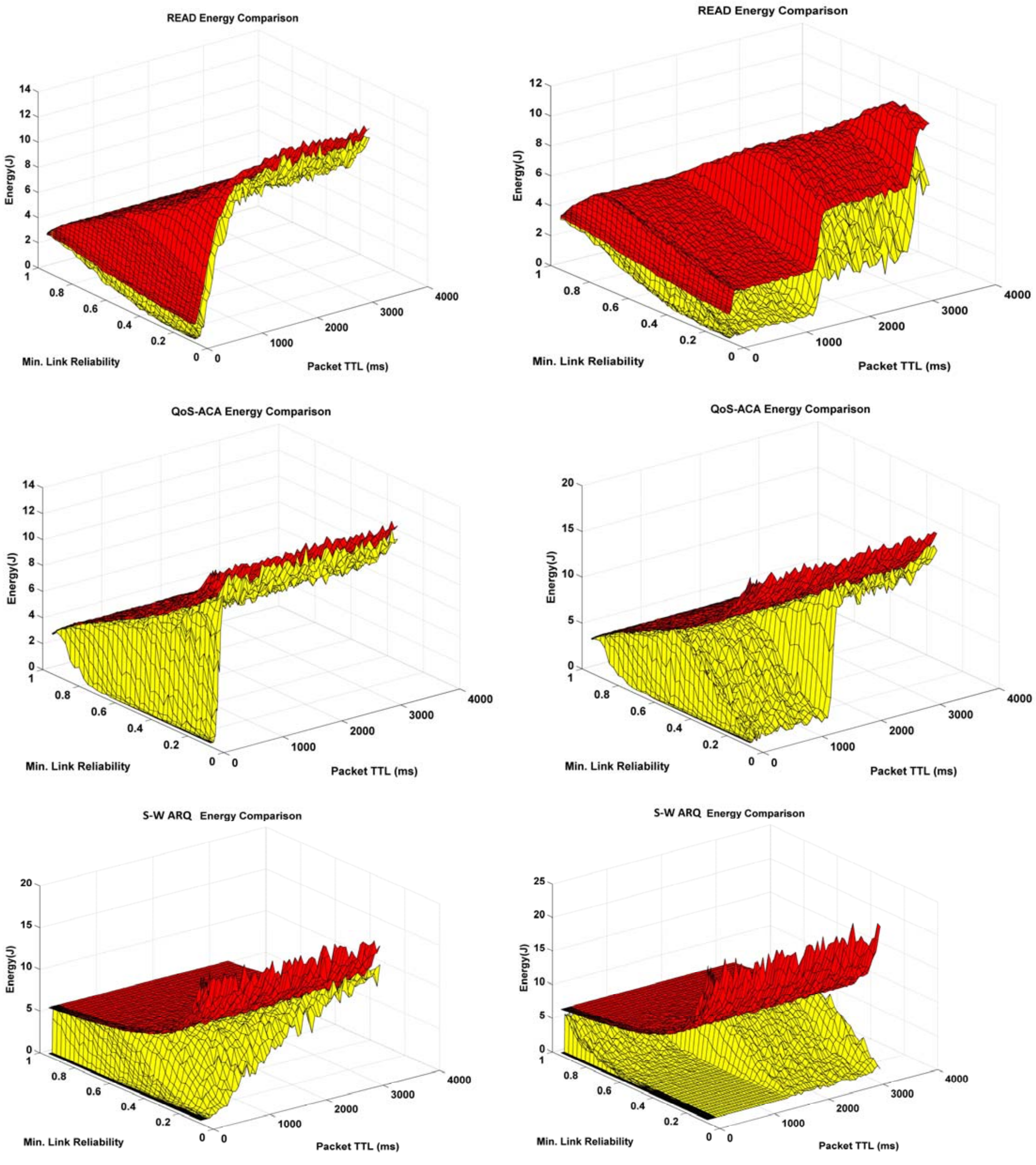


Figure 6. Comparison Between Total and Useful Energy Consumption for READ (top) QoS-ACA(middle) and S-W ARQ(bottom) while duty cycle of the left graphs is 0.99 and right graphs is 0.1

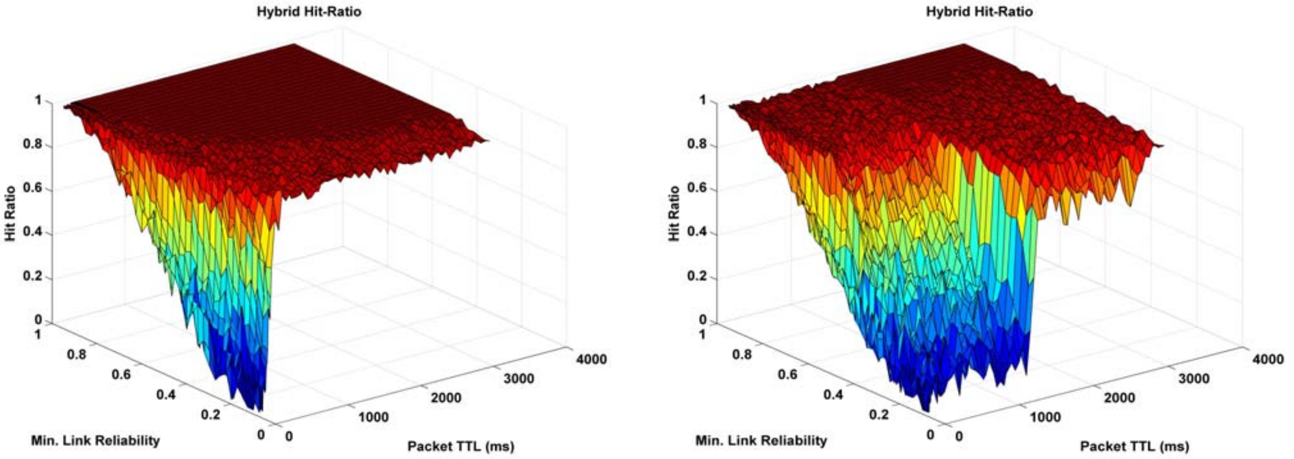


Figure 7. Hit ratio vs. Link reliability vs. Packet deadline for Hybrid approach while duty cycle of the left graphs is 0.99 and right graphs is 0.1

VII. Conclusion and future work

In this paper, we propose READ, a reliable and real-time aggregation-aware data dissemination protocol designs for long chain-type wireless sensor networks, to cope with the problem of efficient data gathering of delay constrained sensor data. Long linear chain-type sensor networks have often a large number of hop counts and to operate for a long time, they usually need to work on a low duty cycle. We investigate the relatively unexplored relationship between TTL and link reliability with the attained hit ratio for time-critical WSNs.

READ allocates available packet's TTL proportionately to the packet loss probability of the links along the forwarding path in order to judiciously and fairly use the packet's TTL on intermediate nodes in such a way that reliability gain and on-time end to end delivery ratio is maximized.

READ assumes the packet loss of each link is fixed for each simulation and therefore does not update the packet loss of the links dynamically based on the last status of the links. This is due to that fact that in this paper we focus on finding the relationship between TTL and link reliability with the attained hit ratio. In our ongoing work, we enhance READ by considering dynamic changes of links reliability and will modify READ in such a way to be able to adaptively change number of copies each node is allowed to send based on the last status of the link reliabilities.

We also consider comparing READ with forward error correction mechanisms in on future work to know how well READ functions .

VIII. Acknowledgement

This work is supported by IST FP7 STREP GENESI: Green sEnsor NEtworks for Structural monItoring project.

Reference

[1] Dezfouli B., Radi M., Nematbakhsh M., and Abd Razak S.,: A Medium Access Control Protocol with Adaptive Parent Selection Mechanism for Large-Scale Sensor Networks. In IEEE 24th International Conference on Advanced Information Networking and Applications, pp 402-408, Singapore (2011).

[2] Taghikhaki, Z., Meratnia, N., Zhang, Y., and Havinga, P.: QoS-aware Chain-based aggregation in cooperating VCN and WSN. In the book of Roadside Networks for Vehicular Communications: Architectures, Applications and Test Fields, In press (2012).

[3] He, T., Stankovic, J., Lu, C., and Abdelzaher, T.: SPEED: A stateless protocol for real-time communication in sensor networks. In 23rd International Conference on Distributed Computing Systems, pp. 46-55, RI (2003).

[4] Felemban, E., Lee, C., Ekici, E., Boder, R., and Vural, S.: Probabilistic QoS guarantee in reliability and timeliness domains in WSN. In INFOCOM, pp 2646-2657(2005).

[5] Kim, K., Park, S., Park H., and Ham, Y.: Reliable and real-time data dissemination in wireless sensor networks. In IEEE Military Communication. pp 1-5, San Diego (2008).

[6] Soyturk, M., Altilar, D.: Reliable real-time data acquisition for rapidly deployable mission-critical Wireless Sensor Networks. In IEEE INFOCOM, pp 1-6, Phoenix (2008).

[7] Lindsey, S., Raghavendra, C.S.: PEGASIS: Power-efficient gathering in sensor information systems. In IEEE Aerospace Conference, pp 1125-1130, Montana (2002).

[8] J. Walrand, Communication Networks: A First Course. Boston, MA: Irwin and Aksen, (1991).

[9] F. Halsall, Data Communications, Computer Networks and Open Systems. U.K.: Addison-Wesley, (1996).

[10] M. Schwartz, Telecommunication Networks: Protocols, Modeling and Analysis. CA: Addison-Wesley, (1987).

Classification of Faults in Sensor Readings with Statistical Pattern Recognition

Valentina Baljak†*, Kenji Tei* and Shinichi Honiden†*
†*University of Tokyo*
**National Institute of Informatics*
Tokyo, Japan
Email: {valentina, tei, honiden}@nii.ac.jp

***Abstract*—In wireless sensor networks, frequent faults are caused by general characteristics and the direct exposure to the environment. Accumulation of these faults can lead to the progressive decrease of reliability and accuracy of sensor readings. We focus on detection and classification of faults within sensory data independently of the underlying cause. We propose a complete and consistent fault classification based on two aspects. The first aspect is continuity and frequency of the occurrence, and the second is the existence of observable and learnable patterns. Given that modeling of faults prior to the detection is a fundamental process, we address it with statistical analysis and theoretical approach. We rely on centralized and straightforward detection methods using neighborhood vote. For the full classification phase, we propose the use of statistical pattern recognition with a priori modeling of faults. Current results show that this method works comparatively well when applied to collected data in data centric dense wireless sensor network.**

***Keywords*-Fault Tolerance; Wireless Sensor Networks; Pattern Recognition; Faults Classification**

I. Introduction

Wireless sensor networks (WSN) are complex systems that consist of a large number of small, cheap devices. We deploy a WSN to collect and process data in order to understand the behavior of a monitored entity. Often, the network needs to perform demanding scenarios in a harsh environment. The significantly lower intrinsic reliability of sensors and actuators than that of integrated circuits in enclosed packaging comes from their direct contact with the environment [1].

Ultimately, the goal of WSN is to provide accurate data about monitored phenomena efficiently over the maximum possible period. Even when perfectly calibrated in the beginning, network will accumulate a number of faults over the time. This leads to the shortening of its effective lifetime, defined as the time of operation in which network reliably provides accurate data.

This research focuses on detecting faults that occur in sensor readings within dense data-centric WSN. In particular, it keeps a focus on the pattern of occurrences as they appear in readings of each sensor node. These observations are independent of the underlying cause of the fault. In this case, a fault is the manifestation of erroneous reading within the data, regardless of the underlying cause for this reading. In this case, we rely on the neighborhood vote and spatial and temporal correlation of the readings, while there is no correlation amongst faults developing on each node.

One of the challenges for fault detection is a proper modeling of faults. Detection of faults relies on these models for the accuracy. We attempt to provide initial models for the detection and classification of faults as they can be observed in collected readings. Frequency and continuity of occurrence is the basis for the proposed classification of faults. The aim is to provide complete and consistent classification, independent of the underlying causes for faults, such as errors in calibration or packet loss.

Amongst challenges that WSN face is the quality of service, where the most important aspect is the amount and the quality of the information that can be extracted at any given sink about the observed object or area. Fault tolerance at all levels of the network is a necessary trait of the required QoS, especially the quick recovery from a fault.

Lifetime of a network is another crucial figure of merit. It depends largely on the energy consumption. For this reason, fault tolerance mechanisms have to find a way to balance the cost of communication and computation. One straightforward definition of a lifetime is given as the time before a first node runs out of energy and stops transmitting [2]. However, data accuracy and reliability might fail long before that, due to accumulated faults. Goal of this research is to use fault mechanism to prolong the effective lifetime of the network, defined as a time in which network provides reliable and accurate data.

Analysis of related work on fault tolerance in WSN is discussed in Section II. Details of the proposed approach and results are presented in Section III, followed by conclusion, discussion and future work in Section IV.

II. Fault Tolerance in Wireless Sensor Networks

Fault tolerance is a fundamental requirement for efficient and reliable operation of WSN. In order to provide a quick recovery, fault latency must be low. This means that a network must be able to eliminate the effect of the fault in a short time. Some classic approaches rely on neighborhood vote techniques or data redundancy, and the existing literature covers them extensively [2], [3]. Apart from these

ISBN: 978-1-61208-207-3

techniques, there is a trend towards using intelligent data fusion and machine learning to provide more flexible fault recovery, for example, Shell et al. propose fuzzy data driven fusion with statistical process control [4].

For example, Takruri et al. propose SVR-UKF-IMM framework for auto calibration [5]. It is an efficient method for fault detection and error correction, but it requires a repetition in each round of data collection. The presented method enables each sensor to predict a correct reading based on information from the previous cycle. Although this approach is flexible, it leaves the network unaware of the occurrence and location of a fault. The sink will never receive an actual reading from the sensor, only a corrected one. With the centralized approach, the network would be aware of the exact node where the fault has occurred.

Koushanfar et al. give a detailed analysis of fault tolerance [1]. Amongst other requirements, they state that it is preferable to develop approaches that require little additional computation regardless of any additional communication requirements as the answer to the common dilemma about the tradeoff between computation and communication. We believe that centralized approach to the fault tolerance corresponds to this requirement since it does not require additional communication, and it ensures enough computational power. Centralized approach is supported by Ni at al. [6]. Also, they point out that any value exceeding a high value threshold is not necessarily a fault. Assumption that nodes need only to be correlated and have similar trends is not enough. Authors emphasize the role of modeling of data since it might provide the basis for comparison if there is a lack of other references. Consistent initial classification of faults can provide a sufficient basis for better modeling of faults initially, as well as learning models from experience. Sharma et al. examine the prevalence of sensor faults in different datasets obtained from real deployments [7]. Authors focus on several types of transient errors and evaluate several different methods for the fault detection. Their conclusion is that the frequency of appearance varies. It ranges from 0% - 20% for different types of errors and different datasets. In the end, authors conclude that the influence to the overall functionality of the network is significant regardless of the frequency of faults. Also, they believe that online handling of faults is an important task. Yao et al. in [8] support this opinion and they propose a straightforward approach to the online detection using time series.

Approach presented here addresses fault tolerance in data centric, densely deployed network with assumed spatial and temporal correlation. For comparison, literature offers some interesting examples of different approaches in different settings, as seen in Tiwari and Thai, who are concerned with providing a fault tolerant virtual backbone as the routing infrastructure [9]. Dual cluster cooperating scheme for data gathering network is proposed by Huang et al. [10]; with the advantages of reduced data loss and a small communication overhead. Also, this scheme can detect both, link faults and node faults. Interesting application and results from a real-time deployment in the avionics is reported by Alena [11].

Fault tolerance is an essential part of reliable sensor network operation. It is obvious that different types of networks and different applications pose varied requirements. To provide a satisfactory fault tolerance scheme, any approach should concern itself with all of the main aspects of fault tolerance, fault models, detection and diagnosis and resiliency mechanisms. Classical techniques, like neighborhood vote, provide a reliable, easy to adapt approach for detection. However, machine learning techniques provide more flexibility for recognizing the type of faults and handling them accordingly. Also, these methods provide a critical difference in terms of early discovery of faults that occur continually over the prolonged period. This significantly affects capability of the network for the quick recovery.

III. Classification with Statistical Pattern Recognition

Part of the fault tolerance recognition and resiliency mechanisms are various criteria for fault classification. In general, error is a manifestation of a fault inside of a program that can occur either at the fault site or at some distance. Together with the fact that each level of abstraction has its own types of faults, current literature covers classification of faults based on various criteria.

Faults can occur in different layers of WSN. If we look at the location and cause of faults, most commonly they appear at physical layer, since sensors and actuators are most prone to malfunctioning. Faults that can occur at this level can besuch as physical layer, where we can have communication faults or hardware malfunctions and energy supply problems. Since sensors and actuators are the most vulnerable components, at this level we can classify faults as:

- calibration systematic errors,
- random noise
- complete malfunctioning

Calibration errors are probably a key source of all faults since they can manifest themselves as a bias or a drift throughout the lifetime of the sensor node.

In the middleware, focus moves towards data aggregation, filtering and sensor fusion, all of which are tasks dependent on accurate and reliable sensor readings. However, it is difficult to provide a fault tolerance at the level of a single sensor node in an economic way. Addressing faults at the application level is efficient, but requires a customized way of addressing each issue. On the other hand, this provides flexibility to address faults regardless of the resource or level in which faults appear [1].

If we take a different look at faults, for example, how often and how long they appear, they can be classified based on the time and persistence as **permanent faults**, continuos and stable in time; **intermittent faults**, occasional manifestation due to unstable characteristics and **transient faults**, reflective of temporary environmental impact.

Ni et al. give extensive taxonomies of faults that cover definitions, causes, duration and impact of faults [6]. For example, calibration fault can be detected as the fault where the reported value is offset in some way from the ground truth. Bias and drift are primary causes of these faults. This fault is persistent, and it remains present throughout the deployment. However, data should not be discarded, since proper calibration formula can correct them. As opposed to this, faults occurring at hardware or connection level render data useless and should be disregarded. In a similar manner, spikes do not provide a meaningful information, and data should be disregarded. The difference is that spikes might occur only sporadically, while hardware error is persistent. It would make sense to keep the first node active, while the second should be deactivated.

We believe that the capability to distinguish between different types of faults would provide WSN with increased flexibility in handling faulty nodes. This flexibility can be achieved by learning from observations. In this way, a network could improve its behavior based on the study of its own experience. Given the goal in this work to recognize and classify different types of faults that occur on sensor nodes, we have chosen to focus on statistical learning, more specifically on statistical pattern recognition [12], [13].

The review of existing literature led us to focus on reliability of data-centric, densely deployed sensor networks. In this sense, we have formed a classification of faults in sensor readings independent of the cause of fault or the location of occurrence.

A. Fault Classification

In this work, we focus on faults that can be observed in readings through the effect they produce in data. This is a data-centric, diagnostic approach. Data features are statistical in nature, and a confident diagnosis of any single fault may require the use of more than one of those features. In dense, data-centric, networks readings are spatially and temporally correlated, so statistical patterns of readings can be used to identify and describe faults.

After analyzing existing classifications and underlying criteria, we have chosen to determine if faults can be recognized based on the pattern of behavior that they leave in the data. We propose a complete and consistent classification of faults in sensory data in terms of:

- Continuity of the occurrence
- Frequency of the occurrence
- Observable and learnable pattern

Criteria we have chosen draw from existing experience, and generalize and abstract existing criteria. We believe that a classification based on these criteria is flexible and applicable to a wide range of sensor readings. At the same time, it decouples models of faults from physical characteristics of a network and from the environment. Underlying cause of the error does not affect this classification. The focus of this work is on the pattern of fault occurrences on each sensor node.

If a sensor reading is represented with $r_i + \varepsilon_i$, where ε_i is a fault, we can define fault classification as follows:

- **Discontinuous** - Fault occurs from time to time, occurrence of ε_i is discrete.
  - **Malfunction** - Frequent occurrence of faulty readings, $\varepsilon_i > \tau$, where τ is threshold frequency. Also, there is no observable pattern in the fault occurrences.
  - **Random** - Infrequent occurrence of a faulty readings, $\varepsilon_i \leq \tau$.
- **Continuous** - After the certain point in time, a sensor returns constantly inaccurate readings, and it is possible to observe a pattern in the form of a function:

$$\varepsilon_i = f(t, [\alpha_1, \alpha_2 ....])$$

  - **Bias** - The function of the error is a constant, $\varepsilon_i = const$. This can be a positive or a negative offset.
  - **Drift** - The deviation of data follows a learnable function, such as polynomial change

$$\varepsilon_i = \alpha_1 * \varepsilon_{i-1}^n + \alpha_2 * \varepsilon_{i-1}^{n-1} + ...\alpha_0$$

Figure 1 illustrates the main concept of this classification. Knowing the type of the error provides a network with flexibility in handling faulty nodes appropriately, according to their type. Random faults can be smoothed out through data fusion while malfunctioning nodes can be switched off. Continuous faults are more interesting and challenging to handle in this case. This classification can provide a basis for the use of hypothesis finding techniques in order to learn a function of the fault and apply that function (model) to subsequent readings. Furthermore, models we learn can be used to update a priori set models of expected faults.

The classification process is a part of a possible framework illustrated in the Figure 2. This framework addresses a full cycle of modeling-detection-resiliency mechanism. However, this full cycle is out of the scope of this paper.

B. Proposed Method and Results

This work deals with the discovery of faults in sensor readings of data-centric WSN. One of the most significant metrics is accuracy, defined as the difference between resulting value and true value [2]. We rely on a neighborhood vote, keeping the assumption of spatial and temporal correlation of measurements. This means that we expect nodes in the same area to measure the same phenomena at the same time. The assumption holds for data-centric densely deployed WSN, as opposed to the use of saving mechanisms that turn off the network interface opportunistically which leads to intermittent connectivity and the formation of a Delay/Disruption

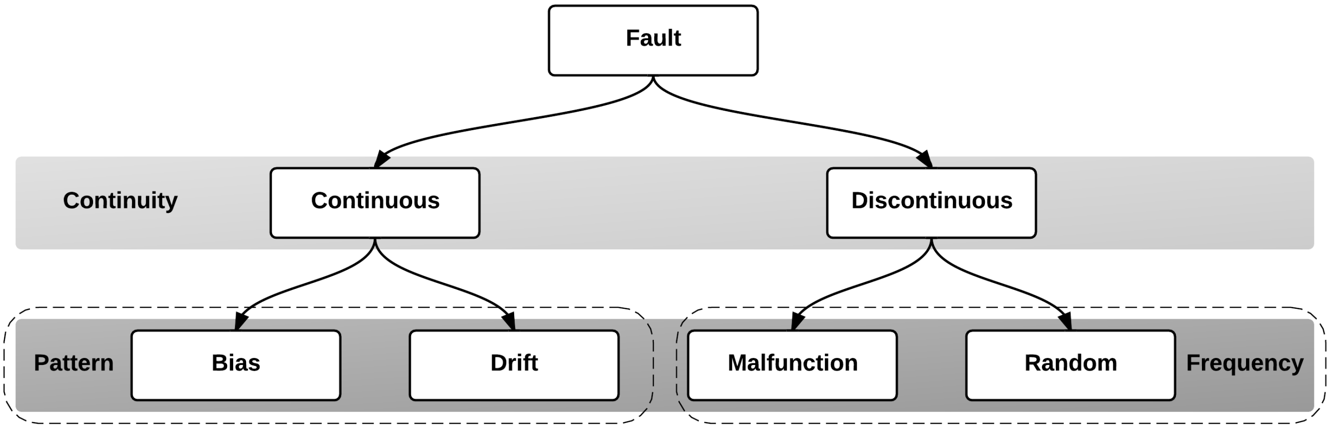


Figure 1. Classification of faults based on frequency, continuity and an observable pattern

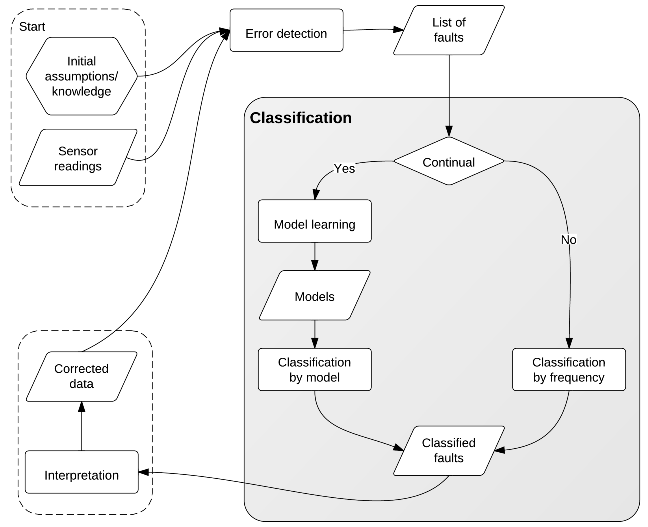


Figure 2. The process of fault handling with automated framework

Tolerant Network (DTN) [14]. An important characteristic of these networks is the lack of a simultaneous path between source and destination, while existing literature on routing in DTNs assumes that data packets are independent and uncorrelated with one another.

For the proposed method, it is necessary to assume the existence of a good connectivity between the nodes in the same area. Nodes are capable of clustering based either on the recognition of a nearby neighbors or an in-built detailed knowledge about the network layout, both physical and relational.

Every fault tolerant system has to decide between connectivity and communication overhead. Some of the existing work shows good results with on-node fault correction, with the benefit of flexibility and a quick response. On the other hand, this approach can cause a significant loss of information about a node's behavior. The network would always have to rely on corrected readings based on the prediction instead of on actual readings. In the centralized approach, the network becomes aware of the behavior of each node. When the network is capable of recognizing the type of fault, it can correct the readings and handle the node accordingly. Centralized approach does not cause any increase in communication overhead and provides sufficient resources for potential heavy computation requirements.

For the experiments, we have used the well known Intel Berkely laboratory dataset [15]. This set provides data on time and epoch of measurements, temperature, humidity, light and voltage. Dataset was split based on the location of sensor nodes, for example, a group of nodes with id 44 - 48.

In the first step, we analyze collected data in the time series to discover any discrepancies that might indicate the existence of faults. We are using statistical features of data for this purpose. For the initial calculations we rely on calculating the median, $\mu = \{r_{ij}\}_{\frac{1}{2}} = \tilde{r}$, of the group for the each epoch and chosen measurement, e.g. temperature. Median is chosen over the average as a more robust metrics. It is less likely to change drastically if extreme spikes in measurements occur on some of the nodes.

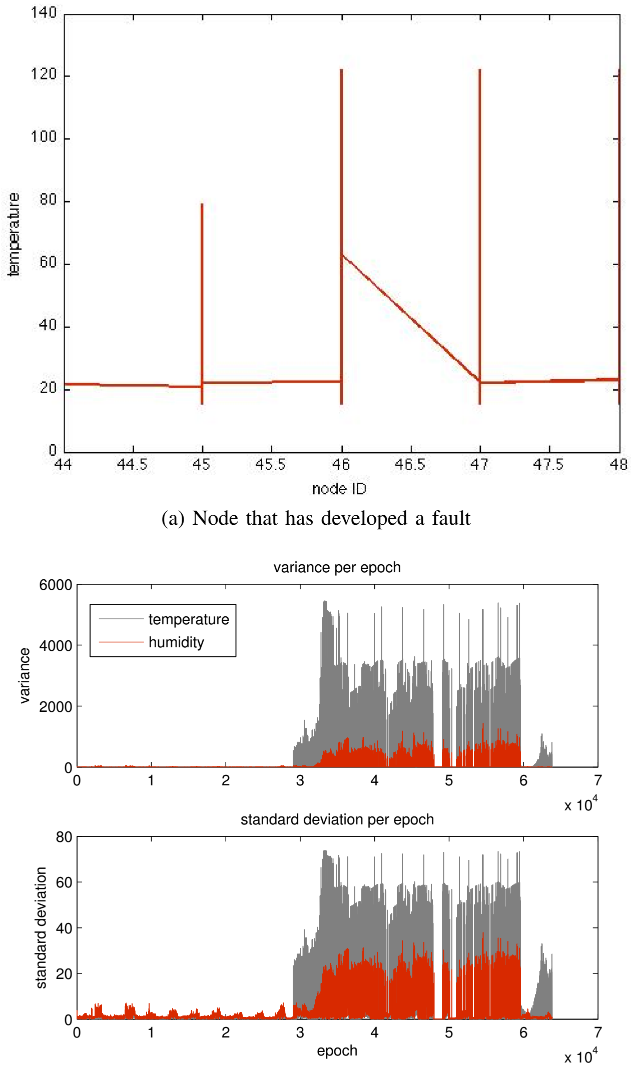
The difference between the reading and the median is calculated for each node, within the given tolerance rate:

$$|\tilde{r_i} - r_{ij}| \leq \tau * \tilde{r_i}$$

Tolerance rate τ largely depends on characteristics and the intended application of the network. For most of the experiments we have set it to $\tau = 0.2$, or 20% margin.

For the further analysis we use standard deviation $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(r_i - \tilde{r})^2$ and variance $\sigma = \sqrt{s^2}$

Figure 3 shows results of this analysis. We can see in Figure 3a which node has developed a fault, and in



(a) Node that has developed a fault

(b) Analysis to identify the time of possible fault development

Figure 3. Identifying nodes that have developed faults

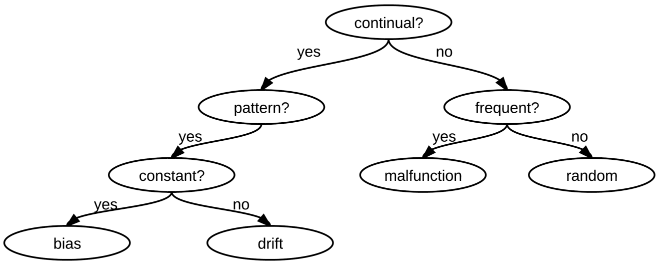


Figure 4. Decision tree

Figure 3b, we can see the time when the fault has started to develop. Also, here we can see that temperature and humidity measurements might be highly correlated. It might indicate that the node is becoming unreliable entirely.

Since we propose the use of a limited number of fault classes based on a small set of classification criteria, decision tree approach for pattern classification is a well fitting algorithm [16]. We can see this tree in the Figure 4. The outline of the process is given in the Algorithm 1.

Algorithm 1 Decision tree algorithm

```
Require: finite set of classes $C \in [\omega_1, \omega_2, \omega_3, \omega_4]$
  find the set of features $[f_1, f_2, ...., f_n]$
  for all data do
    analyze the time series
    check the continuity
    if $\varepsilon_i$ is discrete then
      check the frequency
      if $\varepsilon_i > \tau$ then
        malfunction
      end if
      if $\varepsilon_i \leq \tau$ then
        random
      end if
    end if
    if $\varepsilon_i$ is continuos then
      check the function $\varepsilon_i = f(t, [\alpha_1, \alpha_1 ....])$
      if $f(t, [\alpha_1, \alpha_1 ....]) = const$ then
        bias
      end if
      if $f(t, [\alpha_1, \alpha_1 ....]) \neq const$ then
        drift
      end if
    end if
  end for
```

Much of the work in designing trees focuses on on deciding which property test or query should be performed at each node. For this end, we have used MATLAB Treebagger function to run through various models of trees. This allows us to try many combinations and models of trees in a short time. Which combination will be suitable for a specific data set depends on many factors, and detailed discussion of this process is out of the scope of this paper.

Finally, in Figure 5, we can show at which point of time and which node has developed a drift with high probability.

At the current phase of the work, discovered fault is smoothed out. Since experiments are conducted on already collected dataset, there is no possibility to actually handle a faulty node in an adequate manner, and the comparison of the performance is left for the future work. Further development of the model requires experiments on simulated or deployed network, and it is also a part of future work.

Our experiments show around 15-20% occurrence of faults within the test data, which is consistent with other related experiments on the same dataset [7]. The method was able to correctly detect faults in approximately 90% of fault occurrences. These results are average from several

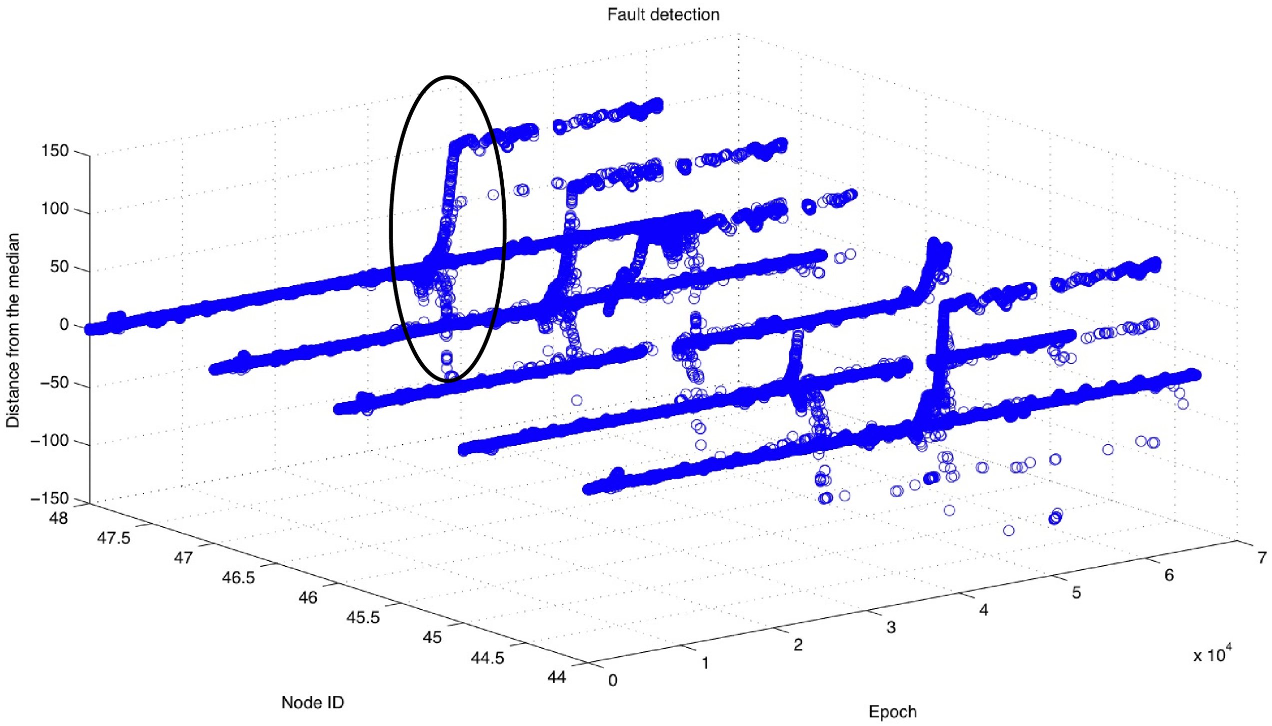


Figure 5. Drift development

runs on the same dataset. Minor differences occur due to the nature of Treebagger function which splits data into training and test set differently in each run.

At the moment, the method focuses on detecting the manifestation of faults in data, without looking into the underlying cause. However, a proper classification of a fault might indicate that cause. For example, malfunction is probably caused by an error in hardware, while drift might be caused by low battery levels or miscalibration. This provides for a flexible way of dealing with each node in an appropriate manner.

IV. Conclusion and Future Work

In this work, we focused efforts on discovering faults in sensor readings within a data-centric dense wireless sensor network. The focus is on the pattern of the occurrences of the faults on the node, regardless of the underlying cause. Key concept is a **classification of faults** based on **frequency, continuity and observable pattern** in the faults. We keep the assumption of spatial and temporal correlation between sensor readings. Also, we use centralized approach in order to ensure computational power with low communication overhead. In this way, the network is aware of when and where faults have developed, and it can handle a faulty node accordingly.

Statistical pattern recognition using decision tree method and classification work comparatively well for the data-centric and dense networks. However, if we can not assume density or spatial and temporal correlation of readings, this method is insufficient. It requires further development of a priori fault modeling to provide a basis for comparison of correct readings against faulty readings.

Providing an economic way of handling faults at the level of a single sensor is difficult. We believe that proposed classification, together with the detection method, provides a way to combine benefits of the centralized approach, such as bigger computational power with the flexibility of on-node correction. The network would be able to recognize fault on each node, handle it appropriately and keep the communication overhead low.

We have conducted experiments on Intel Berkeley dataset. So far they confirm that the method has a potential for flexible handling of faults in WSN. However, we plan to expand the experiments to use more datasets from different settings (urban and outdoor deployments) in order to test the hypothesis about spatial and temporal correlation.

Next, we plan to focus on a priori fault modeling and inclusion of developed models in the detection and classification phase. This would allow lesser dependence on neighboring nodes and stable connectivity.

To ensure the continually reliable operation, WSN has to be capable to address the full cycle of fault tolerance, which includes recognition of a faulty node, elimination or counteracting of its readings and self calibration where applicable. To meet this goal, we plan to expand proposed method by introducing hypothesis finding techniques to discover fault models from the data. This is especially interesting for the continuos errors, such as drift or bias. If the network can learn these models, it can use them for correction of faults. It would enable the network to adapt its behavior and structure based on experience.

Finally, we would like to address some common issues that are often encountered when developing algorithms for WSN in the real world [17]. These issues include the assumption of a reliable communication, precise analysis of energy consumption in terms of communication and computation, synchronicity of readings and the assumption of a stable set of neighbors for each sensor node.

We believe that proposed method is viable for use in dense data-centric network setting with assumed good knowledge of network's topology. More importantly it provides a basis for further improvements and development of a complete scheme for self-repairing wireless sensor networks, which is the ultimate goal of the ongoing research.

Acknowledgment

The authors would like to thank A. Klein, F. Wagner, S. Toriumi, Y. Baba and F. Ishikawa, all of National Institute of Informatics, Japan, for their invaluable feedback. We would like to extend a special thanks for extensive discussion and support to G. Bourgne, also of National Institute of Informatics at the time.

References

[1] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-vincentelli, "Fault tolerance in wireless sensor networks, " book chapter," in *in Handbook of Sensor Networks, I. Mahgoub and M. Ilyas*, 2004.

[2] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2007.

[3] A. Hac, *Wireless Sensor Network Designs*. John Wiley & Sons, 2004.

[4] J. Shell, S. Coupland, and E. Goodyer, "Fuzzy data fusion for fault detection in wireless sensor networks," in *Computational Intelligence (UKCI), 2010 UK Workshop on*, September 2010, pp. 1 –6.

[5] M. Takruri, S. Challa, and R. Yunis, "Data fusion techniques for auto calibration in wireless sensor networks," in *Information Fusion, 2009. FUSION '09. 12th International Conference on*, July 2009, pp. 132 –139.

[6] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, "Sensor network data fault types," *ACM Trans. Sen. Netw.*, vol. 5, no. 3, pp. 25:1–25:29, June 2009.

[7] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor faults: Detection methods and prevalence in real-world datasets," *ACM Transactions on Sensor Networks*, vol. 6, no. 3, pp. 1–39, June 2010.

[8] Y. Yao, A. Sharma, L. Golubchik, and R. Govindan, "Online anomaly detection for sensor systems: A simple and efficient approach," *Perform. Eval.*, vol. 67, pp. 1059–1075, November 2010.

[9] R. Tiwari and M. T. Thai, "On enhancing fault tolerance of virtual backbone in a wireless sensor network with unidirectional links," in *Sensors: Theory, Algorithms, and Applications*, ser. Springer Optimization and Its Applications, V. L. Boginski, C. W. Commander, P. M. Pardalos, and Y. Ye, Eds. Springer New York, 2012, vol. 61, pp. 3–18.

[10] G. Huang, Y. Zhang, J. He, and J. Cao, "Fault tolerance in data gathering wireless sensor networks," *The Computer Journal*, vol. 54, no. 6, pp. 976–987, 2011.

[11] R. Alena, R. Gilstrap, J. Baldwin, T. Stone, and P. Wilson, "Fault tolerance in zigbee wireless sensor networks," in *Proceedings of the 2011 IEEE Aerospace Conference*, ser. AERO '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1–15.

[12] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Pearson Education, 2003.

[13] S. Marsland, *Machine Learning: An Algorithmic Perspective (Chapman & Hall/Crc Machine Learning & Pattern Recognition)*, 1st ed. Chapman and Hall/CRC, April 2009.

[14] F. C. Choo, P. V. Seshadri, and M. C. Chan, "Application-Aware Disruption Tolerant Network," in *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*. IEEE, October 2011, pp. 1–6.

[15] I. B. Laboratory. (2012, June) Intel berkely dataset. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.htmlf

[16] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*, 2nd ed. Wiley-Interscience, November 2001.

[17] G. Halkes and K. Langendoen, "Practical considerations for wireless sensor network algorithms," *Wireless Sensor Network*, vol. 2, no. 6, pp. 441–446, June 2010.

EDGM: Energy Efficient Data Gathering with Data Mules in Wireless Sensor Networks

Nour Brinis
CRISTAL laboratory
National School of Computer Science
University of Manouba, 2010 Tunisia
brinis.nour@gmail.com

Pascale Minet
INRIA Rocquencourt
78153 Le Chesnay cedex, France
pascale.minet@inria.fr

Leila Azouz Saidane
CRISTAL laboratory
National School of Computer Science
University of Manouba, 2010 Tunisia
leila.saidane@ensi.rnu.tn

***Abstract*—This paper deals with prolonging the lifetime of a wireless sensor network by exploiting mobility. We consider data gathering applications where each event occurring in the monitored area must be reported to the sink. The target applications are assumed to be delay tolerant. We define the network lifetime as the time until coverage loss. We propose an energy efficient strategy to collect data from sensor nodes, using data mules. This strategy has the advantage of not requiring network connectivity. Data mules are responsible to carry data to the sink, to schedule sensor nodes activities and to replace energy constrained nodes. They take advantage of energy harvesting, from a generating power terminal, in order to achieve their mission. We simulate the proposed data collection strategy with the NS2 simulator and derive the network lifetime, the energy consumption of sensor nodes and the rate of collected data meeting a given delay. Furthermore, this study leads us to determine the optimal number of data mules needed to meet a given delivery delay deadline.**

Keywords-wireless sensor networks; energy efficiency; data mule; node activity scheduling; delay tolerant network

I. Context and motivation

Wireless Sensor Networks (WSNs) play an important role in different applications, such as environment monitoring, health treatment, space exploration and others [1]. A WSN is composed of a large number of sensor nodes. Each one is characterized by three basic capabilities: sensing, wireless communication and computation. These sensor nodes are usually battery-operated. So, their power must be used very sparingly in order to fulfill the underlined application requirements before batteries depletion. Furthermore, each sensor application requires specific quality of service: delivery delay deadline, data accuracy, etc. Hence, minimizing energy consumption, while meeting required performance constraints of the target application, is a very important challenge to prolong network lifetime.

Sensor nodes are usually densely deployed in order to ensure the total connectivity and coverage of the interest area. However, connectivity may be lost quickly because some nodes, such that located in the proximity of the sink, consume more energy for relaying traffic originated from other farther nodes. Also, the variability of the sensors distance to the sink involves a variability of the data delivery delay. Thus, delivering data to the sink by multi-hop forwardings may lead to connectivity loss in addition to an unfair data delivery delay. Moreover, a complete coverage of an area implies connectivity among the nodes only if the radio range is at least twice of the sensing range [2]. However, if the radio range is too large as compared to the sensing range, the network may be subject to excessive radio interference although its connectivity is ensured [2]. That is why, solutions that do not require connectivity must be investigated in order to preserve data delivery to the sink.

In this paper, we focus on the problem of delivering data to the sink while conserving sensor nodes energy, in a WSN totally covered and not necessarily fully connected. Our proposed strategy is based on the use of data mules for gathering data from sensor nodes that are assumed to be static. A data mule is a kind of mobile robot able to communicate with sensor nodes and to carry the generated traffic to the sink. Our solution targets especially delay tolerant applications, requiring a long network lifetime. We define the network lifetime as the time of the first coverage hole creation. we propose a gathering strategy that extends clearly the network lifetime. Indeed, the data mule arrivals are predictable. So, static sensor nodes wake up at well-defined periods, without need to deplete energy for listening data mules polling messages. In addition, static sensor nodes transmit data only at single hop ranges. Our solution combines the data gathering with a maintaining coverage strategy. For this purpose, constrained energy sensor nodes may be replaced by other redundant nodes in order to maintain the whole network coverage.

The performance evaluation of our solution allows us to obtain the rate of delivered data meeting a given delivery delay and the required number of data mules that guarantee this deadline.

The paper is organized as follows. In Section II, we briefly present the state of the art related to data gathering strategies. Afterwards, we define our data collection scheme by describing sensor nodes and data mules behaviors in Section III. In Section IV, we evaluate the proposed solution

ISBN: 978-1-61208-207-3

and compare it with a double range data gathering strategy in terms of energy consumption and data delivery delay. Finally, we conclude the paper and give some directions for our future works in Section V.

II. Related work

Exploiting mobility for data gathering and routing has been studied in many papers, using various methods. These solutions aim at extending the network lifetime by conserving sensor nodes energy. They can be classified according to the network mobility degree into Networks with mobile sensors and Networks with additional mobile agents.

In mobile WSNs, where all sensor nodes have motion capabilities, the connectivity between mobile nodes is poor, and thus it is difficult to guarantee an end-to-end connections from sensor nodes to the sinks. The main issue is to detect the occasional connectivity between mobile nodes in order to transmit sensed data to the sink. This problem has been treated in ZebraNet [3] project. This project relies on equipping zebras with sensor nodes in order to record the animal position or other relevant sensor readings. For this purpose, zebras wander randomly in the area and send data to the sink when they arrive to its transmission range. Due to their random movement, delay latency can not be bounded and in worst case data can never been send to the sink when the tagged zebra remains far away. So, in order to reduce the data delivery delay, a history-based routing approach has been proposed in [4]. Each sensor node records its past success rate of directly transmitting data packets to the sink. When a sensor meets another sensor, the former transmits data packets to the latter if the latter has a higher success rate.

In static WSNs, having mobile sinks, sinks move towards static sensor nodes in order to collect sensed data. This approach was studied in [5][6]. Otherwise, Additional agents move around static sensor nodes to gather data and carry them to static sinks. In both cases, mobile nodes should gather data via short range communication in order to minimize the energy consumption, while assuring a fairly data delivery to the sink. The data gathering strategies can be classified into: direct-contact data collection and cluster based data collection.

Many applications, especially these deployed for statistics measurement, are more sensitive in terms of network lifetime than that in terms of delivery delay: they are delay tolerant. For this kind of applications, direct-contact data collection is more suitable. The problem has been dealt with in [7] using animal-based mobile collectors. In that case, mobile collectors wander randomly in the interest area to gather data from static sensor nodes. Energy consumption at sensor side is only due to mobile collector discovery and subsequent data transmission. Assume that a mobile collector periodically broadcasts a beacon message while moving. It is very expensive to detect the mobile collector arrival by keeping listening to the beacon message. Moreover, because of the random movement of mobile collectors, the static sensor nodes are not ensured to deliver their sensed data within a bounded delay. Studies in [8] show that, if the mobile collectors move along regular trajectory, then sensors can predict their arrival and so, the network lifetime is improved. One of the regular gathering trajectories that guarantee a single hop data gathering with extreme energy saving is that the mobile collector gathers data packets by sequentially visiting each sensor. This problem is referred in sparse sensor networks to the Travelling salesman problem TSP. In order to shorter the tour length while conserving a single hop transmission, data collectors visit the transmission range of each node in order to gather data. For this purpose, an alternative solution of TSP called TSP neighboring was been proposed in [9]. The above mentioned strategies reduce greatly the energy consumption for data transmissions, however, the data latency increases because of the tour length.

In order to shorten the data delivery latency, most of the previous work relay on introducing multi-hop relays via clustering definition. These strategies aim at retrieving a trade off between network lifetime and delivery time. In these approaches, a mobile collector moves around a subset of sensors, cluster heads, to gather data by local multi-hop communication. By this way, the tour length is shortened and thus the data delivery latency decreases. the main encountered issue is the selection of the appropriate clusters according to a given criteria. Studies in [10] propose a cluster definition according to the definition of the connectivity islands (partitions) of the network. Then, the cluster head is chosen in the middle of the partition and having the maximum residual energy. Studies in [11] presents a bounded relay hop data gathering strategy. An algorithm, called SPT-DCA, selects appropriate nodes that will aggregate captured data transfered by bounded hop transmissions. then, the mobile collector visits the selected nodes to gather data, using the TSP heuristic. SPT-DCA is based on a prior Spanning Tree resolution of the whole network graph. Then, the appropriate cluster head nodes are selected according to a fixed relay hop bound.

Most of the previous data gathering studies deal with extending the network lifetime in a sparsely sensor network. They assume that all the sensors are working simultaneously during the whole process of data collection. Thus, the coverage problem was rarely addressed in the data gathering solutions. Two issues are considered, in this paper, in order to improve the network lifetime: a single hop gathering strategy using data mules, combined with a mechanism for the replacement of some critical nodes before depleting their energy. The goal of our solution is to determine the optimal number of data mules required to extend the covered network lifetime while meeting a given data delivery delay deadline. In the following sections, we detail our proposal.

III. EDGM: Data Collection Scheme

Our solution, called EDGM for Energy efficient Data Gathering with data Mules, is characterized by:

- All sensor nodes in the network are assumed to be static. They are denoted *SN*.
- Each sensor node has two independent components: sensing and communication units. Both units are powered from the same limited source of power (battery).
- The data mule or mobile node, denoted *DM*, is able to recharge itself at a generating energy terminal, called energy terminal. We assume that this energy terminal is located at the proximity of the sink.
- Each sensor node and each *DM* are able to compute their residual energy.
- Each sensor node and each *DM* have a memory sufficient to store the collected data.
- The transmission range of each sensor node is denoted r_c. We suppose that all nodes have the same communication range. Otherwise, r_c denodes the minimal transmission range among all deployed nodes.
- The transmission range of each data mule is denoted *R*. The constraint that must be met by *R* is given by Inequation 1 in Section III-C1.
- For simplification purpose, we assume that recharging a *DM* is sufficient to allow it to move, communicate and collect data before turning back to the energy terminal.

A. Principle of the solution

We now explain how each mobile node collects the sensed data. The goal of each sensor node is restricted to capture data and transmit them to the sink, if this node is located in its transmission range, or to a *DM*, when it is one hop away. In other words, sensor nodes do not relay the traffic generated by other nodes. Two main issues must be solved: the trajectory of the *DMs* and the behavior of the sensor nodes. At the proximity of each static node, a *DM* stops at a Break Point, denoted *BPoint*, for a defined Break Period, denoted *BPeriod*, and sends a *HELLO* message. When it arrives at the proximity of the sink and the energy terminal, a *DM* sends the collected data to the sink and recharges itself. Each sensor node *SN* is associated to a unique *BPoint*. The break point positions are defined in Section 3.3. Each *SN* is awake only during its break period to send its captured data. Then, it turns off its communication unit. During the break period, since several sensor nodes are authorized to send data to the *DM*, collisions may occur. A deterministic medium access scheduling is set by the *DM*, in order to avoid the collision problem. More precisely, each node starts sending its sensed data when it receives an invitation delivered by a *DM* through an *INVIT* message which contains the medium access scheduling. Figure 1 summarizes the data collection strategy. During the first round, each *SN* receives the information needed to synchronize itself with the *DMs* present in the considered area. Once it is synchronized, it keeps awake only during a *DM* break period in its associated *BPoint*. During this time, it is authorized to send its collected data through an *INVIT* message. Detailed presentation of the sensor nodes and the *DM* behaviors is given in Section III-B and III-C respectively.

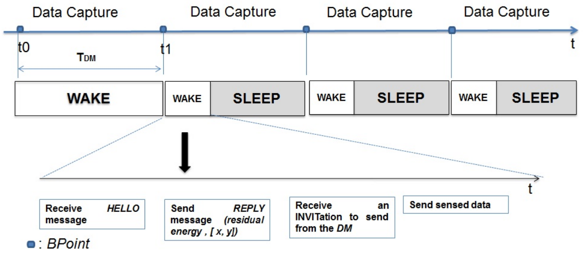


Figure 1. Sensor activities.

B. Sensor Node Behavior

The energy consumed by a node depends on its state. We distinguish the four following states, presented in ascending order of energy consumption:

- *Off*: all sensor units are turned off,
- *Sleep*: only the sensing unit remains active,
- *Receive*: all sensor units are active and the communication unit is receiving a message,
- *Transmit*: all sensor units are active and the communication unit is transmitting a message.

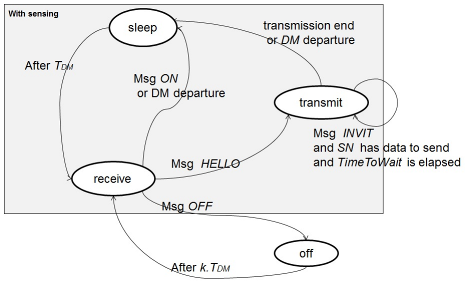


Figure 2. Sensor activities scheduling model.

During the first round of each *DM*, each sensor node remains awake in order to memorize the *DM* round duration, denoted T_{DM}. T_{DM} represents the period between the first and the second received *HELLO* message from the same DM (see for instance t_1 - t_0 in Figure 1). Any sensor node, that is not redundant, will be awake, during a break period, starting when the *DM* stops at its associated break point. Each T_{DM} period, *SN* switches to the *receive* state and processes the message received from *DM*, as described in Figure 2:

- If an *ON* message is received: *SN* switches to the *sleep* state and will switch to the *receive* state for the next arrival of *DM*.

- If an *OFF* message is received: *SN* switches to the *off* state for the *k* following rounds, with *k* a positive integer.
- If a *HELLO* message is received: *SN* switches to the *transmit* state and sends a *REPLY* message.
- If an *INVIT* message is received: *SN* analyses the received message in order to determine the medium access scheduling. It waits its turn, for a *TimeToWait* period, before beginning its data transmission. When this time is elapsed, *SN* starts sending data for a maximum predefined period (*PeriodToSend*). The *TimeToWait* and *PeriodToSend* values are included in the *INVIT* message.

Once *SN* is in the *transmit* state and after a random period, it sends a REPLY message in which it specifies its residual energy and its location. Since each *SN* is static, it sends its location information only in the first round. After that and in all *DMs* rounds, it waits for an *INVIT* message to send its data. It switches to *sleep* state at the *DM* departure or when it ends its data transmission.

Once *SN* is in the *sleep* state, it wakes up after T_{DM} and it switches to the *receive* state.

From the *off* state, a redundant *SN* switches to the *receive* state after $k.T_{DM}$ time units.

C. Data Mule Behavior

In this section, we describe the *DM* behavior. We first define the *DM* trajectory. After that, we present the *DM* behavior at each break point, and the communication between *DMs*. Notice that all *DMs* follow the same trajectory, but with different starting points.

1) DM trajectory : Three conditions must be satisfied for the choice of the *DM* trajectory:

- C1: all static nodes are explored;
- C2: the trajectory length, before turning back to the sink, is minimized;
- C3: the number of break points is minimized.

This problem is reduced to a 2-dimension geometry problem: how to tile a rectangular area *(X,Y)* with a minimum number of disks of radius r_c where each disk center represents a break point of *DMs*? This problem has been solved in [12]. The optimal disks positions are such that the tops of equilateral triangles of edge length $\sqrt{3}.r_c$, are the disk centers. By this way, we ensure the exploration of all static sensor nodes by visiting a break point in their transmission range (Conditions C1 and C3 are satisfied). So, knowing the r_c value, each DM computes the next breakpoint location. In order to minimize the delivery delay needed to transfer gathered data to the sink node, *DM* follows the trajectory visiting first only the break points at even lines, and then that at odd lines, as illustrated in Figure 3 (Condition C2 is satisfied). Besides, when a *DM* meets another *DM*, by receiving its *HELLO* message, the *DM* that moves towards the sink collects all data gathered by the other *DM*. By this way, we reduce the data delivery delay. For this purpose, we use the following assumption:

$$R \geq \sqrt{3}.r_c \qquad (1)$$

The first location of *DMs* are chosen to ensure fairness between all sensor nodes in terms of delivery delay. So, the initial positions of the *DMs* are chosen as follows:

Let *n* be the number of *DMs* in the rectangular area *(X,Y)*, if *n* is even, two *DMs* are set each $\frac{2.Y}{n}$. Otherwise, we set two *DMs* each $\frac{2.Y}{n+1}$. These two *DMs* will move in opposite directions. The remaining *DMs* are set at the first *BPoint* of the extremity lines of the trajectory.

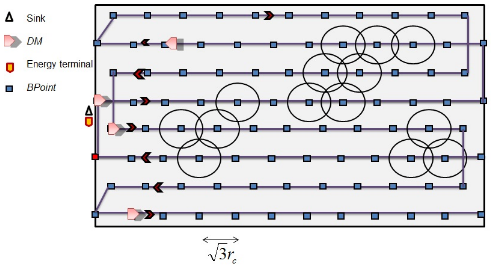


Figure 3. *DM* trajectory.

2) DM behavior at a break point: Stopping at a specified *BPoint*, a *DM* acts as the leader for all the sensors associated to this *BPoint*. When it receives the *REPLY* message from different sensor nodes, it decides whether the sender node is redundant or not, it replaces the not redundant node having critical energy level and it schedules the transmission of the active not redundant nodes. With the knowledge of the *SNs* location sent in the *REPLY* message, *DM* decides whether the sender node is redundant through the algorithm presented in [13]: knowing the positions of all active neighboring nodes, *DM* examines if any semicircle within the interest area of the sender node is empty. In that case, the sender node is not redundant because a portion of its area is covered only by it. In addition, when the residual energy level incorporated in the *REPLY* message is critical, the *DM* sends an *ON* message to the neighboring nodes of the sender node one by one, and each time it checks if the energy constrained node become redundant. In that case, it stops the replacement algorithm; the sender node is replaced. Moreover, using the residual energy sent in each *REPLY* message, *DM* specifies, in the *INVIT* message, the time for each node to wait before starting transmission (*TimeToWait*) and the transmission duration (*PeriodToSend*). Sensors having less residual energy are invited to transmit first. Knowing the number of sensor nodes associated with this *BPoint*, *DM* assigns a fair duration to each node for sending its sensed data.

Table I. Simulation parameters.

Network configuration	Network area	(300m X 300m)
	Number of data mules	1 to 4
	Range (*SN*)	50m
	Range (*DM*)	100m
	Range (sink)	50m
	Bandwidth	2Mbps
	Mac protocol	802.11
	DM(s) speed	1m/s
	BPeriod	15s
	k	3
Traffic parameters	Type	Periodic Data
	Packet size	256 bits
	Throughput	16 Kbps
	DM(s) Buffer size	100 packets
Energy model	Type	Battery
	Initial Energy(*SN*)	15 Joules
	Transmit power	0.36 Watt
	Receive power	0.24 Watt
	Sensing power	0.015 Watt

IV. Performance evaluation

In this section, we present simulation results to evaluate the efficiency of the proposed EDGM strategy. We carried out several simulations using NS2 simulator and selected modules of the pre-implemented MannaSim project [14]. MannaSim extends the NS2 simulator by introducing WSN specifities: battery energy resource, data generator framework...

A. Evaluation metrics

Our simulation analysis emphasizes on the following performance metrics:

- The energy consumption rate of a node *i*, denoted *ConsumedEnergy*$_i$, is defined by (2) where *InitEnergy*$_i$ denotes the initial energy of node *i* and *Energy*$_i$ is the residual energy of node *i*.

$$ConsumedEnergy_i = \frac{InitEnergy_i - Energy_i}{InitEnergy_i} * 100 \quad (2)$$

- The average data delivery delay, denoted *AvgDeliveryDelay*, is defined as the sum of packet delivery delays divided by the number of received packets (see (3)):

$$AvgDeliveryDelay = \frac{Sum\_of\_all\_Pkt\_Delays}{Number\_of\_Received\_Pkts} \quad (3)$$

- The rate of packets, received before a given delay deadline, among the total number of received packets, denoted *MeetDDDRate*, is defined by (4). The Data Delivery Deadline (*DDD*) is the duration elapsed from data generation to data reception by the sink:

$$MeetDDDRate = \frac{Received\_Pkts\_Meeting\_DDD}{Total\_Received\_Pkts} \quad (4)$$

B. Simulation results

Table I summarizes the simulation scenarii parameters.

We suppose that each node has a transmission range equal to its sensing range, denoted *Range* in Table I.

Firstly, we consider a topology, where 2 *DMs* are used, 16 sensor nodes are deployed to ensure coverage, and 134 redundant sensor nodes are deployed randomly. Notice that the node having identifier 0 is the sink and the nodes 1 and 2 are the *DMs*. Nodes 3 to 18 constitute the 16 nodes needed to ensure coverage. Nodes, having an identifier strictly higher than 18, are initially redundant. Figure 4 visualizes the useful and the total energy consumption rate versus the sensor node identifiers. The useful energy refers to the amount of energy really required to capture and transmit the monitored events. We suppose that the near death energy is 2 Joules. So, all nodes consuming more than 86% of their energy tend to be out of service and the appropriate nodes have been selected to cover the coverage hole before its creation. We observe that energy consumption is clearly minimized for redundant nodes that are mostly kept in off state, starting from the second round. Some nodes, such as 31, 38, 54 and 69, waste more energy than other initially redundant nodes because they have been selected to replace energy constrained nodes. We also notice that the gap between the useful and the total energy is not too significant, due to the frequent turning off of the radio component of each node.

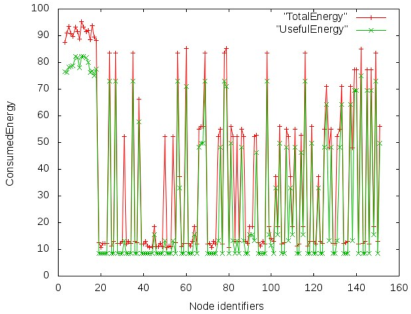


Figure 4. energy consumption rate.

Figure 5 depicts the network lifetime when the total number of nodes varies from 20 to 150 to represent different node density. For each new measurement, we add randomly the appropriate number of nodes on those already deployed. We notice that less *DMs* used leads to longer network lifetime since static sensors can turn frequently to powesaving mode. It was also shown that more redundant nodes are deployed, latter is the first coverage hole creation. This is reasonable because more redundant sensors would provide more opportunities to replace the nodes having critical energy level. However, when the number of used *DMs* is larger, the number of attempts to replace nodes exhausting their energy increases and so, the replacement success rate increases. Thus, the ability to replace a node has more effect

ISBN: 978-1-61208-207-3

on the network lifetime when the number of redundant nodes is larger.

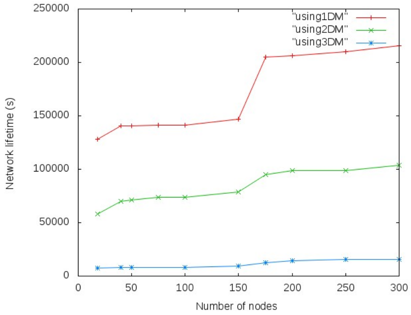


Figure 5. Network lifetime.

Now, we conduct similar simulations with the same configuration, but using 300 static nodes, each one having an initial energy value 30 Joules.

Figure 6 shows the average delivery delay when the square sensing field varies from 300m to 800m, for different number of *DMs*. Shorter data gathering time appears using more data mules on a small sensing field since the tour length is shorter. In addition, the effect of exchanging data between encountered *DMs* appears clearly when the network area is larger. However, the data delivery delay gain become more acute as the number of *DMs* increases, this is due to the concurrent use of multiple *DMs* and the more frequently *DMs* meetings during their tours.

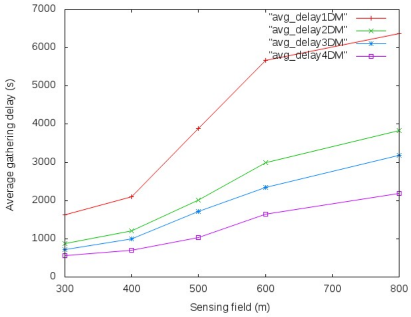


Figure 6. Average data delivery delay.

Table II presents the average delivery delay, denoted *Avg-delay*, and the corresponding consumed energy, denoted *Avg-energy*, on a square area of side length 600m. The consumed energy is the average of the amount of energy consumed in each round of the DMs, starting from the second round. the SPT-DCA strategy uses one mobile collector. We consider that the relay hop bound is 2 hops. We notice that SPT-DCA offers a better data delivery delay in spite of a significant consumed energy. This is due to the local double hop transmission range. Some nodes will relay trafic generated by other sensors. SPT-DCA strategy demonstrates a data delivery delay quite higher of the EDGM strategy using 2 DMs, wile consuming about 18% more energy.

Table II. Simulation results.

Data gathering scheme	Avg-energy	Avg-delay(s)
SPT-DCA	29%	3478.35
EDGM (using 1 DM)	07%	5668,20
EDGM (using 2 DMs)	11%	2995,63
EDGM (using 3 DMs)	12%	2345,19
EDGM (using 4 DMs)	17%	1647,99

Figure 7 plots the rate of packets, meeting a given delivery delay, among all the received packets, on a square area of side length 600m. Through this result, we can determine the number of *DMs* needed to meet a given deadline. For example, for an application having $DDD = 5200s$, using two *DMs* is acceptable to guarantee data delivery to the sink before the deadline. However, using this number of *DMs* is unacceptable for an application having $DDD = 2000s$, this means that some data may arrive to the sink after the deadline. In this case, four *DMs* are required.

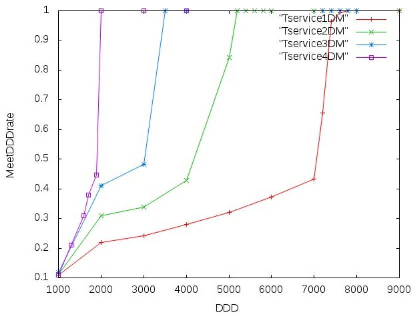


Figure 7. Rate of received packets meeting a given *DDD*.

V. Conclusion and future work

Energy efficient data collection strategies in wireless sensor networks constitute a challenging research domain. In this paper, we proposed a data collection strategy, using data mules. We target especially delay tolerant monitoring applications, requiring a long lifetime. The originality of our contribution consists of joining the coverage and data collection problems. Previous data gathering studies have mainly focused on sparsely deployed sensor networks. Our strategy aims at preserving the coverage property of the sensor network. Indeed, only selected nodes, that are mandatory to ensure coverage, capture data and send them to a single hop node. To validate the proposed solution, we conducted extensive simulations and analyzed the network lifetime and the rate of packets meeting a given delivery delay deadline. Hence, we determine the number of data mules needed to meet a given delivery delay. Larger is the monitoring area leads to more required data mules. Therefore, our future work consists of exploiting the eventual fully connected partition of the sensor network to retrieve a tradeoff between the delivery delay deadline and the number of required data mules. In addition, we are developing an analytical model for predicting the network lifetime obtained by our proposal.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, *Wireless sensor networks: A survey*. Computer Networks, vol. 38, no. 4, March 2002.

[2] H. Zhang, J. C. Hou, *Maintaining sensing coverage and connectivity in large sensor networks*. Technical Report UIUCDCS-R, June 2003.

[3] The ZebraNet Wildlife Tracker. *http://www.princeton.edu/mrm/zebranet.html* [retrieved: April, 2011].

[4] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. *Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet*. In Proc. of conference ASPLOS, 2002.

[5] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, *Exploiting sink mobility for maximizing sensor networks lifetime*. in Proc. 38th HICSS, Big Island, HI, Jan. 2005.

[6] Sh. R. Gandham, M. Dawande, R. Prakash and S. Venkatesan. *Energy-Efficient Schemes for Wireless Sensor Networks with Multiple Mobile Base Stations*. Proceedings of IEEE Globecom, December 2003.

[7] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, *Exploiting mobility for energy efficient data collection in wireless sensor networks*. Mobile Netw. Appl. (MONET), vol. 11, no. 3, pp. 327-339, Jun. 2006.

[8] A. Chakrabarti, A. Sabharwal, and B. Aazhang. *Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks*. In Proc. of IPSN, vol. 2634 of LNCS, pp. 129-145, 2003.

[9] S. Nesamony, M. K. Vairamuthu, and M. E. Orlowska.*On Optimal Route of a Calibrating Mobile Sink in a Wireless Sensor Network*. In Proc. of INSS, pp. 61-64, 2007.

[10] M. B. Soares et al. *Hybrid Mobile Robot Navigational Strategy for Efficient Data Collection in Sparsely Deployed Sensor Networks*. In Proc. of IROS, pp. 2833-2838, 2007.

[11] Miao Zhao and Yuanyuan Yang.*Bounded relay hop mobile data gathering in wireless sensor networks*. Mobile Adhoc and Sensor Systems MASS '09, 2009.

[12] R. E. Tarjan. *Depth-First Search and Linear Graph Algorithms*. SIAM Journal on Computing, pp. 146-160, 1972.

[13] Z. Butler and D. Rus, *Event-based motion control for mobile-sensor networks*. IEEE Pervasive Computing, vol. 2, no. 4, pp. 34-42, 2003.

[14] Mannasim. *http://www.mannasim.dcc.ufmg.br* [retrieved: Nov, 2011].

Design and Analysis of Almost-Always-Sleeping Schedulers for Embedded Systems

Biswajit Mazumder, Hao Jiang, and Jason O. Hallstrom
School of Computing
Clemson University
Clemson, SC 29634, USA
{bmazumd, hjiang, jasonoh}@cs.clemson.edu

***Abstract*—Limited energy resources dictate the design of many embedded applications composed of small, modular tasks, scheduled periodically. In this model, the embedded device wakes, executes a task-set, and returns to sleep. These systems spend most of their time in a state of deep sleep to minimize power consumption. We refer to these systems as *almost-always-sleeping* (AAS) systems. In this paper, we describe a series of task schedulers for AAS systems designed to maximize sleep time. We consider four scheduler designs, model their performance, and present detailed performance analysis results under varying load conditions. This is the first systematic analysis of this important class of schedulers.**

***Keywords*—Wireless sensor networks; scheduling; power consumption.**

I. Introduction

A significant class of embedded applications are characterized by low duty-cycle operation and time-triggered, periodic execution. These systems sleep for relatively long periods, wake in response to a timer interrupt, perform a short computation, and return to sleep. We refer to these systems as *almost-always-sleeping* (AAS) systems. The wireless sensor network domain is rife with representative examples. Environmental monitoring networks [12], [17], [18], for instance, comprise distributed sensors that periodically wake to collect and transmit environmental stimuli before returning to sleep. Indeed, nearly *every* sensing system adopts a variant of this strategy, as do numerous other embedded applications.

The broad adoption of AAS designs is due to the energy efficiency they afford. Modern microcontrollers (MCUs) support sleep states in which internal circuitry may be powered-down, reducing energy consumption by several orders of magnitude. As an example, common wireless sensor networking platforms consume 10s of *milli*watts in the active state, and only 10s of *micro*watts when idle [13]. For devices that exhibit this two-phase consumption profile, the best conservation strategy is to sleep as often as possible.

The active period of an embedded device is partitioned into two components: the time spent executing application code (*tasks*), and the time spent executing scheduling code. Reducing the runtime of individual tasks can only be achieved on an application-by-application basis. Reducing the scheduling overhead, however, can be achieved through careful analysis and design of the underlying scheduling system – our focus.

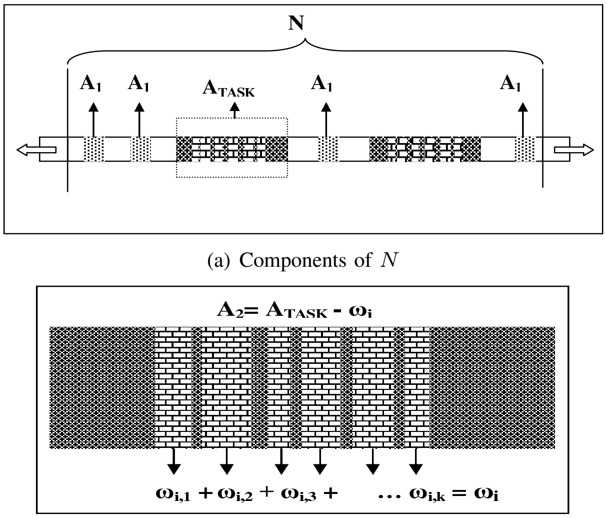
Contributions. In this paper, we detail the design and implementation of four progressively more efficient scheduling systems designed to support AAS embedded applications. The designs are applicable to virtually any modern MCU. For the sake of presentation, we focus on the popular *ATmega* family of devices, which are used in a number of sensor networking platforms [14]–[16]. For each scheduler implementation, we present a closed-form algebraic model that captures the scheduling overhead as a function of task load and other parameters. These models are used to characterize the comparative performance among the designs. To supplement this analysis, we also conduct physical power profiling studies using an ATmega644-based sensor networking platform. The results provide a clear picture of the power consumption profile associated with each design, as well as the comparative lifetime benefits they provide.

We emphasize that these designs are practically motivated. They evolved over the course of 18 months while developing a large-scale environmental monitoring network deployed in the city of Aiken, South Carolina [8]. In 2011, the city's stormwater treatment system was redesigned to reduce the environmental impacts associated with stormwater runoff. The monitoring network was installed in targeted areas throughout the city to monitor the modified treatment system. Our subteam was responsible for the design of the wireless sensor platforms and the associated firmware used to construct the network. The design process was guided by the need to support continuous, uninterrupted data collection in the face of unattended operation (since Aiken is relatively remote). Maximizing the lifetime of our almost-always-sleeping system was a principal goal. In addition to yielding a successful network deployment, the experience resulted in the first systematic analysis of AAS schedulers, which we present here.

In Section II we provide a formal definition for the scheduling problem in AAS systems and present the related work in Section III. In Sections IV and V we present the designs and the corresponding algebraic models for the schedulers, respectively. The comparative analysis and experimental results are presented in Section VI. Section VII concludes the paper.

II. Problem Statement

The smallest unit of work that may be scheduled in an AAS system is a *task*, an action taken in response to a timer event. When a scheduler wakes and has no tasks to execute, a small amount of time is expended, referred to as the *null activation period*, denoted by A_1. The amount of time expended when the scheduler wakes and there are tasks to execute, including



(a) Components of N

(b) A_{TASK} Expanded

Fig. 1. A_1, A_2, A_{TASK}, ω_i, and N

task execution time, is referred to as the *task activation period*, denoted by A_{TASK}.

In a given time period N, a scheduler experiences A_1 and A_{TASK} multiple times and sleeps the rest of the time. The number of times the scheduler experiences A_1 and A_{TASK} in a time period N is given by n_1 and n_2, respectively. Each instance, i, of A_{TASK} within N consists of time spent executing the task functions, given by ω_i, and the rest of the time expended prior to, inbetween, and after task execution, denoted by A_2. The relationship between A_1, A_2, A_{TASK}, ω_i, and N is illustrated in Figure 1. The total time spent executing the task functions in the time period N is given by W, calculated as the sum of all ω_i, where $i = \{1, 2, ..., n_2\}$. In the i^{th} occurence of A_{TASK}, ω_i is calculated as the sum of all $\omega_{i,j}$, where $j = \{1, 2, ..., \mathtt{n_{executed}}\}$; $\mathtt{n_{executed}}$ denotes the number of task functions executed in the i^{th} task activation period. Assuming all tasks are periodic, and $\mathtt{n_{executed}}$ is constant for all values of $\mathtt{i}$, the total time taken to execute all task functions, W, in time period N is calculated as:

$$W = \sum_{i=1}^{n_2} \sum_{j=1}^{\mathtt{n_{executed}}} \omega_{i,j} \qquad (1)$$

The *scheduler load* α is the fraction of time the system is either busy scheduling tasks or executing them within time period N. The *task load* β is the fraction of time the system is busy executing just the task functions, given by W, within time period N. Assuming n_1, n_2, W, and N are fixed, and $n_2 A_{TASK} = n_2 A_2 + W$, α can be expressed as:

$$\begin{aligned} \alpha &= \frac{(\mathtt{n_1 A_1 + n_2 A_{TASK}})}{\mathtt{N}} \\ &= \frac{(\mathtt{n_1 A_1 + n_2 A_2 + W})}{\mathtt{N}} \\ &= \frac{\mathtt{n_1 A_1 + n_2 A_2}}{\mathtt{N}} + \beta \qquad (2) \end{aligned}$$

Objective. In an ideal scheduler, with no scheduling overhead, $\alpha = \beta$. To minimize the value of α, both A_1 and A_2 need to be minimized. Our objective is to design a scheduler with the least possible A_1 value; since $n_1 \gg n_2$ in AAS systems, a lower A_1 value, even at the expense of a higher A_2 value, will help in maximizing the efficiency and battery life expectancy of a scheduler.

III. Related Work

Levis et al. present TinyOS [10], one of the most widely-used sensor network operating systems. TinyOS includes a task scheduler that executes non-preemptive tasks *posted* for later execution. TinyOS uses a fixed-length, FIFO scheduler by default. To reduce energy consumption, the scheduler puts the processor to sleep whenever the task queue is empty. Its successor, TinyOS2 [11], uses a similar FIFO scheduler; an earliest-deadline-first implementation is also available. Compared to TinyOS, TinyOS2 introduces more overhead when posting and executing a task, but less overhead when the task queue is empty.

Han et al. present SOS [9], another event-driven operating system. Software modules communicate using direct calls and message passing via a FIFO scheduler with two levels of priority. High priority messages are reserved for time critical events, such as hardware interrupts.

Dunkels et al. present Contiki [7], another event-based operating system with support for event prioritization. A non-preemptive event scheduler schedules asynchronous and synchronous events. *Asynchronous* events are deferred procedure calls enqueued in a FIFO handling queue. *Synchronous* events are immediately scheduled at the front of the queue.

Bhatti et al. present MANTIS [2], a multi-threaded sensor network operating system. In MANTIS, a fixed thread table maintains all threads, which are executed using round-robin scheduling within priority levels. The scheduler is driven by a timer interrupt, which triggers context switching among threads. MANTIS also allows users to specify the sleep period of threads. The scheduler calculates the earliest wake-up time and uses an idle background thread to put the CPU to sleep when all other threads are blocked.

Chen et al. present Enix [6], a cooperative threading solution for sensor networks, which uses *setjump* and *longjump* to implement low overhead context switching. It supports priority-based and round-robin scheduling policies using linear search and bitmap-based thread lookups. Other multi-threaded sensor network operating systems, including LiteOS [3] and RETOS [5], use similar schedulers. In particular, LiteOS supports priority-based and round-robin scheduling policies, and RETOS supports POSIX scheduling, which boosts the priority of a thread when events need to be handled quickly.

While each has its advantages, none of these systems are well matched for AAS scheduling. Event-based schedulers using FIFO mechanisms or priorities are not designed to account for the sleep requirements of AAS systems. Thread-based schedulers are also inefficient in this context. POSIX-like soutions introduce significant overhead, while the use of small epochs in other multi-threaded solutions is energy-inefficient. By contrast, our work focuses on the systematic design and analysis of scheduling solutions suited specifically to AAS systems.

Caracas et al. describe an energy efficient optimization strategy based on variable sleep intervals [4]. They define a *knapsack problem* to compute the minimum number of (pre-specified) sleep intervals required to achieve a given sleep period, but do not provide the solution details. We present the implementation details for a similar variable-sleep scheduling strategy, where we use a greedy solution to the knapsack problem and analyze its complexity and performance characteristics.

IV. AAS Scheduling

We focus on a canonical implementation of an AAS scheduler, where a task is composed of a function pointer, a task type, a period, and a due date. The function pointer points to the executable `task` body. The task `type` is either `one_shot` or `periodic`, corresponding to a task that expires after it has been executed, and a task that is continually rescheduled, respectively. The `period` specifies how often the task should be activated. The due date records the time at which the task should occur next.

The basic scheduling functions in our implementation are `scheduler_init()`, `schedule_task()`, and `scheduler_run()`. `scheduler_init()` handles scheduler initialization during system start-up, and `schedule_task()` is used to schedule new tasks. The system spends much of its lifetime in `scheduler_run()`; it contains the core of the scheduling logic and is invoked to start the scheduler.

The scheduler designs presented in the next sections depend on the hardware system, particularly the timer mechanism. The target MCU implements the system clock using an 8-bit counter register, driven by an external oscillator oscillating at a rate of `32.768KHz`. A prescaler of `128` results in an overflow interrupt being triggered once per second; this suspends the executing instruction and begins the interrupt service routine (ISR), where the system time is updated. If the processor is in a sleep state, it wakes and enters the ISR. Upon completion, the processor resumes execution following the call to sleep.

A. A Basic Scheduler

We present a basic AAS scheduler implementation that parallels the design of existing embedded task schedulers [9]–[11]. `system_task_buffer`, an `N`-element array, is initialized (with $NULL$ entries) within `scheduler_init()`. `schedule_task()` finds the first empty slot and stores the task passed as argument.

`scheduler_run()`, shown in Listing 1, iterates indefinitely in the outer `while` loop. In each iteration, referred to as an *execution cycle*, the scheduler steps through `system_task_buffer` and executes each task with an expired due date. When a `one_shot` task completes, the task is removed from `system_task_buffer`. When a `periodic` task completes, its due date is updated based on its period. When there are no tasks to execute, the scheduler enters its sleep cycle.

This simple scheduler has a significant power consumption footprint due to the time required to determine whether there are tasks to execute. Even when there are no tasks to execute, the scheduler wakes and cycles through the entire task buffer. Since the time expended is bounded by `N`, an increase in task capacity degrades system performance. A scheduler that could perform a constant time lookup into the task array for available tasks would be more desirable.

```
void scheduler_run() {
 while(true) {
  bool task_executed;
  do {
   task_executed = false;
   uint32_t current_time = current_system_time();
   uint8_t task_index;
   for(task_index = 0; task_index < TASK_QUEUE_CAPACITY; task_index++) {
    // if the current (non-empty) task is due
    if((system_task_buffer[task_index].task != NULL) &&
       (current_time >= system_task_buffer[task_index].due_date)) {
     // execute the task function
     (*system_task_buffer[task_index].task)();
     // handle rescheduling / removal
     if(system_task_buffer[task_index].type == ONE_SHOT) {
      system_task_buffer[task_index].task = NULL;
     } else {
      system_task_buffer[task_index].due_date +=
       system_task_buffer[task_index].period;
     }
     task_executed = true;
    }
   }
  } while(task_executed);
  set_sleep_mode(SLEEP_MODE_PWR_SAVE);
  sleep_mode();
 }
}
```

Listing 1. `scheduler_run()` (Basic Scheduler)

```
void scheduler_run() {
 while(true) {
  bool task_executed;
  do {
   task_executed = false;
   uint32_t current_time = current_system_time();
   uint8_t task_index = 0;
   while((task_index = 16 - ffs(task_bitmap_active)) < 16) {
    if (current_time >= system_task_buffer[task_index].due_date) {
     task_executed = run_task(task_index,current_time);
    } else {
     task_bitmap_active &= ~(1 << (15 ^ task_index));
     task_bitmap_inactive |= (1 << (15 ^ task_index));
    }
   }
  } while(task_executed);
  task_bitmap_active = task_bitmap_inactive;
  task_bitmap_inactive = 0;
  set_sleep_mode(SLEEP_MODE_PWR_SAVE);
  sleep_mode();
 }
}

static inline bool run_task(uint8_t task_index,uint32_t current_time) {
 // execute the task
 (*system_task_buffer[task_index].task)();
 // handle rescheduling / removal
 if(system_task_buffer[task_index].type == ONE_SHOT) {
  task_bitmap_active &= ~(1 << (15 ^ task_index));
 } else {
  system_task_buffer[task_index].due_date +=
    system_task_buffer[task_index].period;
  if(system_task_buffer[task_index].due_date > current_time) {
   task_bitmap_active &= ~(1 << (15 ^ task_index));
   task_bitmap_inactive |= (1 << (15 ^ task_index));
  }
 }
 return (true);
}
```

Listing 2. `scheduler_run()` and `run_task()` ($O(1)$ Scheduler)

B. The $O(1)$ Scheduler

The $O(1)$ scheduler is based loosely on the Linux 2.6.8.1 scheduler [1]. Adapted to our system, when there are no tasks in the queue, the scheduler performs a constant-time lookup and returns to sleep. This scheduler also uses `system_task_buffer` to store scheduled tasks. Two supporting queues are also introduced; the *active task queue* stores tasks which must be executed in the current execution cycle, and the *idle task queue* stores tasks that have been executed, but which must be re-evaluated the next time the system wakes. To achieve constant-time task lookup, the queues are implemented using bitmaps; a 1 at bit position n indicates a task in the n^{th} element of `system_task_buffer`. At boot time, `schedule_task()` locates the first free index in the task buffer and the corresponding location in the active and idle bitmaps are set and cleared, respectively.

In the execution phase, a call to `ffs()` is performed on the active task bitmap, as shown in Listing 2. The `ffs()` function, provided by the Atmel AVR C library [19], returns the position of the least significant bit set in a 16-bit word; or 0, if none are set. If a task is identified in the active task queue with a due date greater than the current system time, its index position is cleared in the active task bitmap and set in the idle task bitmap. If the identified task has an expired due date, it is executed by `run_task()`, followed by its removal or rescheduling. Task removal entails removal of the corresponding task bit from the active task bitmap. Task rescheduling involves updating the two bitmaps and the due date of the task in `system_task_buffer`.

During the execution cycle, if there are no tasks to execute, the scheduler performs an $O(1)$ lookup into the active task queue and returns to sleep. While $O(1)$ run-time is desirable, a large constant results in increased power consumption. We next consider a design that introduces increased overhead when there are tasks to execute, but very little overhead when there are no tasks to execute — our common case.

C. The $O(n)$ Scheduler

The $O(n)$ scheduler removes the call to the expensive `ffs()` function; it requires constant time to identify a task to execute, and linear time to reschedule the task post-execution.

Tasks are stored as nodes in a linked list instead of the statically allocated task array. *Slab allocation* is implemented using a static block of memory capable of holding `N` task nodes, `task_free_list`, a pointer to the list of free memory (within the static memory block), and `task_queue`, a pointer to the linked list of tasks. Task scheduling involves allocating a node from `task_free_list`, populating the node, and inserting the node in `task_queue` based on due date.

```
void scheduler_run() {
 uint32_t system_sleep_cycle_counter = 0;
 while(true) {
  bool task_executed;
  do {
   task_executed = false;
   uint32_t current_time = current_system_time();
   while((task_queue != NULL) &&
         (task_queue->due_date <= current_time)) {
    // execute the task
    task_node_ptr_t task_ptr = task_queue;
    (task_ptr->task)();
    task_executed = true;
    task_queue = task_queue->next;
    // handle rescheduling / removal
    if(task_ptr->type == ONE_SHOT) {
     free_list_free(&task_free_list, (node_ptr_t) task_ptr);
    } else {
     task_ptr->due_date += task_ptr->period;
     insert_task_in_scheduling_queue(&task_queue, task_ptr);
    }
   }
   system_sleep_cycle_counter = task_queue->due_date - current_time;
  } while(task_executed);
  set_sleep_mode(SLEEP_MODE_PWR_SAVE);
  while(system_sleep_cycle_counter--) {
   sleep_mode();
  }
 }
}
```

Listing 3. `scheduler_run()` ($O(n)$ Scheduler)

`scheduler_run()`, shown in Listing 3, traverses the list of scheduled tasks and executes those that are due. The removal of `one_shot` tasks is handled by freeing the corresponding task node and returning it to `task_free_list`. Rescheduling of `periodic` tasks is handled by updating the corresponding task's due date and re-inserting the task at the correct position in the priority queue.

Since tasks are ordered by due date, it is straightforward to determine when the next task needs to be executed, just prior to sleeping. When the system is done executing tasks, the difference between the earliest task due date and the current system time is recorded. After the system wakes up, a simple check on this value allows the scheduler to decide if there are any tasks to execute, and saves it from having to access the node list. The scheduler therefore experiences shorter wake cycles when there are no tasks to execute.

Since an AAS system typically wakes to find nothing to execute, even a small amount of time expended during a wake cycle can add performance penalties. With the given hardware and interrupt design, where the processor has to wake every second, this is the best performance that could be achieved. However, a scheduler capable of altering the interrupt behavior would yield even better performance.

D. The Intelligent Sleep Scheduler

The basis of the intelligent sleep scheduler (ISS) is the $O(n)$ scheduler, with multiple updates to the wake, sleep, and clock logic. The central idea is that the rate at which the overflow interrupt is generated can be changed by choosing a different clock prescaler, thus making the duration of the processor sleep period tunable; the clock prescaler can be set to 128, 256, or 1024, so that overflow interrupts are triggered at 1, 2, and 8 second intervals, respectively.

```
void scheduler_run() {
 uint32_t system_sleep_cycle_counter = 0;
 while(true) {
  bool task_executed;
  do {
   task_executed = false;
   uint32_t current_time = current_system_time();
   while((task_queue != NULL) &&
         (task_queue->due_date <= current_time)) {
    ... same as O(n) scheduler ...
   }
   system_sleep_cycle_counter = task_queue->due_date - current_time;
  } while(task_executed);
  intelligent_sleep(system_sleep_cycle_counter);
 }
}

inline void intelligent_sleep(uint32_t int_system_sleep_counter) {
 int_system_sleep_counter = int_system_sleep_counter - 1;
 // determine the number of 1, 2, and 8 second sleep cycles.
 // 1 second sleep required?
 sleep_cycle[0] = (int_system_sleep_counter & 0x1);
 // 2 second sleep required?
 int_system_sleep_counter >>= 1;
 sleep_cycle[1] = (int_system_sleep_counter & 0x1);
 int_system_sleep_counter >>= 1;
 sleep_cycle[1] += ((int_system_sleep_counter & 0x1) << 1));
 // 8 second sleep required?
 int_system_sleep_counter >>= 1;
 sleep_cycle[2] = int_system_sleep_counter;

 // compute total number of sleep cycles and begin sleeping
 int_system_sleep_counter = sleep_cycle[0]
   + sleep_cycle[1] + sleep_cycle[2];
 set_sleep_mode(SLEEP_MODE_PWR_SAVE);
 do {
  sleep_mode();
 } while(int_system_sleep_counter--);
}
```

Listing 4. `scheduler_run()` and `intelligent_sleep()` (Intelligent Sleep Scheduler)

Listing 4 presents the `scheduler_run()` implementation. The difference between the earliest task due date and the current system time is recorded at the end of an execution cycle. The system then invokes `intelligent_sleep()`, which partitions this value into multiple divisors, so as to calculate the least number of sleep cycles that can be created from 1, 2, and 8-second intervals.

The current rate at which the interrupt is triggered is called an *epoch*. Changing the clock prescaler (and the epoch) at any arbitrary instant causes the 8-bit counter register to contain a value less than 256, accounting for the partial second of elapsed time since the last overflow interrupt. Since epoch values vary over time, the semantics of this *partial time* change in a complex way. Let the epoch be e_1 at time t_1 when the overflow interrupt is triggered. Let the epoch assume the value e_2 at t_2. Partial time is defined as $(t_2 - t_1)$, calculated as a function of e_1 and the value in the 8-bit counter register when the epoch was changed to e_2. Partial times for each epoch (i.e. 1, 2, 8) are stored in an array.

```
#define PARTIAL_TIME_UPDATE() \
  // update system time based on partial time accumulation \
  system_clock_cycles += system_time_fraction*temp_system_time_epoch; \
  if(system_clock_cycles & ~(0xFF)) { \
   system_time += system_clock_cycles>>8; \
   system_clock_cycles &= 0xFF; \
  }

// timer2 overflow handler
ISR(SIG_OVERFLOW2, ISR_BLOCK) {
 // increment system time by current epoch
 system_time += system_time_epoch;
 if (sleep_cycle[0]) {
  // 1-second sleep required; current epoch 1-second
  sleep_cycle[0] = 0;
 } else if (sleep_cycle[1]) {
  // 2-second sleep required; decrement cntr, change prescaler if required
  sleep_cycle[1]--;
  if (system_time_epoch != 2) {
   temp_system_time_epoch = system_time_epoch;
   system_time_epoch = 2;
   TCCR2B = (1 << CS22) | (1 << CS21);
   while(ASSR & 0x1F);
   system_time_fraction = TCNT2;
   TCNT2 = 0x0;
   while(ASSR & 0x1F);
   PARTIAL_TIME_UPDATE();
  }
 } else if (sleep_cycle[2]) {
  // 8-second sleep required; decrement cntr, change prescaler if required
  sleep_cycle[2]--;
  if (system_time_epoch != 8) {
   ... analogous to above case ...
  }
 } else if (system_time_epoch != 1) {
  // all counters are 0; prescaler reset for mandatory 1-second sleep
  ... analogous to above case ...
 }
}
```

Listing 5. Overflow ISR (Intelligent Sleep Scheduler)

To obtain the least accumulated partial epoch, the overflow ISR is identified as the optimal place to change the prescaler. Thus, after an execution cycle, the processor enters a 1-second sleep period, waits for the ISR to be triggered, and then changes the prescaler. Listing 5 contains the code for the updated overflow ISR. The overflow ISR ensures that the prescaler is set to the 1-second interval for the mandatory sleep cycle after the 2 and 8-second sleep cycles have been executed.

At the start of the ISR, the system time is updated using the value of the current epoch. Next, the change of prescaler (and epoch) is performed, if needed. If the clock prescaler is updated, the corresponding partial time is recorded, and the value of accumulated partial time is calculated as the sum of its previous value and the product of the current partial time and the last epoch value. Since every 256 fractions represents 1 second of time, if accumulated partial time is greater than or equal to 255, the system time is incremented and the accumulated partial time is appropriately updated.

V. Algebraic Models

The schedulers were implemented for the MoteStack, a state-of-the-art in-situ sensing platform, which uses an ATMega644 Atmel 8-bit AVR RISC-based MCU operating at 10 MHz at 3.3V (gcc -0s). A line-by-line code analysis was performed with the assistance of *AVR Studio*, a cycle accurate simulator, to derive the closed-form algebraic models.

A. The Basic Scheduler

In the basic scheduler, the null activation period, A_1, is given (in μs) by:

$$A_1 = 8.9 + 1.5 * \mathtt{n_{queue\_capacity}} + 1.3 * \mathtt{n_{in\_queue}} \tag{3}$$

where $\mathtt{n_{queue\_capacity}}$ denotes the capacity of the task queue, and $\mathtt{n_{in\_queue}}$ denotes the number of tasks in the queue.

A_2 (in μs) is given by the following formula:

$$\begin{aligned} A_2 = {} & 8.9 + 3.1 * \mathtt{n_{executed}} + 2.6 * \mathtt{n_{iter}} \\ & + (1.5 * \mathtt{n_{queue\_capacity}} + 1.3 * \mathtt{n_{in\_queue}}) * \mathtt{n_{iter}} \end{aligned} \tag{4}$$

Recall that $\mathtt{n_{executed}}$ denotes the number of task functions executed in the current task activation period; $\mathtt{n_{iter}}$ denotes the number of times the main scheduler loop executes (Listing 1, *lines 4-24*). Assuming that $\forall i,\ (\omega_i + A_2) \leq 1\ second$, the value of $\mathtt{n_{iter}}$ is calculated as follows:

$$\mathtt{n_{iter}} = 1 + \lceil \frac{1}{\mathtt{task\_period_{min}}} \rceil \tag{5}$$

where $\mathtt{task\_period_{min}}$ is the smallest period value present in the task queue associated with a task that has a due date earlier than the current system time.

B. The $O(1)$ Scheduler

In the $O(1)$ scheduler, A_1 is given by:

$$\begin{aligned} A_1 = {} & 14.5 + 24 * \mathtt{n_{in\_queue}} \\ & + (2.8 * (\lceil \frac{\mathtt{n_{queue\_capacity}}}{16} \rceil - 1)) * \mathtt{n_{in\_queue}} \end{aligned} \tag{6}$$

A_2 for the $O(1)$ scheduler is given as follows:

$$\begin{aligned} A_2 = {} & 19.9 + 6.5 * \mathtt{n_{executed}} + \ 24 * \mathtt{n_{in\_queue}} * (\mathtt{n_{iter}} - 1) \\ & + \ 2.8 * (\lceil \frac{\mathtt{n_{queue\_capacity}}}{16} \rceil - 1) * \mathtt{n_{in\_queue}}) * (\mathtt{n_{iter}} - 1) \end{aligned} \tag{7}$$

C. The $O(n)$ Scheduler

The $O(n)$ scheduler has a constant null activation period of 7 μs (A_1).

A_2 is given by the following formula:

$$\begin{aligned} A_2 = {} & 14.4 + (13.7 + \mathtt{t_{ins}}) * \mathtt{n_{executed}} \\ & + \ 5.6 * (\mathtt{n_{iter}} - 1) \end{aligned} \tag{8}$$

$\mathtt{t_{ins}}$ denotes the time spent within the insertion sort during rescheduling, post task execution. The value of $\mathtt{t_{ins}}$ is given by the following formula:

$$\mathtt{t_{ins}} = \begin{cases} 0.2, & \text{if } \mathtt{n_{in\_queue}} = 0; \\ 3.7 * [1, \mathtt{n_{in\_queue}}) & \text{if } \mathtt{n_{in\_queue}} > 0; \end{cases} \tag{9}$$

where $[1, \mathtt{n_{in\_queue}})$ denotes any value between 1 and ($\mathtt{n_{in\_queue}}$ - 1).

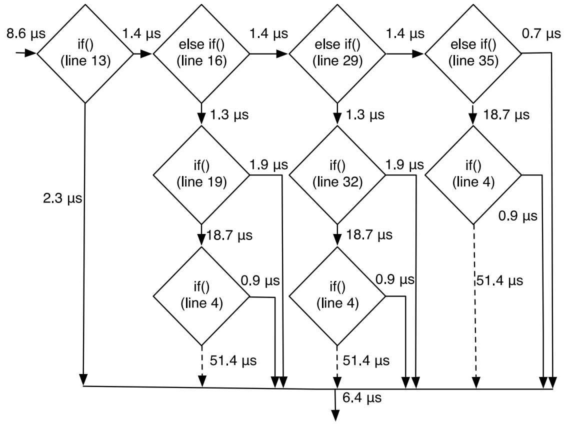


Fig. 2. ISR Execution Profile ($\mathtt{t_{ISR}}$) (Intelligent Sleep Scheduler)

D. The Intelligent Sleep Scheduler

The null activation period (A_1) for the ISS is the hardest to analyze due to its complex ISR control flow paths. A flow chart indicating the different paths is shown in Figure 2. The value A_1 assumes in a given null activation period depends on the values of the various system variables in that specific period and is given by:

$$\mathtt{A_1} = \mathtt{0.6} + \mathtt{t_{ISR}} \tag{10}$$

$\mathtt{t_{ISR}}$ denotes the amount of time elapsed between the start of the overflow ISR (*line 10*, Listing 5) and the start of the scheduling loop of `scheduler_run()` (*line 6*, Listing 4), shown in Figure 2.

The accumulation of partial time fractions in the clock update logic requires 51.4 μs. However, this value is ignored for modeling purposes. The latest possible invocation of the partial update logic (*lines 35-38*, Listing 5) is approximately 31.5 μs after the start of the ISR. Thus, the maximum partial time accumulated is approximately 31.5 μs, close to a single oscillation of the external oscillator. Even in the 1-second interval case, the prescalar is set to 128, and the probability of partial time accumulation is small. Even if it does accumulate, for these 31.5 μs intervals to total 1 second, approximately 31,746 occurences of A_1 or A_2 are required. Hence, the time is assumed to be negligible.

A_2 for the ISS is given by:

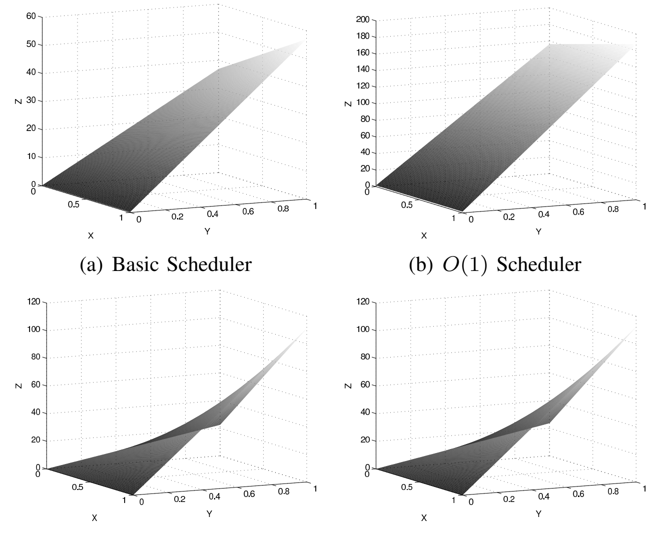
$$\begin{aligned} A_2 = {} & \mathtt{t_{ISR}} + \mathtt{13.4} + \mathtt{5.6} * (\mathtt{n_{iter}} - \mathtt{1}) \\ & + (\mathtt{13.7} + \mathtt{t_{ins}}) * \mathtt{n_{executed}} \end{aligned} \tag{11}$$

where $\mathtt{n_{iter}}$, $\mathtt{n_{executed}}$, $\mathtt{t_{ins}}$, and $\mathtt{t_{ISR}}$ are defined as before.

VI. Results

We first consider the performance of the schedulers based on the algebraic models of their behavior. We then measure the scheduler power consumption for a given set of tasks on physical hardware.

A. Comparative Analysis



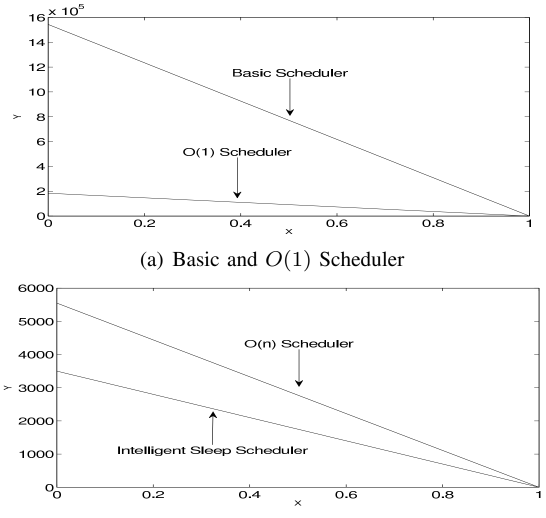
(a) Basic Scheduler

(b) $O(1)$ Scheduler

(c) $O(n)$ Scheduler

(d) Intelligent Sleep Scheduler

Fig. 3. X = $\frac{n_2}{n_1+n_2}$, Y = $\frac{\mathtt{n_{task\_executed}}}{\mathtt{n_{in\_queue}}}$, Z = $n_1A_1 + n_2A_2$



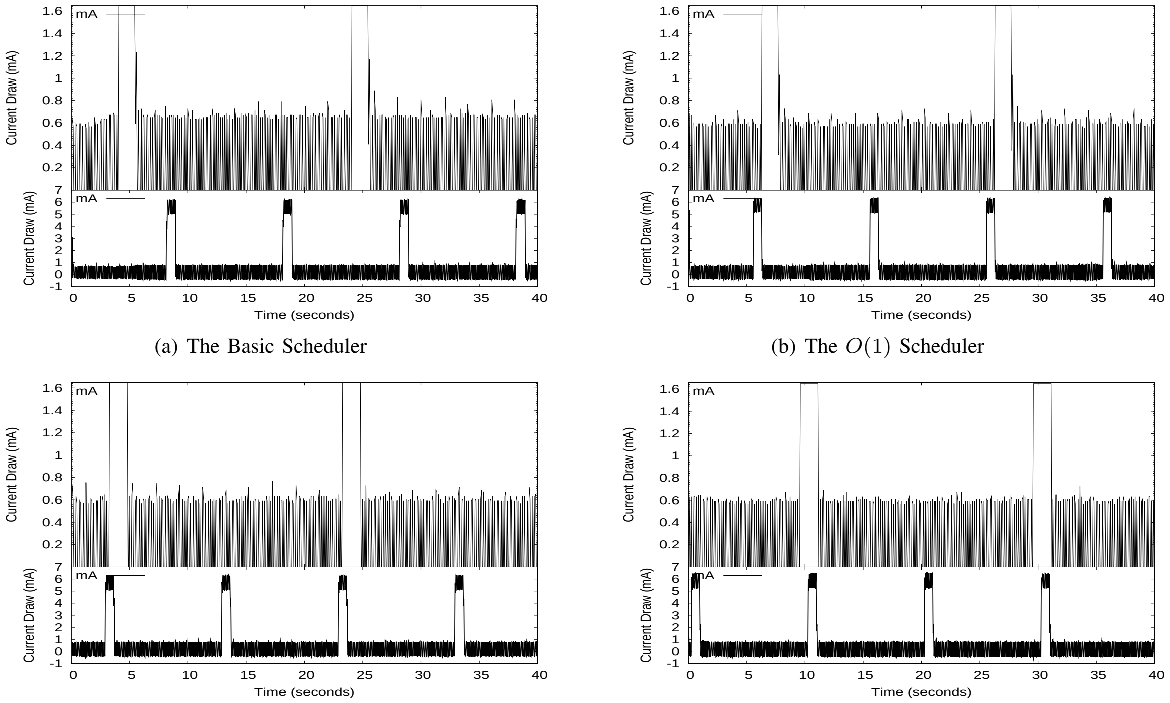
(a) Basic and $O(1)$ Scheduler

(b) $O(n)$ and Intelligent Sleep Scheduler

Fig. 4. *Null Activation Period* Contributions (X = $\frac{n_2}{n_1+n_2}$, Y = n_1A_1 μs)

We compare the scheduling overhead of each scheduler under varying load conditions; results are shown in Figures 3 and 4. Due to the number of variables in the equations for A_1 and A_2, we make some assumptions to limit the evaluation space. We fix both $\mathtt{n_{queue\_capacity}}$ and $\mathtt{n_{in\_queue}}$ to 128, and $\mathtt{n_{iter}}$ to 2 (limiting $\mathtt{task\_period_{min}}$ to greater than or equal to 1 second – Eq. (5)). We generate the values of $\mathtt{t_{ins}}$ using a pseudo random number generator and fix the values for all subsequent calculations across the schedulers. For each scheduler, we measure the *scheduling overhead*, given by $n_1A_1 + n_2A_2$, in seconds, on the *Z-axis*, when N is set to 500 seconds. N is composed of $(n_1 + n_2)$ 1-second counts. We plot the *fraction of tasks executed* on the *X-axis*, given by $\mathtt{n_{task\_executed}}$ over $\mathtt{n_{in\_queue}}$, and the *load factor* (given by n_2 over $(n_1 + n_2)$) on the *Y-axis*. The system load factor is helpful in understanding the interplay between A_1 and A_2.

Figures 3(a) and 3(b) show the results for the basic and $O(1)$ schedulers, respectively. The planar slopes for both graphs are similar, owing to the fact that both schedulers yield A_2 values that depend primarily on similar $\mathtt{n_{queue\_capacity}}$ and $\mathtt{n_{in\_queue}}$ coefficients. At higher load factors, where $n_2 >> n_1$, the



(a) The Basic Scheduler

(b) The $O(1)$ Scheduler

(c) The $O(n)$ Scheduler

(d) The Intelligent Sleep Scheduler

Fig. 5. Scheduler Power Consumption Profiles

$O(1)$ scheduler performs worse than the basic scheduler, but at lower load factors, the differences are negligible. Figures 3(c) and 3(d) show the results for the $O(n)$ scheduler and ISS, respectively; again the curves are similar. The $O(n)$ scheduler and ISS incur less overhead than the basic and $O(1)$ schedulers at load factors below 0.8, as they are not dependent on $n_{\texttt{queue\_capacity}}$. We also observe that at higher load factors, the value of $n_{\texttt{task\_executed}}$ affects all schedulers significantly. At lower load factors, both the $O(n)$ and intelligent schedulers exhibit very low overhead (<2% for load factors of 0.3). To further differentiate the two schedulers, we consider their performance at very low load factors, on the order of 0.001, typical in AAS systems. Since the overhead contribution of A_1 is significantly larger than A_2 at very low load factors, we focus on the impact of A_1 in isolation. In Figures 4(a) and 4(b), we measure, for each scheduler, the contribution of A_1, given by $n_1 A_1$, on the *Y-axis*, against the load factor, given by n_2 over $n_1 + n_2$, on the X-axis. With a side-by-side comparison, we see that the basic and $O(1)$ schedulers have a much higher null activation period contribution than the other two schedulers — approximately three orders of magnitude larger and are relatively inefficient at lower load factors. We also observe that the ISS performs the best among all the schedulers presented. Its ability to sleep for longer periods of time gives the ISS an edge over schedulers which need to wake every second.

B. Power Consumption Profile

We now characterize the power consumption profiles of the four schedulers. For this purpose, we installed a test application on the MoteStack device, using each scheduler. The application schedules a periodic $null$ task with a duration of $750ms$, executed every $10s$. We connected a 10Ω resistor in series with the power supply of the MoteStack and measured the voltage difference across the resistor, using an oscilloscope. The voltage change is directly proportional to the current draw (and power consumption, when voltage is constant) by Ohm's Law. Figures 5(a) – 5(d) summarize the consumption profiles for the four schedulers. In each graph, the horizontal axis represents time, and the vertical axis represents current draw. The bottom halves of the figures show the complete consumption profile; the task activation periods are visible. The top halves show a magnified view of the profile, such that the null activation periods can be seen. The peaks for the null activation periods can be observed at the end of each second in Figures 5(a), 5(b), and 5(c), while fewer such peaks can be noticed in Figure 5(d), indicating longer sleep periods.

We sample data over a 10-second window, which captures current draw values for a single task activation period, multiple null activation periods, and the associated sleep periods. We calculate the average overall and A_{TASK} current draws – the A_{TASK} values vary due to the inherent scheduler designs. The average current draw for the basic scheduler (Figure 5(a)) over the window is 0.613 mA (average A_{TASK} current draw is 5.52 mA), while the average current draw for the $O(1)$ scheduler (Figure 5(b)) is 0.605 mA (average A_{TASK} current draw is 5.28 mA). The average current consumption for the $O(n)$ (Figure 5(c)) and the intelligent sleep (Figure 5(d)) schedulers is 0.616 mA (average A_{TASK} current draw is 5.56 mA) and 0.603 mA (average A_{TASK} contribution is 5.49 mA), respectively.

Figure 6 presents the life expectancy of a $1000mAh$ battery,

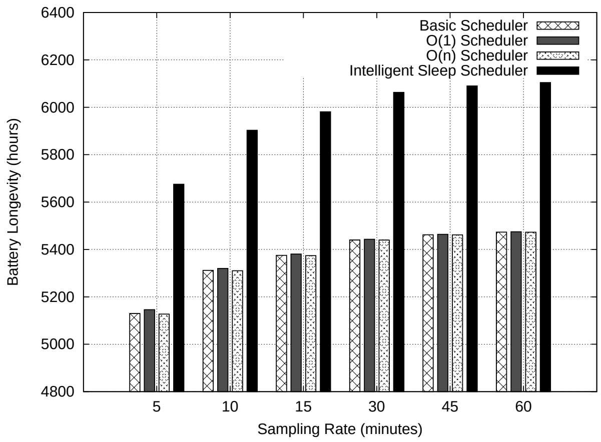


Fig. 6. Battery Life Expectancy

when it is supplying power to a MoteStack, running the four schedulers under different almost-always-sleeping scenarios. Data for Figure 6 was obtained by extrapolating the average current draw and average A_{TASK} current draw values from Figures 5(a) – 5(d), and applying them to applications which sleep for 5, 10, 15, 30 45, and 60 minutes between task executions. We observe that the intelligent sleep scheduler consistently yields higher battery longevity for all the applications.

Specifically, consider the application which sleeps for 15 minutes between tasks, a typical sampling period for environmental monitoring networks of the type deployed in Aiken, SC. A MoteStack running this application and drawing its power from a $1000mAh$ battery would last approximately 5,375 hours using the basic scheduler. The same MoteStack would last for 5,380 hours using the $O(1)$ scheduler. A MoteStack using the $O(n)$ scheduler would last for 5,374 hours, while the ISS offers the longest runtime, of approximately 5,980 hours – 10% longer than any of the other schedulers. This is a significant increase in longevity in the context of large sensor network deployments. Though all the scheduler designs dictate a linear decrease in power consumption with an increase in the time period between task activation periods, not surprisingly, the rate of the decrease for the ISS is higher compared to the others, due to its ability to sleep for longer periods, thus enabling a longer battery life.

VII. Conclusion And Future Work

In this paper, we presented the design, implementation, and analysis of four progressively more efficient schedulers designed to support *almost-always-sleeping* embedded applications. This is the first systematic consideration of this increasingly relevant class of schedulers. We presented a *basic scheduler* which uses a rudimentary array to store tasks. We next presented the $O(1)$ *scheduler* based on the Linux 2.6.8.1 scheduler. This design incurs performance penalties due to an expensive call to `ffs()`. Next, we presented the $O(n)$ *scheduler*, which uses a priority queue to store tasks and improves its tracking of sleep cycles, performing significantly better than the previous schedulers. Finally, we presented the *intelligent sleep scheduler*, in which we make use of hardware features to extend physical sleep cycles, design a variable-sleep scheduling strategy, and further reduce scheduling overhead. On analyzing the scheduler runtimes, we observed that the $O(n)$ and the intelligent sleep schedulers work well below a certain load factor. However, under lower load cycles, the intelligent sleep scheduler design performs markedly better than all other designs due to its variable-sleep strategy, even at the expense of added code complexity.

VIII. Acknowledgment

This work was supported in part by the National Science Foundation (awards CNS-0745846, CNS-1126344).

References

[1] Josh Aas. Understanding the linux 2.6.8.1 cpu scheduler. *Silicon Graphics International, 2005. http://joshaas.net/linux/* [retrieved: June, 2012].

[2] Shah Bhatti et al. MANTIS OS: An embedded multithreaded operating system for wireless micro sensor platforms. *Mob. Netw. Appl.*, 10:563–579, 2005.

[3] Qing Cao, Tarek Abdelzaher, John Stankovic, and Tian He. The LITEOS operating system: Towards UNIX-like abstractions for wireless sensor networks. IPSN ’08, pages 233–244. IEEE, 2008.

[4] Alexandru Caracas et al. Energy-efficiency through micro-managing communication and optimizing sleep. In *SECON*, pages 55–63, 2011.

[5] Hojung Cha et al. RETOS: Resilient, expandable, and threaded operating system for wireless sensor networks. IPSN ’07, pages 148–157. ACM, 2007.

[6] Yu-Ting Chen, Ting-Chou Chien, and Pai H. Chou. ENIX: A lightweight dynamic operating system for tightly constrained wireless sensor platforms. SenSys ’10, pages 183–196. ACM, 2010.

[7] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. CONTIKI - a lightweight and flexible operating system for tiny networked sensors. *Annual IEEE Conference on Local Computer Networks*, pages 455–462, 2004.

[8] Gene W. Eidson et al. The South Carolina digital watershed: End-to-end support for real-time management of water resources. *International Journal of Distributed Sensor Networks*, 2010.

[9] Chih-Chieh Han, Ram Kumar, Roy Shea, Eddie Kohler, and Mani Srivastava. A dynamic operating system for sensor nodes. MobiSys ’05, pages 163–176. ACM, 2005.

[10] Philip Levis et al. TINYOS: An operating system for sensor networks. In *Ambient Intelligence*, pages 115–148. Springer, 2005.

[11] Philip Levis et al. T2: A 2^{nd} generation OS for embedded sensor networks. 2005.

[12] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *WSNA ’02*, pages 88–97. ACM Press, 2002.

[13] Joseph Polastre, Robert Szewczyk, and David Culler. TELOS: Enabling ultra-low power wireless research. In *IPSN ’05*, pages 364–369. IEEE Press, 2005.

[14] Crossbow Technologies. Iris datasheet. http://bullseye.xbow.com:81/. [retrieved: June, 2012]

[15] Crossbow Technologies. Mica2 datasheet. http://bullseye.xbow.com:81/. [retrieved: June, 2012]

[16] Crossbow Technologies. MicaZ datasheet. http://bullseye.xbow.com:81/. [retrieved: June, 2012]

[17] Andreas Terzis et al. Wireless sensor networks for soil science. *International Journal of Sensor Networks*, 7:53–70, 2010.

[18] Gilman Tolle et al. A macroscope in the redwoods. In *SenSys ’05*, pages 51–63. ACM Press, 2005.

[19] Joerg Wunsch et al. AVR Libc. http://www.nongnu.org/avr-libc/. [retrieved: June, 2012]

Heterogeneous Wireless Sensor Network Simulation

D. Navarro, M. Galos, F. Mieyeville, W. Du
Université de Lyon, Institut des Nanotechnologies de Lyon (INL)
UMR5270 - CNRS, Ecole Centrale de Lyon, Ecully, F-69134, France
David.Navarro@ec-lyon.fr

Abstract - **Based on our previous work on the development of a Wireless Sensor Network (WSN) simulation platform, we present here its ability to run simulations on heterogeneous nodes. This platform allows system-level simulations with low level accurate models, with graphical inputs and outputs to easily simulate such distributed systems. In the testbed we consider, the well known IEEE 802.15.4 standard is used, and different microcontrollers units (MCU) and radiofrequency transceivers compose the heterogeneous nodes. It is also possible to simulate complex networks or inter-acting networks; that is a more realistic case, as more and more hardware devices exist and standards permit their interoperability. This simulation platform can be used to explore design space in order to find the hardware devices and IEEE 802.15.4 algorithm that best fit a given application. Packet Delivery Rate (PDR) and packet latency can be evaluated, as other network simulators do. Energy consumption of sensor nodes is detailed with a very fine granularity: partitioning over and into hardware devices that compose the node is studied.**

Keywords – Wireless Sensor Network, WSN, heterogeneous, simulation, model, SystemC

I. Introduction

Wireless Sensor Networks (WSN) are widespread sensory systems. They are used in a variety of applications, such as environmental data collection, security monitoring, logistics or health [1]. Wireless Sensor Networks are large-scale networks of resource-constrained sensor nodes that are deployed at different locations. Limited resources are: energy, memory and processing. The sensor nodes cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration or pressure. Because of autonomy requirements, they have a specific architecture; they are typically composed of one or more sensors, an 8-bit or 16-bit MCU, sometimes a non-volatile memory, a radiofrequency transceiver and an energy supply. Typical hardware structures are detailed in Figure 1, where we can recognize two heterogeneous nodes. Manufacturers of WSN hardware include: ATMEL, Texas Instruments or Microchip MCU and Texas Instruments, ATMEL, Freescale, or ST-Microelectronics radiofrequency transceivers. Linux systems composed of 32-bit RISC processors exist – like the well known Crossbow's Stargate platform - but energy consumption is prohibitive and autonomy is largely affected, thus relegating these products to the border of the WSN field, often for high data rate applications. We do not consider such systems, and we focus on several months of battery life systems.

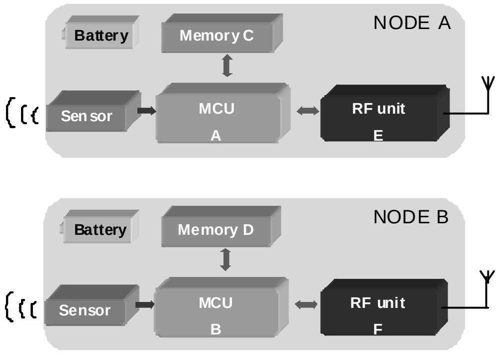


Figure 1. Typical node architectures in a Wireless Sensor Network (heterogeneous network)

Wireless Sensor Networks design is a difficult task, because designer has to develop a network at system level, with low level (at sensor node: hardware and software) constraints. CAD tools are also required to make system-level (hardware and software) simulations, taking low-level parameters into account. It is what our simulator –IDEA1- permits. Moreover, we detail in this paper a new feature of our simulation platform, which supports heterogeneous sensor nodes.

II. Wireless sensor networks simulators

Many simulators have been developed last few years [2-6]. Unfortunately, most of them are restricted to specific hardware or precisely focus on either network level or node level. Research on sensor network evaluation can be broadly divided in two categories: network simulators enhanced with node models, and node simulators enhanced with network models. A thorough exploration of this field is given in [7].

Typical network simulators are general purpose network simulator, such as Network Simulator NS-2 [5] and OMNeT++ [6] (and their declinations). The problem is these interesting network simulators are not sensor platform specific and they are also too high level for hardware considerations. Moreover, there is no separation between computation and communication models. That modeling is

ISBN: 978-1-61208-207-3

also not suitable for hardware replacements and explorations. Then, such simulators do not have accurate energy models [8].

Node simulators refer to precise hardware description, with a synchronization strategy among the nodes, such as Avrora [9], or SCNSL - SystemC Network Simulation Library [10]. These simulators are better suited for embedded system designers, requiring precise low level models for top-down (network to node) approach. SCNSL is a networked systems simulator, written in SystemC and C++. Because SystemC is a C++ class library, it has the advantage to be able to model hardware, software, and network. SystemC is a classical and widely used modeling language in electronic systems design and particularly in System-On-Chip design where it enables hardware/software communication and protocol exploration. Our simulation platform is also based on SystemC and C++, and SCNSL was the starting point of our work.

III. PPROPOSED SYSTEM-LEVEL MODELS

Our models architecture is close to hardware architecture, as Figure 2 (compared to Figure 1) shows. Software and the whole IEEE 802.15.4 standard with many configurations are modeled too.

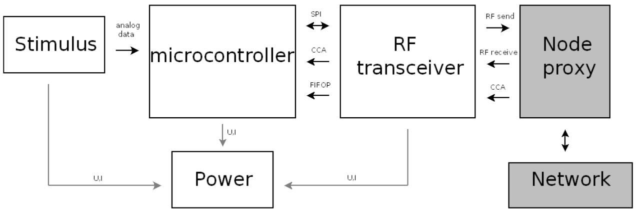


Figure 2. Node model architecture

The stimulus block generates analog sensor data towards the MCU. MCU and radiofrequency transceiver are modeled separately, so that designers can switch these inter-changeable devices. These two parts communicate through SPI (Serial Peripheral Interface) interfaces.

MCU is the central unit for processing and controlling purposes. In our typical case, MCU initializes the radiofrequency transceiver, then it reads (converts) data from sensor, and communicates data to radiofrequency transceiver. Switch between such architectures is done by changing some parameters. MCU model can for example switch from ATMEL ATMega128 to Microchip PIC16LF88. Figure 3 shows the generic FSM (Finite State Machine) we have implemented for MCU. Parameters depend on the MCU itself and from the radiofrequency transceiver (according for example to its hardware support of IEEE 802.15.4 or no).

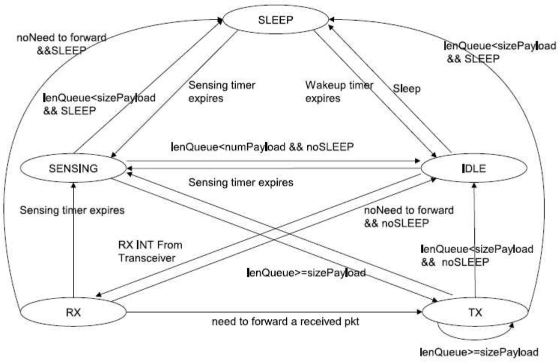

SLEEP
noNeed to forward &&SLEEP
lenQueue<sizePayload && SLEEP
lenQueue<sizePayload && SLEEP
Sensing timer expires
Wakeup timer expires
Sleep
lenQueue<numPayload && noSLEEP
SENSING
IDLE
Sensing timer expires
Sensing timer expires
noNeed to forward && noSLEEP
RX INT From Transceiver
lenQueue<sizePayload && noSLEEP
lenQueue>=sizePayload
RX
need to forward a received pkt
TX
lenQueue>=sizePayload

Figure 3. Generic FSM for MCU

Radiofrequency transceivers are modeled individually because of their complexity and wide differences (that would make difficult a generic FSM). Below are drawn two FSM examples of two well known radiofrequency transceivers.

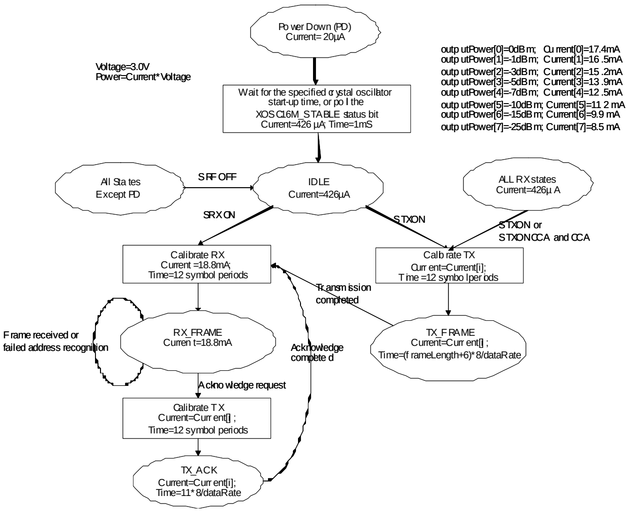

Power Down (PD)
Current= 20µA
Voltage=3.0V
Power=Current*Voltage
outputPower[0]=0dBm; Current[0]=17.4mA
outputPower[1]=-1dBm; Current[1]=16.5mA
outputPower[2]=-3dBm; Current[2]=15.2mA
outputPower[3]=-5dBm; Current[3]=13.9mA
outputPower[4]=-7dBm; Current[4]=12.5mA
outputPower[5]=-10dBm; Current[5]=11.2 mA
outputPower[6]=-15dBm; Current[6]=9.9 mA
outputPower[7]=-25dBm; Current[7]=8.5 mA
Wait for the specified crystal oscillator start-up time, or poll the XOSC16M_STABLE status bit
Current=426 µA; Time=1mS
All States Except PD
SRFOFF
IDLE
Current=426µA
ALL RX states
Current=426µ A
SRXON
STXON
STXON or STXONCCA and CCA
Calibrate RX
Current =18.8mA;
Time=12 symbol periods
Calibrate TX
Current=Current[i];
Time =12 symbol periods
Transmission completed
Frame received or failed address recognition
RX_FRAME
Current=18.8mA
Acknowledge completed
TX_FRAME
Current=Current[i];
Time=(frameLength+6)*8/dataRate
Acknowledge request
Calibrate TX
Current=Current[i];
Time=12 symbol periods
TX_ACK
Current=Current[i];
Time=11*8/dataRate

Figure 4. TI CC2420 non slotted CSMA-CA Finite State Machine

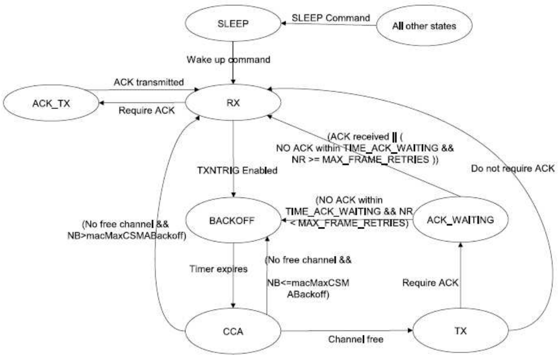

SLEEP
SLEEP Command
All other states
Wake up command
ACK transmitted
ACK_TX
Require ACK
RX
(ACK received || (NO ACK within TIME_ACK_WAITING && NR >= MAX_FRAME_RETRIES))
TXNTRIG Enabled
Do not require ACK
(NO ACK within TIME_ACK_WAITING && NR < MAX_FRAME_RETRIES)
BACKOFF
ACK_WAITING
(No free channel && NB>macMaxCSMABackoff)
Timer expires
(No free channel && NB<=macMaxCSMABackoff)
Require ACK
CCA
Channel free
TX

Figure 5. MRF24J40 non slotted CSMA-CA Finite State Machine

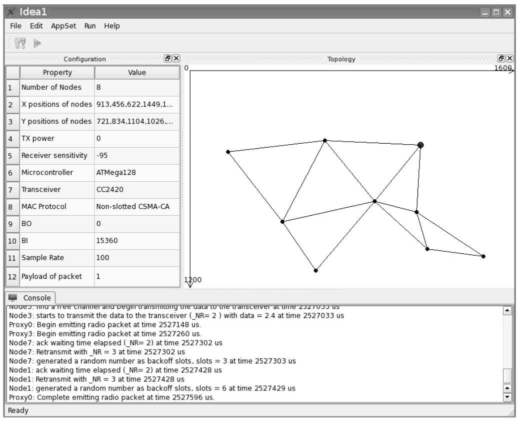
As a whole, several MCU and several radiofrequency transceivers can be selected; the current library is detailed in table I. Each MCU can be mapped to each radiofrequency transceiver. Each radiofrequency transceiver includes the whole IEEE 802.15.4 standard. As it has been previously published, all of these models have been validated with experimental measurements on many testbeds [11], and are 6% accurate.

TABLE I. MODELED DEVICES IN SIMULATOR LIBRARY

MCU	Radiofrequency transceivers
ATMEL ATMega128 Microchip 16LF88 Texas Instruments MSP 430	Texas Instruments CC2420 Texas Instruments CC1000 Microchip MRJ24J40

IV. SIMULATOR

The presented models can be used to simulate heterogeneous network communications at system level. To help non-specialists to use easily the simulation tool, we developed a graphical interface that is shown in the figure below.



	Property	Value
1	Number of Nodes	8
2	X positions of nodes	913,456,622,1449,1...
3	Y positions of nodes	721,834,1104,1026,...
4	TX power	0
5	Receiver sensitivity	-95
6	Microcontroller	ATMega128
7	Transceiver	CC2420
8	MAC Protocol	Non-slotted CSMA-CA
9	BO	0
10	BI	15360
11	Sample Rate	100
12	Payload of packet	1

```
Node3: find a free channel and begin transmitting the data to the transceiver at time 2527033 us
Node3: starts to transmit the data to the transceiver (_NR= 2 ) with data = 2.4 at time 2527033 us
Proxy0: Begin emitting radio packet at time 2527148 us.
Proxy3: Begin emitting radio packet at time 2527260 us.
Node7: ack waiting time elapsed (_NR= 2) at time 2527302 us
Node7: Retransmit with _NR = 3 at time 2527302 us
Node7: generated a random number as backoff slots, slots = 3 at time 2527303 us
Node1: ack waiting time elapsed (_NR= 2) at time 2527428 us
Node1: Retransmit with _NR = 3 at time 2527428 us
Node1: generated a random number as backoff slots, slots = 6 at time 2527429 us
Proxy0: Complete emitting radio packet at time 2527596 us.
```

Figure 6. Simulator graphical user interface

The user interface is composed of different sub-windows. A graphical viewer shows spatial position of nodes and the possible communications according to locations, power of transmission and sensitivity of receiver. Hardware parameters are some of selectable MCU and radiofrequency transceivers. One of the many IEEE 802.15.4 configurations and superframe parameters can be selected. Nodes sensors sampling rate and payload of packets can also be set. By clicking on the launch button in graphical interface, a SystemC simulation is launched in background. Simulation log is displayed in the bottom window of graphical interface, and a timing trace (Value Change Dump format: VCD) viewer is opened. Output log files are also generated. Concrete examples are given in the following section. From these results, we can explore design space for best solution.

V. HETEROGENEOUS SIMULATION RESULTS

Heterogeneous support in simulators with fine and accurate hardware and software models is necessary but few simulators support this feature [12][13]. One reason is the need of a complex organization of models. As test example, we simulated a 9 nodes network: 1 coordinator and 8 nodes composed of Microchip PIC16LF88 and ATMEL AVR ATMega128L MCU and Microchip MRF24J40 and Texas Instruments CC2420 radiofrequency transceiver, as specified in table II.

TABLE II. NODES DEVICES FOR TESTBED

WSN device	MCU	RF Transceiver
Coordinator	ATMega128	CC2420
Nodes 0..3	PIC16LF88	MRF24J40
Nodes 4..7	ATMega128	CC2420

Nodes sense the environment periodically and transmit data over the network. Each transmission (packet) includes 2 data bytes (payload). Sensor nodes enter sleep mode as long as they can, coordinator doesn't enter sleep mode. IEEE 802.15.4 classical data-rate (theoretically 250 Kb/s) is used. A non beacon CSMA-CA with no acknowledge is used, but all of the IEEE 802.15.4 can be chosen for more complex applications and wider exploration. Previous works on non heterogeneous (homogeneous) WSN with this simulation platform have already been published [14] to validate accuracy of the models by experimental measurements.

By clicking on the launch button in graphical interface, SystemC batch command is launched in background, and simulation log is displayed in the bottom window. Then, a trace (VCD) viewer is opened. A part of VCD trace in shown in Figure 7.

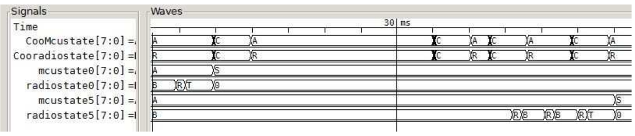


Figure 7. Extract of the output VCD file, focus on coordinator and nodes 0 and 5 (MCU and radiofrequency transceiver states)

We can observe the coordinator's and nodes' MCU and radiofrequency transceiver states. It is possible to monitor more signals in order to see the channel usage and data transfers from sensor to radiofrequency transceiver through MCU on each node, and data from radiofrequency transceiver to MCU on coordinator. It is then possible to monitor latency and packet delivery rate (PDR) from these curves. From log file, PDR and latency can be precisely displayed; energy can be read to draw graphs such as these in the following figures.

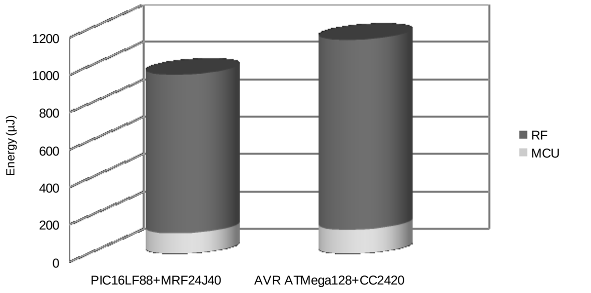


Figure 8. Heterogeneous nodes energy consumption

ISBN: 978-1-61208-207-3

Figure 8 shows energy partitioning between MCU and radiofrequency transceiver for 2 heterogeneous nodes (node 0 and node 5). We can also see the (relative) low impact of MCU energy compared to energy consumed by radiofrequency transceiver.

It is moreover possible to have finer granularity and to detail energy consumption of each block within hardware devices. Figure 9 shows energy spent during SPI communications, active (CPU), and sleep states. It is to note that radiofrequency transceiver impacts active duration of MCU. Indeed, in that example, CC2420 transceiver just modulates the packet, MCU has all in charge. It has for example to check for free channel, to use delays for backoffs, to generate IEEE 802.15.4 compliant packets, to take acknowledge in charge if activated, etc. More SPI communications are also required. On the other hand, MRF24J40 is a more autonomous circuit, as it supports all the aspects above in hardware, MCU is also less active.

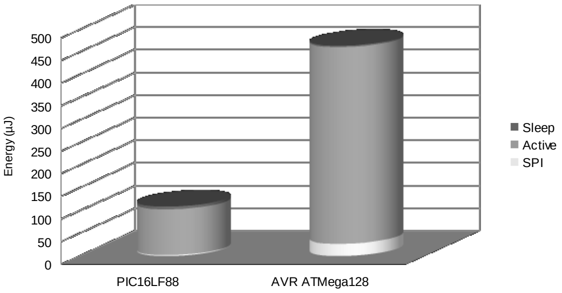


Figure 9. MCU energy consumption comparison

With the same fine granularity, it is possible to detail states of radiofrequency transceivers, as shown in Figure 10.

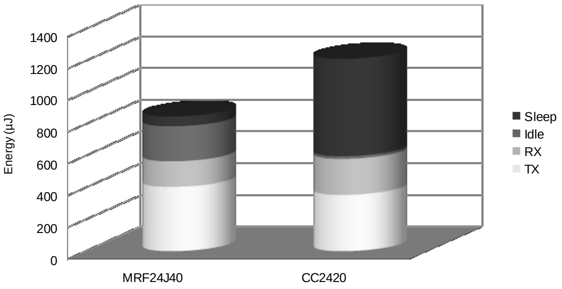


Figure 10. Radiofrequency transceiver energy consumption comparison

For each radiofrequency transceiver, this figure shows it is possible to monitor energy consumed during sleep mode, idle, receiving (RX) and transmitting (TX). According to the testbed (or real application) states durations, it is also possible to optimize total energy, with such a deep exploration.

VI. Conclusion

Heterogeneous support of our system-level simulator for Wireless Sensor Networks has been presented. This simulator is written in SystemC, which combines advantages of being a widely used language in electronic systems design flow, and permitting hardware and software modeling. The simulator graphical interface permits to easily configure a network and the sensor nodes characteristics, to obtain easy to read waveforms and easy to process output logs. As the whole IEEE 802.15.4 and many hardware devices are modeled, it is possible to simulate and compare IEEE 802.15.4 algorithms on many interchangeable (and parameterized) hardware devices (even by mixing them within a network) in order to run design space exploration. Classical network simulators outputs - Packet Delivery Rate (PDR) and packet latency - are accessible; as well as accurate timing, accurate and detailed power consumption of hardware devices.

References

[1] M. Horton and J. Suh, “A vision for wireless sensor networks”, IEEE Microwave Symposium Digest, 2005.

[2] S. Park, A. Savvides and M. B. Srivastava, “SensorSim: A Simulation Framework for Sensor Networks”, Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2000.

[3] P. Levis, N. Lee, M. Welsh and D. Culler, “Tossim: Accurate and scalable simulation of entire tinyos applications”, Embedded Networked Sensor Systems, 2003.

[4] M. Varshney, D. Xu, M. Srivastava and R. Bagrodia, “sQualNet: An Accurate and Scalable Evaluation Framework for Sensor Networks”, Information Processing in Sensor Networks, 2007.

[5] S. McCanne and S. Floyd, “Network Simulator NS-2”, http://ww.isi.edu/nsnam/ns, 2010.

[6] A. Varga, “The OMNeT++ discrete event simulation system”, European Simulation Multiconference, 2001.

[7] W. Du, D. Navarro and F. Gaffiot, “Towards a Taxonomy of Simulation Tools for Wireless Sensor Network”, International Conference on Simulation Tools and Techniques, 2010.

[8] F. Chen, I. Dietrich, R. German and F. Dressler, “An Energy Model for Simulation Studies of Wireless Sensor Networks using OMNeT++”, PIK Volume 32, Pages 133–138, 2009.

[9] B. Titzer, D. Lee, and J. Palsberg, “Avrora: Scalable sensor network simulation with precise timing”, Information Processing in Sensor Networks, 2005.

[10] F. Fummi, D. Quaglia and F. Stefanni, “A SystemC-based Framework for Modeling and Simulation of Networked Embedded Systems”, Forum on Specification and Design Languages, 2008.

[11] F. Mieyeville, W. Du, I. Daikh and D. Navarro, "Wireless Sensor Networks for active control noise reduction in automotive domain", 14th International Symposium on Wireless Personal Multimedia Communications, 2011.

[12] F. Fummi, G. Perbellini, D. Quaglia, and A. Acquaviva, “Flexible energy-aware simulation of heterogeneous wireless sensor networks,” Conference on Design, Automation and Test in Europe, 2009.

[13] L. Girod, T. Stathopoulos, N. Ramanathan, J. Elson, D. Estrin, E. Osterweil, and T. Schoellhammer, “A system for simulation, emulation, and deployment of heterogeneous sensor networks,” Int. Conf. on Embedded Networked Sensor Systems, 2004.

[14] F. Mieyeville, D. Navarro, W. Du and M. Galos, "Energy-centric simulation and design space exploration for Wireless Sensor Networks", CRC Press, Taylor & Francis Group, Book in Press, 2012.

Self-monitoring Reinforcement Metalearning for Energy Conservation in Data-ferried Sensor Networks

Ben Pearre and Timothy X. Brown
University of Colorado, Boulder, CO, USA
{benjamin.pearre,timxb}@colorado.edu

***Abstract*—Given multiple widespread stationary data sources such as ground-based sensors, an unmanned aircraft can fly over the sensors and retrieve their data via a wireless link. When sensors have limited energy resources, they can reduce the energy used in data transmission if the ferry aircraft is allowed to extend its flight time. Complex vehicle and communication dynamics and imperfect knowledge of the environment confound planning since accurate system models are difficult to acquire and maintain, so we present a reinforcement learning approach that allows the ferry aircraft to optimise data collection trajectories and sensor energy use *in situ*, obviating the need for system identification. We address a key problem of reinforcement learning—the high cost of acquiring sufficient experience—by introducing a metalearner that transfers knowledge between tasks, thereby reducing the number of flights required and the frequency of significantly suboptimal flights. The metalearner monitors the quality of its own output in order to ensure that its recommendations are used only when they are likely to be beneficial. We find that allowing the ferry aircraft to double its range can reduce sensor radio transmission energy by 60% or better, depending on the accuracy of the aircraft's information about sensor locations.**

***Keywords*-Sensor networks; data ferries; energy optimisation; reinforcement learning; metalearning**

I. Introduction

We consider the problem of collecting data from widespread energy-limited stationary data sources such as ground-based sensors. Our approach uses a fixed-wing unmanned aircraft (UA) to fly over the sensors and gather the data via a wireless link [1]. We assume that the UA has a known range limit and can be recharged/refuelled at a base station, and that the sensors may continuously generate data over long periods, so that the UA needs to ferry the data to a collection site over repeated flights. The goal is to trade energy used by the UA against energy saved by the sensor nodes. The system is difficult to model, so the challenge is to develop a model-free approach that can quickly learn to minimise the sensors' radio transmission energy subject to the UA's range constraints.

The problem may be subdivided into the following pieces: *Aircraft trajectory optimisation* seeks to discover a flight path over the sensor nodes (a so-called *tour*) that minimises some mission cost. We decompose this piece as follows:

- *Tour Design* decides in what order to visit sensor nodes of known location, or establishes a search pattern when the locations are unknown.
- *Trajectory Optimisation* finds a sequence of waypoints the UA should follow in order to visit the sensor nodes.
- *Vehicle Control* translates the waypoints into control surface and engine commands.

Radio energy optimisation consists of the following:

- *Radio Design* chooses radio hardware and protocols to support high-efficiency communication.
- *Power Management* varies the transmission power of nodes' radios during interaction with the ferry aircraft.

This paper focuses on the Aircraft Trajectory Optimisation and Radio Power Management layers. We assume that the tour is given and that the nodes' locations are known only approximately as when, for example, the sensors have been deployed from an aircraft. Vehicle control to track a set of waypoints requires an autopilot, whose behaviour is a complex function of the waypoints, weather, aircraft dynamics, and the control models within the autopilot. Similarly, communication system performance is a complex function of the radio protocols, antenna patterns, noise, and interference. We assume autopilot and communication systems are black boxes whose specific functionality is unknown to the upper layers. Only aggregate performance of the ferry system is reported to the learner.

In [2], we examined model-free minimisation of UA trajectory length. Here we extend the technique: since network lifetime or maintenance costs may depend on the energy reserves of the sensor nodes, we seek to minimise their transmission energy cost per bit.

Data ferries can be highly effective for reducing radio energy requirements. Jun et al. [3] compare ferry-assisted networks with hopping networks in simulation and finds that a ferry can reduce node energy consumption by up to 95% (further gains would have been possible with a broader configuration space). Tekdas et al. [4] reach a similar conclusion on a real toy network in which wheeled robots represent ferries. Anastasi et al. [5] consider the total energy requirement per message including overhead associated with turning a node's radio on in order to search for a fixed-trajectory ferry. Ma and Yang [6] optimise the lifetime of nodes by choosing between multi-hop node-to-node routing and changing the ferry's route and speed. Optimal solutions under the trade-off between energy use and latency have been examined for a fixed ferry trajectory [7], and adaptively with a known radio model and simple flight dynamics [8]. In [9], a node learns whether to transmit to the ferry or wait depending on the anticipated trajectory; range affects transmission power. Anastasi et al. [10] review techniques for energy minimisation for both ferried and general sensor networks.

Past work on ferried networks has made various assumptions about ferry and communication system dynamics, which we broadly categorise as follows. In *Visit* models, the ferry exchanges all data upon visiting a node [11]–[14]. *Communication radius* models exchange all data instantaneously below a certain range [4], [6], [15]–[18]. A learning variant to this model is found in [19]: planning assumes a communication radius, but the ferry also exchanges data opportunistically, possibly allowing improvements to the trajectory. *Variable rate* models base rate on communication range [20], [21]. Stachura et al. [22] assume probabilistic packet loss based on distance, achieving the same effect. Mobile nodes are treated in [23], which uses a model of a UA equipped with a beamforming antenna to plan trajectories that maximise the signal-to-noise ratio (SNR) to each node.

Models are necessarily approximate, but inaccuracies can lead to poor performance in the field. The difficulty of generating and updating sufficiently accurate models under possibly changing conditions motivates our question: can a sensor network system simultaneously learn to optimise both the aircraft's flight path and the sensors' radio policies, in a reasonable time, directly on a radio field? The goal is to provide a UA with approximate information about the geometry of a sensor network, and to have it improve its performance rapidly and autonomously. We focus on minimising the number of flight tours required to get a good solution, since computational costs are comparatively minor.

Our contributions are as follows:

- We demonstrate the feasibility of a reinforcement learning approach for rapid discovery of energy-saving network policies that trade UA flight time for sensor energy. The policies are learned without a system model and despite potentially inaccurate sensor node locations, unknown radio antenna patterns, and ignorance of the internals of the autopilot.
- We introduce a reinforcement metalearner that learns to speed up and stabilise the performance of the learner, and transfers acquired knowledge about the policy optimisation process to unseen problems.
- A poorly trained metalearner may degrade the learner's performance, and should not be allowed to influence learning. We introduce a method of monitoring the performance of the metalearner without allowing it to affect network behaviour before it has been trained sufficiently.
- We show our two independent optimisers—waypoint location and transmission power policy—can operate simultaneously on the same sampling flights.

Our learning framework quickly produces trajectories that exploit the limits of ferry endurance. For example, when the allowable flight length is twice that of a handcoded reference trajectory and accurate sensor location information is available, the system learns to reduce sensor communication energy by roughly 60% after a few dozen flights, and when the sensor location information is approximate, the learners do even better. More important than this specific result is the development of a general technique that allows data-ferry networks to efficiently optimise their performance *in situ*.

Section II describes our radio model. Section III describes how our simulated autopilot control policies interpret waypoints. Section IV reviews the learning algorithm we use and describes how we apply it to our sensor energy optimisation problem. In Section V, we develop our metalearner. Section VI presents our results. Section VII concludes.

II. Radio environment

Our goal is to evaluate the use of model-free optimisation in a complex, unknown radio environment. We introduce a radio model that incorporates several complicating factors that are rarely considered: variable-rate transmission, point noise sources, and directional antennas. This model extends that introduced in our previous work [2] only by giving the nodes dipole antennas.

The signal to noise ratio at node a from node b is given by

$$\mathrm{SNR}_{ab} = \frac{P(a,b)}{N + \sum_i P(a,n_i)} \tag{1}$$

$P(a,b)$ is the power received by node a from node b, N is background noise from electronics and environment, and n_i are noise sources. The power between a and b is usually computed as

$$P(a,b) = \frac{P_{0,a} d_0^{\epsilon}}{|X_a - X_b|^{\epsilon}} \tag{2}$$

for reference transmit power $P_{0,a}$, reference distance $d_0 = 1$, distance between transmitter and receiver $|X_a - X_b|$, and propagation decay ϵ. However, antenna shape and radio interactions with nearby objects make most antennas directional, so the orientations of the antennas affect power. We model the aircraft's antenna as a short dipole with gain $1.5 = 1.76$ dBi oriented on the dorsal-ventral axis of the aircraft, yielding a toroidal antenna pattern. We model the nodes' fields similarly with random fixed orientations, so we adjust the power computation in (2) to:

$$P'(a,b) = \sin^2(\xi_{ab}) \sin^2(\xi_{ba}) P(a,b) \tag{3}$$

where ξ_{xy} is the angle between antenna x's pole and the direction to y. This depends on the vector between the UA's and node's positions ($\in \mathbb{R}^3$), the aircraft's orientation ($\in \mathbb{R}^3$), and the transmitter's orientation, although the latter is assumed not to change. Here we consider only constant-altitude trajectories.

In order to evaluate (3), we require the UA's position and orientation. A full dynamical simulation of the aircraft is unnecessarily complex for our purposes, so we assume that course and heading $\dot{\phi}$ are the same (yaw = 0), pitch = 0, and roll $\psi = \frac{\pi}{2} \tanh 2\dot{\phi}$, which varies between 0 for a straight course and ±54° for our maximum turning rate of $\dot{\phi} \simeq 0.347$ rad/s (i.e. the UA flies a complete circle in 20s).

We use the Shannon-Hartley law [24] to compute the data transmission rate between transmitter a and receiver b:

$$R_{ab} = \beta \log_2(1 + \mathrm{SNR}_{ab}) \tag{4}$$

This assumes that data rate varies continuously. The hardware may use discrete rates that are chosen according to current SNR conditions, but [21] indicates that the difference in trajectories and performance outcomes between continuously variable and the discrete rates of 802.11g may be negligible for this type of problem.

This model ignores many characteristics of a real radio environment such as occlusion, reflection, higher-order directionality, and multipath propagation. Moreover, we do not simulate obvious protocol modifications that would allow other sensor nodes to cease transmission and thereby reduce interference with the active node. However, in part due to the latter omission, the model produces fields that have irregularities similar to those that occur in real radio environments, and thus it meets our aim of having a complex simulation environment within which we can test whether the aircraft can learn *in situ.*

III. Autopilots

The aircraft is directed to follow some trajectory through an autopilot control policy.

A. Reference autopilot control policy

The *Reference autopilot* is borrowed from [21] and does not learn. It assumes that the aircraft has an estimate of the locations of the sensor nodes (although these can be difficult to discover [25]). While communicating only with the target node the aircraft flies at constant speed v towards the tangent of a circle of minimum turning radius about the node's nominal location, circles it at the maximum turning rate ω until D bytes are received, and then proceeds to the next node. The result is the Reference trajectory.

B. Learning autopilot control policy

The *learning autopilot* places a GPS waypoint at the assumed location of each sensor node (we will assume that node identities and approximate locations are known during tour initialisation, although the assignment could instead occur on the fly as nodes are discovered). The aircraft collects data from any node opportunistically while flying towards the tangent of a minimum-turning-radius circle about the waypoint, and then if necessary circles the waypoint exchanging data only with the assigned node until it has collected sufficient data. The true node location and the waypoint location may differ; final waypoint positions are learned as described next.

IV. Learning

Policy gradient reinforcement learning (PGRL) [26] consists of a family of techniques for model-free optimisation of control policies. Key elements are a policy gradient estimator, a reward function, and a policy representation. We introduce these by describing and applying the well-known "episodic REINFORCE" estimator to our waypoint-placement problem, and then introduce the more sophisticated radio transmission power policy.

A. Policy Gradient Reinforcement Learning (PGRL)

A policy $\pi(s, u; \theta) = \Pr(u|s; \theta)$ defines the probability of choosing action u given state s under the policy parametrised by $\theta \in \mathbb{R}^n$. The expected reward averaged over all states and actions under a policy $\pi(s, u; \theta)$ is denoted $J(\pi(s, u; \theta))$ or abbreviated as $J(\theta)$. PGRL techniques are ways of estimating the gradient of the expected reward: $\widehat{g_\theta} = \widehat{\nabla_\theta J(\theta)}$.

Our task can be broken down into distinct "trials", each consisting of a complete execution of $\pi(\theta)$ over some bounded time (e.g., the aircraft flying a complete tour τ) and receiving reward r at the end. During a trial, the policy defines a probability distribution over the action chosen at any point. Assume that the controller makes some finite number H of decisions u_k at times $t_k, k \in 1 \dots H$ during a trial; discretising time in this manner makes it possible to compute the probability of a trajectory under a policy as the product of the probabilities of each (independent) decision at each time t_k. So $\Pr(\tau|\theta) = \prod_{k=1}^{H} \Pr(u_k|s_k; \theta)$.

To optimise θ, we estimate the gradient using stochastic optimisation's "likelihood-ratio trick" [27] or reinforcement learning's "episodic REINFORCE" (eR) [26], [28] with non-discounted reward. Each element ∇_{θ_i} of the gradient is estimated as:

$$\widehat{g_{\theta_i}} = \left\langle \left(\sum_{k=1}^{H} \nabla_{\theta_i} \log \Pr(u_k|s_k; \theta) - \mu_{\Sigma_\nabla} \right) \left(\sum_{k=1}^{H} \gamma^{t_k} r_k - b_i \right) \right\rangle \tag{5}$$

where μ_{Σ_∇} is the mean over trials of the $\sum \nabla_{\theta_i}$ terms, b_i is a "reward baseline" for element θ_i, computed as the inter-trial weighted mean of rewards, using the per-trial weight $\left(\sum_{k=0}^{H} \nabla_{\theta_i} \log \Pr(u_k|s_k; \theta)\right)^2$ [26], and $\langle \cdot \rangle$ is the average over some number N of trajectories. The term μ_{Σ_∇} is the mean over trials of the $\sum \nabla_{\theta_i}$ terms; this term does not appear in [26] but is included here in order to reduce the variance of the "characteristic eligibility"; we found it to further improve the gradient estimation process. We postpone discussion of the temporal discount factor γ to §V-C; for now consider $\gamma = 1$.

Once a policy gradient estimate $\widehat{g_\theta} = \widehat{\nabla_\theta J}$ for episode e is obtained, we take a step of some length α in that direction,

$$\theta_{e+1} = \theta_e + \alpha \frac{\widehat{g_\theta}}{|\widehat{g_\theta}|} \tag{6}$$

thus creating a new policy. The gradient estimation and update may be repeated until a design requirement is met, until convergence to a local maximum, or forever to track a time-varying environment. The theoretical guarantee of convergence to a locally optimal policy is only available if α decreases over time. That guarantee is useful, but does not directly apply to learners operating in a changing environment and is unnecessary for our tasks.

B. Reward

The reward function (or its additive inverse, the cost function) expresses the desiderata of solutions as a scalar quantity.

We seek the policy that leads to the lowest total transmission power subject to a maximum desired tour length $d_{\max}$, which indicates that we are nearing the endurance limit of the aircraft. We use the following:

$$r = -\left(\max(0, d - d_{\max})^{\varrho} + \sum_{j \in \text{nodes}} \varphi_j \sum_{k=1}^{H} P_{jk} \Delta t\right) \tag{7}$$

d is the current trajectory path length, $d_{\max}$ is the soft maximum range of the aircraft, ϱ controls the severity of the soft maximum distance penalty, P_{jk} is the transmission power of node j at timestep k of length Δt, and φ_j is a weighting for the value of energy for node j. Note that d is not penalised until the aircraft exceeds $d_{\max}$.

The aircraft's autopilot may be programmed to return to base by some hard length constraint $d_{\text{hardmax}} > d_{\max}$, and a cost for underrunning data-collection objectives could be included in (7). This produces similar results in our simulations but complicates our explanation.

C. Waypoint placement

We consider a sequence of nodes that need to be visited in some order $\{a, b_1, \ldots, b_n, c\}$ that was determined by a higher-level planner [13], [17]. The aircraft must fly a trajectory that starts at a and ends at c and allows exchange of D_j bytes of data with each of the n sensor nodes b_1 to b_n. Thus we seek the path $a \to c$ that minimises (7). That sufficient data are collected from each node is guaranteed by the autopilot policies (§III).

Trajectory policies for the autopilot are implemented as sequences of constant-altitude waypoints. So for m waypoints, the waypoint policy's parameter vector $\theta = [x_1\ y_1\ x_2\ y_2 \ldots x_m\ y_m]^T$. In order to be used by (5) the controller adds noise such that $\Pr(\tau|\theta)$ can be computed. In a real system, actuator noise or autopilot error E can be used for this purpose if $\nabla_\theta \log \Pr(u + E \mid \theta)$ can be computed, but in our simulations we simply add Gaussian noise directly to the waypoint locations at the beginning of each tour:

$$\begin{aligned} u &\sim \mathcal{N}(\theta, \Sigma) & (8) \\ \nabla_\theta \log \Pr(u|s;\theta) &= \frac{1}{2}\left(\Sigma^{-1} + \Sigma^{-1\,T}\right)(u - \theta) & (9) \end{aligned}$$

D. Radio transmission power

The data-ferrying approach allows sensors to communicate with distal base stations without the need for high-powered radios, but the energy that nodes spend in communicating with the ferry is still non-negligible [3], [4]. Recall the data rate from Section II:

$$R_{ab} = \beta \log_2(1 + \text{SNR}_{ab}) \tag{10}$$

The derivative of $\frac{\text{rate}}{\text{power}}$

$$\nabla_P \frac{\beta}{P} \log_2\left(1 + \frac{P}{N}\right) = \frac{\beta}{N\,P\,\log(2)\,\left(1 + \frac{P}{N}\right)} - \frac{\beta \log\left(1 + \frac{P}{N}\right)}{P^2 \log(2)} \tag{11}$$

is negative whenever

$$\frac{P}{P+N} < \log\left(\frac{P+N}{N}\right) \tag{12}$$

which is true except at $P = 0$. So while reducing power results in a lower energy cost per bit, lower transmission rates require longer trajectories. Given an externally defined trade-off between ferry trajectory length and node energy savings, when should a sensor transmit, and at what power?

The difficulty of predicting the SNR between transmitter and aircraft again suggests reinforcement learning. We assume that at each timestep a sensor can transmit with power $P \in [0, P_{\max}]$, that it occasionally sends short probe packets at $P = P_{\max}$, and that the aircraft's radio can use this to measure the current maximum SNR and provide instructions to the node. These packets are too brief to transmit sensor data or use much power, so we do not model them explicitly. Here, too, the learning approach will silently optimise around such quirks of real hardware.

The *power policy* is a learned function that controls the power a node uses to transmit given a reported SNR, given in dB. Our desired behaviour is to transmit at a target power $P_{\text{target}} \leq P_{\max}$ whenever the probed SNR exceeds some threshold R_T. So, the policy has action $u = P$, state $s = \text{SNR}_{\text{probed}}$, and is parametrised by $\theta = [P_{\text{target}}, R_T]$. Reinforcement learning requires exploration noise, so at each timestep the policy π draws the actual transmission power P from a Gaussian (truncated on $[0, P_{\max}]$) whose mean is taken from a sigmoid of height P_{target}:

$$\begin{aligned} \pi(s, u; \theta) &= \Pr(u|s;\theta) & (13) \\ &\sim \mathcal{N}\left(\frac{P_{\text{target}}}{1 + e^{\phi(R_T - s)}}\ ,\ \sigma\right), \text{ truncated on } [0, P_{\max}] \end{aligned}$$

When $s = R_T$, the mean transmit power is 50% of P_{target}, going to 100% as s increases above R_T and vice versa, thus implementing the desired behaviour with exploration. ϕ controls the sigmoid's width, and σ controls Gaussian exploration. For example, when $P_{\text{target}} = P_{\max}$ and R_T and σ are small, the policy mimics the full-power Reference policy.

The policy's derivatives are:

$$\nabla_\theta \log \pi(s, u; \theta) = \begin{bmatrix} \dfrac{u - \frac{P_{\text{target}}}{1 + \mathrm{e}^{\phi\,(R_T - s)}}}{\sigma^2 \left(1 + \mathrm{e}^{\phi\,(R_T - s)}\right)} \\[2ex] -\dfrac{P_{\text{target}}\,\phi\,\mathrm{e}^{\phi\,(R_T - s)} \left(u - \frac{P_{\text{target}}}{1 + \mathrm{e}^{\phi\,(R_T - s)}}\right)}{\sigma^2 \left(1 + \mathrm{e}^{\phi\,(R_T - s)}\right)^2} \end{bmatrix} \tag{14}$$

Unlike the waypoint-placement policy, this one is closed-loop: sensing packets detect SNR, which informs the choice of action u (transmission power) at each timestep. Thus we use the full loop-closing capabilities of the episodic REINFORCE algorithm of §IV-A. This policy and the waypoint-placement one run in parallel, using the same flights to estimate their reward gradients.

As a concession to practicality, the power policy incorporates a failsafe mechanism: when the aircraft's soft maximum

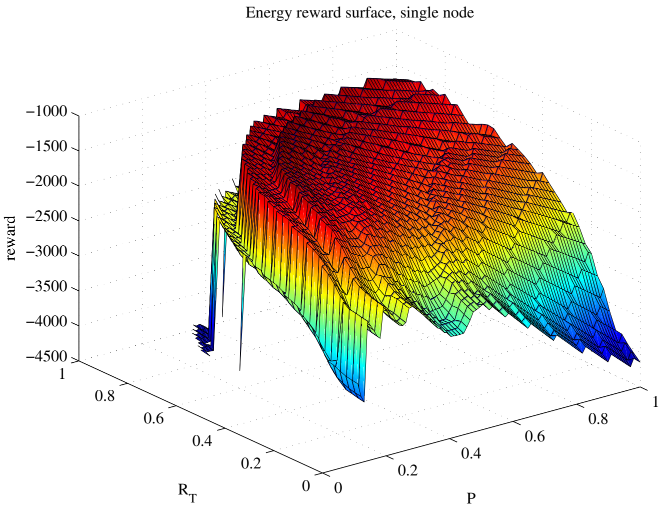


Fig. 1. Energy reward landscape for an example single node, with fixed waypoint position. As transmission power P and threshold SNR R_T change, energy savings may lead to greater reward up to a certain point. But the high cost of exceeding the aircraft's range constraint creates a steep "cliff" in the reward landscape.

range has been exceeded, radio power is set to 100%, ensuring that the UA does not become stuck in nearly infinite loops.

V. Metalearning

Waypoint location optimisation is fairly straightforward [2]. However, our energy reward function (7) is highly nonlinear with respect to the power policy parameters in the vicinity of the optimal solution. Figure 1 shows a portion of the reward landscape for trajectories looping a typical node. When R_T is small and P is near 1, the gradient is not difficult to estimate—exploration noise will generally average over the ridges and valleys. However, in the vicinity of the optimal solution (the crest of the hill), there is a steep cliff: if R_T becomes too high or P becomes too low and the aircraft must remain near the sensor for a long time in order to collect all the data, which activates the aggressive length-overrun term of the reward function.

Conventional PGRL repeatedly estimates the reward gradient near the current policy and takes a hillclimbing step. Near the optimum, hillclimbing updates can result in the learner taking a step off the cliff, or "cliff-jumping". Furthermore, the cliff contains local regions in which a problematic reverse-sloped ledge structure is apparent—it is possible for a local gradient estimate to suggest a step further off the cliff. Confidence regions can mitigate this problem, but they generally fail to re-use information acquired during past steps (but see [29] for a counterexample). The problematic structure in the reward landscape motivated the development of a technique to encode knowledge of the process of optimising on reward landscapes like ours.

Consider the intuition: when a trial leads to an overly long trajectory, it is generally helpful to increase power or allow transmission at lower SNR. Conversely, for a trajectory that does not use the aircraft's full range node energy can be reduced by using lower power. The PGRL gradient estimation finds policy updates that, on average, tend to obey these heuristics, but the microstructure and the abrupt cliff near the global optimum frequently lead to poor updates. Our goal in developing a metalearner is to give the UA the ability to use experience with past problems to improve learning speed and robustness on new problems and automatically capture such heuristics. We investigate the following questions:

- Can a metapolicy that encodes knowledge about optimising policies in this domain be learned through experience?
- Can such a metapolicy transfer knowledge between problems?
- Can we monitor the quality of the metapolicy's recommendations in order to prevent a poor metapolicy from adversely affecting the optimisation process?
- Can the metapolicy be used to speed or stabilise the learning of energy-saving policies for sensor networks?

A. Metapolicy

Our energy "metapolicy" examines each trajectory and produces a guess as to the best update, $\Delta\theta$, to the base power policy's parameters $\theta = [P_{\text{target}}, R_T]$. For each element θ_i of θ, we use a simple neural network, a so-called single-layer perceptron (see [30]) with one input—the fraction of allowed aircraft range used—and two outputs—suggested changes to the base policy's two parameters:

$$\begin{aligned}
s_\mu &= \frac{d}{d_{\max}} && (15)\\
u_\mu &= \Delta\theta_i && (16)\\
\pi_\mu(s_\mu, u_\mu; \Theta_i) &= \Pr(u_\mu | s_\mu; \Theta) && (17)\\
&= \mathcal{N}\left(\tanh\left(\Theta_{1,i} s_\mu + \Theta_{2,i}\right),\ \ \sigma_\mu\right) &&\\
\nabla_{\Theta_i} \log \pi(s_\mu, u_\mu; \Theta) &= \begin{bmatrix} s_\mu Z \\ Z \end{bmatrix} && (18)\\
\text{where } Z &= && (19)
\end{aligned}$$

$$\sigma_\mu^{-2}\left(u_\mu - \tanh\left(s_\mu \Theta_{1,i} + \Theta_{2,i}\right)\right)\left(1 - \tanh(s_\mu \Theta_{1,i} + \Theta_{2,i})^2\right)$$

If the perceptual space is enriched with other inputs or requires a richer representation, other models can be used. Note, however, that more complex models with more parameters increases the number of runs necessary for learning a good metapolicy.

B. Metareward

The metalearner's objective is to learn a metapolicy that takes as input a policy operating on a trajectory, and outputs an "action" consisting of an improvement of the base policy's parameters. So as our metareward r_μ we choose the reward improvement between trials:

$$r_\mu = r_i - r_{i-1} \tag{20}$$

where r_i is the base reward received on trajectory i.

C. Time-discounted credit

The metalearner receives μ-reward (20) after every μ-action, and each μ-action also—to a lesser extent—affects future μ-states and thus potential μ-rewards, so it would be appropriate to use a time-discounted eligibility ($\gamma < 1$ in (5)). But further improvements are to be gained by using a more sophisticated gradient estimator, which we introduce here:

1) G(PO)MDP: (Here we drop the μ-prefix, as this section describes a well-known general technique.) In reinforcement learning, when an action u is taken at time t_u and a reward r is received at future time t_r, the action is assigned *credit* for the reward based on an estimate of how important the action was in producing the reward. In eR (§IV-A), greater weight may be given to rewards received early in the episode than on those received later, modulated by the term γ^{t_k}, $0 < \gamma \leq 1$ in (5). G(PO)MDP [31] uses the separation in time between t_u and t_r to assign credit in proportion to $\gamma^{t_r - t_u}$, $t_u < t_r$. We use G(PO)MDP as described in [26]. The gradient estimator is related to (5):

$$\widehat{g_{\theta_i}} = \left\langle \sum_{p=0}^{H} \left(\sum_{k=0}^{p} \nabla_{\theta_i} \log \Pr(u_k | s_k; \theta) \right) \left(\gamma^{t_k} r_k - b_i \right) \right\rangle \qquad (21)$$

We use the optimal baseline b_i shown in [26].

2) Sliding trajectory windows: As presented above, (21) learns from rewards received early in the trajectory but not later, since γ^t drives the value of later rewards to 0. Therefore we break a trajectory into sequences of $\langle s_\mu, a_\mu, r_\mu \rangle$, with one sequence starting at each timestep, and present those as separate trajectories to (21). Sequence length n is chosen such that $\gamma^n \geq 0.05 > \gamma^{n+1}$: the terms beyond this increase computational burden without significantly improving accuracy.

D. **PGRL+μ:** *Combining gradient and metapolicy updates*

Changes to the base policy's θ can come from the base PGRL estimator (§IV-D) after every epoch, or from the metapolicy (17) via u_μ after every trial. When the base PGRL estimator produces an estimate, we use it to adjust θ. But we can also pretend that it came from π_μ, and use it as u_μ for the computation of $\nabla_\theta \log \pi(s_\mu, u_\mu; \Theta)$. Thus both the PGRL and the π_μ updates and metarewards can be used to form the μ-trajectory for (21).[1] Although tuning can improve performance, for expository simplicity we set the magnitudes of all updates to be the same.

E. **Autodetect metalearner:** *Monitoring metapolicy quality*

Early in training, the metalearner can give poor advice, leading to high-cost policies. If ample non-mission training time is allocated, then such runs do not pose a problem. In our single-node example, roughly 50 runs of 100 trials each were required before the metapolicy reliably improved upon PGRL. Our hope is that any knowledge encoded by the metalearner can be transferred between tasks and that therefore metapolicy training time for any actual scenario may be low, but here we show how the metalearner can be trained, tuned, and tested without significantly interfering with the performance of a live network.

After each epoch, PGRL adjusts θ by some amount $\Delta_\nabla \theta$, and the following trial yields a μ-reward (20). If instead it had been the metalearner that had recommended the same update $\Delta_\mu \theta = \Delta_\nabla \theta$, the same reward (on average) would have been observed. Moreover, when the two recommend "opposite" changes to θ, it is more often the case than not that the reward earned by the metalearner would at least have had the opposite sign to that actually seen. Thus it is possible to make an educated guess as to whether the metapolicy's output would have led to an improvement in the base policy, without actually making the suggested change.

We define the *update disagreement* δ as the unsigned angle between $\Delta_\nabla \theta$ and $\Delta_\mu \theta$. We estimate metapolicy quality by tracking the average disagreement between base policy and metapolicy updates. For each policy update, we score the comparison between the metapolicy to PGRL as follows:

∇ good:	$\delta \leq \frac{\pi}{2}$	1
$r_\mu > \tau$	$\delta \geq \frac{3\pi}{2}$	-1
∇ bad:	$\delta \leq \frac{\pi}{2}$	-1
$r_\mu < -\tau$	$\delta \geq \frac{3\pi}{2}$	1

Our estimate of metapolicy quality is the mean of the trial-by-trial scores during a run, which is roughly equivalent to the negative of the slope of a linear regression of metareward vs. disagreement, but offers better numerical properties: the result is bounded, which eliminates the effect of outliers due to unusual exploration noise; it discards the ambiguous cases in which the PGRL and δ differ by around $\frac{\pi}{2}$; and it produces a valid result when the metapolicy and policy [dis]agree perfectly. τ was roughly hand-tuned to yield a positive metareward quality at, on average, the same time as the standard PGRL+μ metalearner started to outperform PGRL, yielding $\tau = 1000$. We perform exponential smoothing (base 0.9) across runs on the result in order to use more available information.

VI. Results

We generate random data-ferrying *problems* each of which consists of a random position and orientation for each sensor. At each *timestep* the aircraft flies some distance towards the next waypoint, measures the current SNRs via probe packets, and requests some data from a node at the power indicated by the power policy. A *trial* is a single complete flight over the radio field. An *epoch* is a small number of trials, after which we estimate the policy gradients and update the policies. A *run* is an attempt to optimise radio power and waypoint position policies for a given *problem*, and here consists of 100 trials. For each problem we generate a new random radio field and re-initialise the policies, but not the metapolicies. Although it is possible to learn as soon as we have enough trials to produce a gradient estimate, for simplicity we instead hold

[1] While the base gradient update provides a legitimate value for u_μ and hence $\nabla_\theta \log \pi(s_\mu, u_\mu; \Theta)$, it means that the forward roll-out is off-policy. This is theoretically interesting, but in practice, for our application, not problematic.

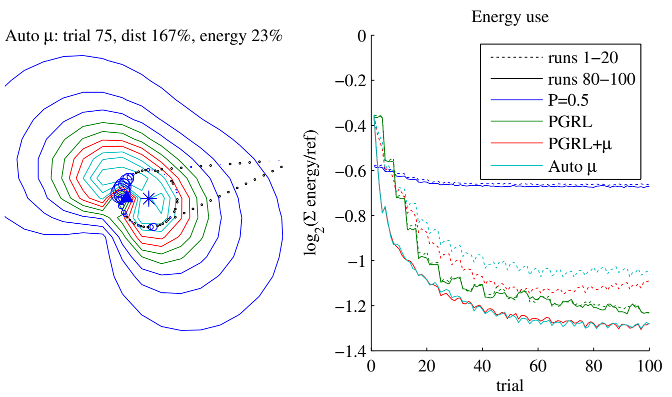


Fig. 2. Learning to minimise energy use for the single-node case. *Left:* sample trajectory plotted in space, superimposed over reference rate contours that show what the aircraft would see in flat level flight (not what it actually sees as it steers and banks). The aircraft starts at ⋆; the waypoint is at ∗. Circle size is proportional to data rate. *Right:* energy use for the three algorithms for the current experiment, averaged over the last few runs, which measures the performance gains possible from a well-trained metalearner.

the metapolicy's parameters Θ constant during each run. An *experiment* is a set of 100 runs during which the metalearners have the opportunity to adapt. For each experiment we re-initialise the metapolicy parameters. To generate the graphs, we average over 50 experiments.

We will compare the metalearners to two non-learning approaches and to PGRL:

Reference is the non-learning autopilot defined in §III-A.

Half-power learns waypoint placement as described in §IV-C, but instead of learning the power policy, sets $P = \frac{P_{\max}}{2}$. This is guaranteed to increase trajectory length by a factor ≤ 2.

PGRL uses the Learning autopilot and the conventional PGRL approach described in §IV-D, without the metalearner.

Parameters: The aircraft flies at speed $v = 1$ at altitude $z = 3$. The maximum turning rate of $\omega = 20°/s$ yields a turning radius $r = \frac{v}{\omega} \approx 2.9$. Radios use $P_{\max} = 150$ and bandwidth $\beta = 1$, and the background noise $N = 2$. Each sensor's data requirement *req* = 20. These parameters do not qualitatively affect the results, and can be replaced by appropriate values for any given hardware. For waypoint placement the learning rate $\alpha = 0.5$ and the exploration parameter $\sigma = 1$. The power policy uses $\alpha = 0.3$, $\sigma = 0.2$, $\phi = 1, \varphi_j = 1 \forall j$. Policy gradient estimates for waypoint placement and energy are computed and applied every 4 trials (one epoch) for reasons described in [2]. Metapolicy gradient estimates are computed and applied after each run as described in §V-C. The metalearner's temporal reward discount $\gamma = 0.25$.

A. Learning from base gradient, Metalearning

Figure 2 shows the energy use of our learners on a single node over the course of 100 trials. The behaviour of the metalearners' early learning is illustrated by performance plots over runs 1–20. We contrast this with the performance of well-trained networks over runs 80–100. All performance graphs show $\log_2$ of the ratio of performance compared to Reference.

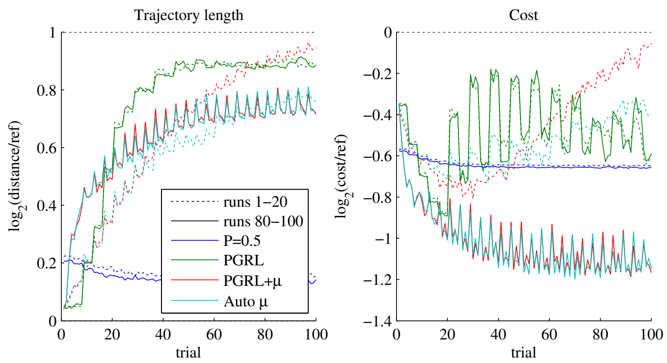


Fig. 3. Further learning details for the 1-node case. *Left:* trajectory length. *Right:* aggregate cost vs. trials relative to Reference (averaged over the last few runs of all experiments).

In our scenarios, **Half-power** (or "P=0.5") consistently reduces sensor energy requirements to about $2^{-0.65} \simeq 65\%$ of Reference in exchange for a 10% length increase, although this varies with data requirement—for high requirements (e.g., req > 100) Half-power reduces energy use by only 20% with a 60% increase in trajectory length. Despite the fixed power policy, energy use decreases slightly over time as waypoint positions are optimised.

Figure 3 shows how the learners' behaviours change during each run as the waypoint and energy policies are refined. After 20 or so trials, **PGRL** performs very well (Figure 3, Cost), but as it nears the optimal solution it falls into a cycle of discovering and rediscovering the cliff when random exploration steps take trajectory length over $d_{\max}$, resulting in frequent high-cost trajectories. It still outperforms Reference and untrained (runs 1–20) PGRL+μ, but Half-power is superior. In contrast, the trained (runs 80–100) metalearners outperform PGRL both early in each problem (the first few trials) during which the metapolicies rapidly push the policies towards lower energy use, and later, where they almost completely eliminate cliff-jumping.

The higher-level time-varying behaviour shown by the metalearners can be seen in the difference between runs 1–20 and 80–100, and more explicitly in Figure 4. As the metalearners observe the learners solving new problems, they refine their μ−policies, yielding performance that improves from run to run. Figures 2 and 3 show snapshots of per-trial performance averaged over runs 1–20 and 80–100, and the improvement of average per-trial performance over runs can be seen in Figure 4.

PGRL+μ usually outperforms the non-meta approaches by a significant margin after about 30 runs, allowing discovery of policies that use only 40% of the sensor energy of Reference and 65% that of Half-power, while seldom exceeding the aircraft's soft range limit. Perhaps surprisingly, even with such a simple representation we found that handcoding a metapolicy that outperforms the learned one was not easy. However, Figure 4 shows that early in metapolicy training, especially for the first 20 runs, PGRL+μ performs poorly, producing

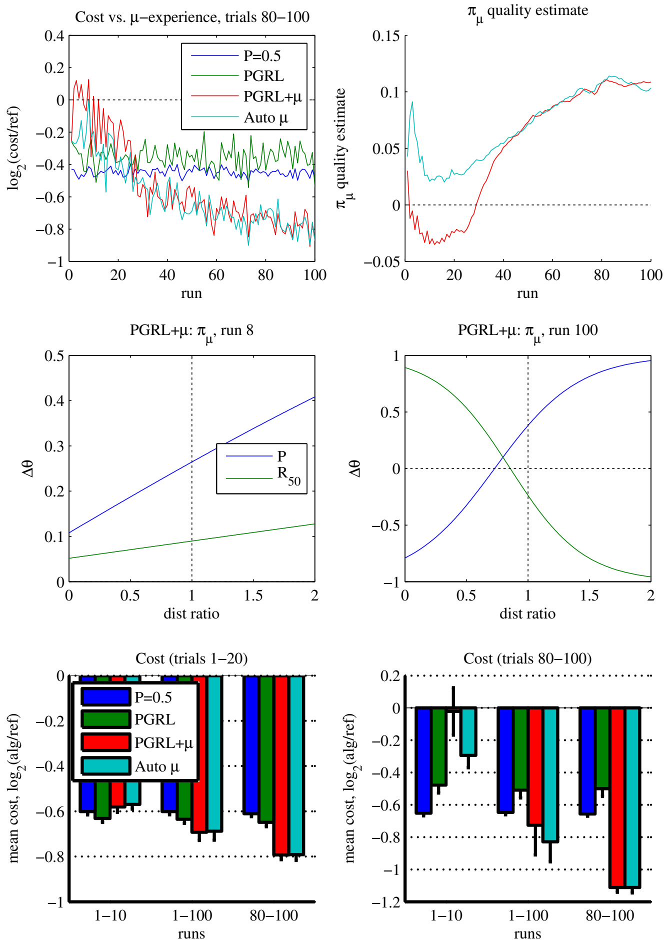


Fig. 4. Metalearning on 1 node. *Top left:* Trained (trials 80–100) performance of the algorithms over runs, during which the metalearners improve with experience. *Right:* The estimate of the quality of the metalearner's output through time, both for PGRL+μ, which does not use the information, and for Autodetect, which does. *Middle:* representations of the metapolicies' average actions $u_\mu = \Delta\theta$ vs. the observation of the previous trajectory, "dist ratio" $s_\mu = \frac{d}{d_{\text{max}}}$, shown early in μ–training (run 8) and late (run 100). *Bottom:* Average performance of the algorithms relative to Reference at key points during learning and metatraining. Shown are the values measured across the runs indicated on the horizontal axis for *left:* just the first 20 trials and *right:* (same legend) the last 21 trials in each run. Error bars show problem-to-problem standard deviation 1σ of solution quality compared to Reference.

trajectories that are on average no better than Reference and are frequently far worse.

This problem is alleviated by **Autodetect**: in early runs, before its metapolicy has been well-trained, the quality measure often prevents the metapolicy from being used. As can be seen from the π_μ quality graph in Figure 4, the quality measure is on average above 0 (the threshold for use of the metapolicy's output), but in individual experiments it drops below 0 at the appropriate times and thus disables the use of the metapolicy. The Cost vs. μ–experience and bar graphs show the consequence: Autodetect's performance tracks that of PGRL early in training, and as the metalearner's experience with different problems in the domain grows, it surpasses PGRL and performs as well as PGRL+μ.

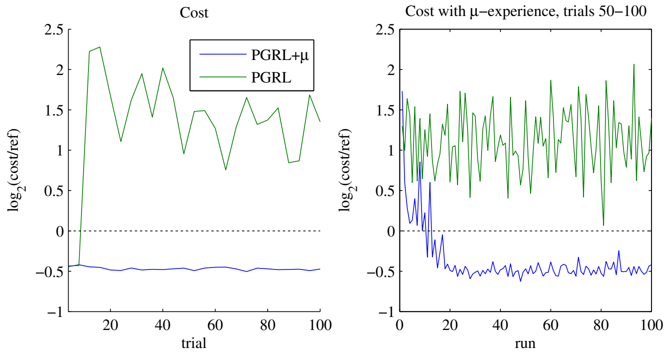


Fig. 5. Compensating for poorly chosen learning rates: PGRL vs. Autodetect with $\alpha_w = 0.97^{\text{trial}}$ and energy policy learning rate $\alpha_e = 1$ — a value only twice one that performs well. (10-experiment average)

The bar graphs in Figure 4 break down performance over the two learning timescales in order to show how quickly learning progresses without (runs 1–10) and with (runs 80–100) a well-trained metalearner. Performance over trials 1–20 shows initial learning speed, while performance over trials 80–100 shows what may be expected as the network matures. This further emphasises Autodetect's ability to nearly match the performance of the better of PGRL or PGRL+μ.

Another interesting feature of the Autodetect learner, visible in Figure 4's π_μ quality graph, is that the measured metapolicy quality progresses differently from that of PGRL+μ: the quality estimate stays higher initially, but begins to track that of PGRL+μ around run 40. The two metapolicies' parameters evolve differently due to the differences in training: while PGRL+μ always sees meta-actions from both the metapolicy and PGRL, Autodetect sees only the latter until it has proven itself, thus obtaining fewer training examples drawn from a different distribution. Raising Autodetect's threshold makes it less likely that the metapolicy gives bad advice to the learner, but reduces μ–learning speed. The effect that this has on our metapolicy quality estimate through time is intriguing, but we leave investigation as future work.

Much of the training time shown in Figure 4 may be required only once in a "lifetime" due to the transferability of the trained metapolicies. Once their metapolicies are trained, the metalearners facilitate the discovery of a good policy extremely rapidly—*after only a few trials*. They aggressively push policy changes that they have found in the past result in higher performance: quickly reducing energy use until nearing the UA's range limit and then backing off without requiring further exploration of the cliff's high-cost trajectories.

B. Sensitivity to learning rates

The base PGRL learner can achieve stable results if the learning step sizes such as α in (6) are chosen carefully. If learning rates are small, learning is slow, but a larger values exacerbate the learner's tendency to cliff-jump. Therefore it is typical in reinforcement learning to have α decrease over the course of a run—for example $\alpha = \alpha_0 x^e, x < 1$ for the update after epoch e. Ideally $\alpha \to 0$ as the policy nears the optimal.

For our task, this both provides a convergence guarantee for the hillclimber and reduces cliff-jumping, but it requires hand-tuning of x and α_0. Furthermore it eliminates the system's ability to adapt to a slowly changing environment; e.g., foliage growth or physically shifted sensors.

A poor choice of α_e interferes with the learners' ability to remain near the optimum. Here we have set $\alpha_e = 1$. We set the waypoint-placement learning rate $\alpha_w = 0.97^{\text{trial}}$ in order to allow both policies to converge more quickly, which is unnecessary in our other tests but helpful here. Figure 5 shows results: while $\alpha_e = 0.3$ (as shown in §VI-A) produces reasonable results, $\alpha_e = 1$ results in the exploration of many high-cost trajectories. PGRL performs extremely poorly, but PGRL+μ learns to anticipate the large jumps that result from the poor choice of α_e, resulting in costs worse than with a carefully chosen α_e but still significantly better than PGRL. The ability of the metalearner to stabilise learners with less hand-tuning may be a great asset in real-world systems.

C. Knowledge transfer with metapolicy initialisation

While the learned energy and waypoint policies are highly problem-specific, the metapolicy is more broadly applicable. This is shown in §VI-A by the gains on new problems after training on previous ones. But can a metapolicy trained for our single-node scenario be used to accelerate policy learning for problems drawn from a broader domain? The question of metapolicy transferability motivates us to modify our problem space as follows:

- For each problem, we place 5 sensor nodes randomly on a 40×40 field. This yields a great variety of radio field shapes since the nodes interfere with each other.
- For each problem, the data requirement for each node is drawn randomly from $[10 \ldots 30]$.
- We have corrupted the aircraft's information about the nodes' locations with Gaussian noise with $\sigma = 3$.

Figure 6 shows training results: for "Transfer" the metapolicy is set to the final Autodetect value learned during the generation of Figures 2–4:

$$\Theta = \begin{bmatrix} 1.70 & -1.16 \\ -1.61 & 1.34 \end{bmatrix} \tag{22}$$

Figure 6 shows the results in the extended domain: the earlier metapolicy achieves better than a 60% reduction in sensor radio transmission energy over Reference, and again offering speed and stability improvements over non-meta PGRL. The ability of single-node metapolicies to generalise to a broader problem configuration space is encouraging, but it is not perfect. For example, if we allow node data requirements to be drawn randomly in $[1 \ldots 10]$ the metapolicy is not cautious enough in its recommendations for policy updates due to the short contact times. Over multiple runs the metapolicy does continue to adapt based on policy learning from the new problems, but due to space constraints we do not study multi-node metapolicy learning here.

We show progress to 200 trials for this scenario. Figure 6 shows that for Transfer this is overkill, but PGRL has not yet converged. Performance gains in this scenario, even with a repurposed metapolicy, are slightly greater than in our first example, with Transfer achieving a 62% savings and even Half-power effecting a 45% savings. This difference is primarily due to the sensor position misinformation: Reference does not understand the error and so its performance degrades sharply as the information is compromised, whereas the learners—even Half-power—modify their trajectories to work around the misinformation.

VII. Conclusion and future work

We have demonstrated the feasibility of a reinforcement learning approach for autonomous discovery of energy-saving behaviours for sensor networks. With a soft trajectory length limit of twice a handcoded reference and a single sensor, the UA tended to fly 75% further than under Reference while sensors reduced communication energy by 60%. When the UA's knowledge of sensor positions is degraded by even a modest amount, the advantage of the learning approaches increases rapidly.

The UA does not model the environment, but learns directly from experience, saving the costs of locating the sensors and building system models, and eliminating the effects of modelling errors. We use a trajectory encoding that is well-suited to the task of interfacing between learning algorithms and proprietary autopilots. Additionally, we concurrently optimise both power and waypoint-placement policies.

Our reinforcement metalearner uses experience with the process of learning data-ferrying policies in order to accelerate and stabilise a conventional PGRL system, and transfers acquired knowledge about the policy-optimisation process to a range of unseen problems. This furnishes a new mechanism for approaching the global optimum of an unseen data-ferrying scenario extremely quickly while sampling few high-cost trajectories.

Much work remains to be done. For example, the metalearner reduces the necessity of hand-tuning the system by compensating for poor choice of parameters. A poorly chosen exploration rate frequently produces trajectories that greatly exceed the aircraft's range limit. Preliminary results show that as the metalearner gains experience with the base optimiser it learns to compensate, modulating the problematic policy updates and keeping trajectory costs lower, but a full investigation into the range and limits of this capability, both in the data-ferrying domain and for general model-free optimisation, remains to be done. Also, we have examined only learning in the vicinity of a few nodes. While preliminary results on the transfer of this metaknowledge to multi-node problems is promising, efficiently scaling the metalearning mechanism to sensor networks of many nodes is important future work. Finally, a more detailed characterisation of when and to what extent the metaknowledge is transferable will be key to understanding the broader applicability of the technique.

References

[1] A. Jenkins, D. Henkel, and T. Brown, "Sensor data collection through gateways in a highly mobile mesh network," in *Proc. IEEE Wireless*

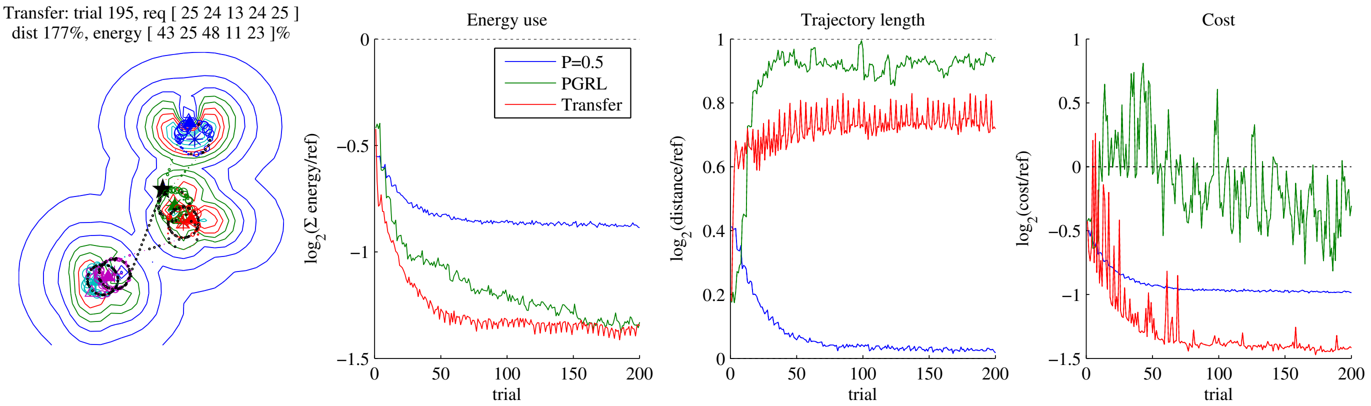


Fig. 6. Transferring the metapolicy to a larger field with varying data requirements and incorrect node position information. See Figures 2–4 for descriptions of the graphs. *Left:* the nodes are at ▲; their assumed positions, to which the waypoints were initialised, are at △; and "energy" is relative to Reference.

Communications and Networking Conference (WCNC). Hong Kong: IEEE, 2007, pp. 2784–2789.

[2] B. Pearre and T. X. Brown, "Fast, scalable, model-free trajectory optimization for wireless data ferries," in *IEEE International Conference on Computer Communications and Networks (ICCCN)*, 2011, pp. 370–377.

[3] H. Jun, W. Zhao, M. H. Ammar, E. W. Zegura, and C. Lee, "Trading latency for energy in densely deployed wireless ad hoc networks using message ferrying," *Ad Hoc Netw.*, vol. 5, pp. 444–461, May 2007.

[4] O. Tekdas, J. Lim, A. Terzis, and V. Isler, "Using mobile robots to harvest data from sensor fields," *IEEE Wireless Communications special Issue on Wireless Communications in Networked Robotics*, vol. 16, pp. 22–28, 2008.

[5] G. Anastasi, M. Conti, and M. Di Francesco, "Reliable and energy-efficient data collection in sparse sensor networks with mobile elements," *Perform. Eval.*, vol. 66, pp. 791–810, December 2009.

[6] M. Ma and Y. Yang, "Sencar: An energy-efficient data gathering mechanism for large-scale multihop sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, pp. 1476–1488, 2007.

[7] R. Sugihara and R. K. Gupta, "Optimizing energy-latency trade-off in sensor networks with controlled mobility," in *IEEE INFOCOM Mini-conference*, 2009, pp. 2566–2570.

[8] D. Ciullo, G. Celik, and E. Modiano, "Minimizing transmission energy in sensor networks via trajectory control," in *IEEE Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2010, pp. 132–141.

[9] L. Bölöni and D. Turgut, "Should i send now or send later? a decision-theoretic approach to transmission scheduling in sensor networks with mobile sinks," *Wireless Communications and Mobile Computing*, vol. 8, no. 3, pp. 385–403, 2008.

[10] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: a survey," *Ad Hoc Netw.*, vol. 7, pp. 537–568, May 2009.

[11] Y. Gu, D. Bozdag, R. W. Brewer, and E. Ekici, "Data harvesting with mobile elements in wireless sensor networks," *Computer Networks 50*, vol. 17, pp. 3449–3465, 2006.

[12] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling with dynamic deadlines," *IEEE Transactions on Mobile Computing*, vol. 6, no. 4, pp. 395–410, 2007.

[13] D. Henkel and T. X. Brown, "Towards autonomous data ferry route design through reinforcement learning," in *Proceedings of the 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*. Washington, DC, USA: IEEE, 2008, pp. 1–6.

[14] T. He, K. won Lee, and A. Swami, "Flying in the dark: Controlling autonomous data ferries with partial observations," in *Proceedings of the eleventh ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*. New York, NY, USA: ACM, 2010, pp. 141–150.

[15] W. Zhao and M. H. Ammar, "Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks," in *Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems*, ser. FTDCS '03. Washington, DC, USA: IEEE, 2003, pp. 308–314.

[16] M. Dunbabin, P. Corke, I. Vasilescu, and D. Rus, "Data muling over underwater wireless sensor networks using an autonomous underwater vehicle," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2091–2098.

[17] M. M. Bin Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in *MobiHoc: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2006, pp. 37–48.

[18] R. Sugihara and R. K. Gupta, "Improving the data delivery latency in sensor networks with controlled mobility," in *Proc. 4th IEEE international conference on Distributed Computing in Sensor Systems*, ser. DCOSS. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 386–399.

[19] ——, "Path planning of data mules in sensor networks," in *ACM Trans. Sen. Netw.*, vol. 8, no. 1. New York, USA: ACM, Aug. 2011, pp. 1–27.

[20] D. Henkel and T. X. Brown, "On controlled node mobility in delay-tolerant networks of unmanned aerial vehicles," in *International Symposium on Advance Radio Technolgoies*, 2008, pp. 7–16.

[21] A. Carfang, E. W. Frew, and T. X. Brown, "Improved delay-tolerant communication by considering radio propagation in planning data ferry navigation," in *Proc. AIAA Guidance, Navigation, and Control*. Toronto, Canada: AIAA, August 2010, pp. 5322–5335.

[22] M. Stachura, A. Carfang, and E. W. Frew, "Active sensing by unmanned aircraft systems in realistic communication environments," in *IFAC Workshop on Networked Robotics*, 2009, pp. 62–67.

[23] F. Jiang and A. L. Swindlehurst, "Optimization of uav heading for the ground-to-air uplink," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 5, pp. 993–1005, 2012.

[24] C. E. Shannon, "Communication in the presence of noise," in *Proc. Institute of Radio Engineers*, vol. 37, no. 1, 1949, pp. 10–21.

[25] N. Wagle and E. W. Frew, "A particle filter approach to wifi target localization," in *AIAA Guidance, Navigation, and Control Conference*. Toronto, Canada: AIAA, August 2010, pp. 2287–2298.

[26] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Networks*, vol. 21, no. 4, pp. 682–697, 2008.

[27] P. Glynn, "Likelihood ratio gradient estimation: An overview," in *Proceedings of the 1987 Winter Simulation Conference*, 1987, pp. 366–375.

[28] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 1992.

[29] J. Z. Kolter, Z. Jackowski, and R. Tedrake, "Design, analysis and learning control of a fully actuated micro wind turbine," in *Proceedings of the 2012 American Control Conference (ACC)*, 2012.

[30] J. A. Hertz, A. S. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Perseus Books, 1991.

[31] J. Baxter and P. L. Bartlett, "Infinite-horizon policy-gradient estimation," *Journal of Artificial Intelligence Research*, vol. 15, pp. 319–350, 2001.

Smart Parking System using Wireless Sensor Networks

Joseph Jeffrey, Roshan Gajanan Patil, Skanda Kumar Kaipu Narahari, Yogish Didagi, Jyotsna Bapat, Debabrata Das
International Institute of Information Technology, Bangalore

***Abstract*—Reliable communication is imperative for realising a vision of "Ambient Intelligence" where different devices gather and process information from different sources to exert control over the physical environment. Wireless Sensor Networks (WSN) have been employed to fulfil the necessity of reliable energy efficient communication between these devices. These networks consist of individual motes that are able to sense and control the environment with the help of sensors and actuators. In this paper, we describe the implementation of an energy and cost efficient smart parking system for multi-floor parking facility using WSN. The system monitors the availability of free parking slots and guides the vehicle to the nearest free slot. Additional information such as the amount of time the vehicle has been parked is also monitored for billing purposes along with the status of each mote. Cost is minimized by keeping the number of sensors lower without sacrificing the reliability. Energy consumption of each mote is kept in check by allowing the systems to sleep periodically and by reducing their communication range.**

***Index Terms*—smart environment; energy efficiency; wireless sensor networks; sensor mote; light-dependent sensors.**

I. Introduction

In recent years, there has been tremendous amount of research in the field of wireless sensor networks. These networks are typically ad-hoc in nature consisting of large number of sensing motes communicating with a Central Supervisory Station (CSS). The sensor mote is a battery operated device with limited computation and communication capabilities. The mote can be interfaced with several types of sensors to measure different environmental parameters. The wireless sensor networks have varied advantages like flexibility, inherent intelligence, low cost, rapid deployment and more sensing point, especially in area that cannot be wired. It is because of these advantages that they found their way in diverse application domains such as facility management, health care, environment monitoring, intelligent buildings, disaster relief applications etc. In this work, we have developed a smart parking management system that can track available parking slots economically and reliably and in turn contribute considerably to fuel and time conservation. An emerging trend in wireless sensor networks is its use in parking facility management. Typical car parking management systems monitor the number of cars passing the entry and exit points for estimating the free slots available. This result is then displayed at strategic locations for assisting the user.

A number of WSN systems have been developed to address the car facility management. In [1], the system is developed using the DSYS25z [2], mote with magnetic sensors. The system described in [1] concentrates on issues such as connectivity, sensing and network performance. In [3], [4] and [5], a WSN based car parking system are proposed, where each of these papers explore the possibility of using different kinds of sensors. They also propose different routing mechanisms for transferring the data from the source to the sink. Most of the existing systems have discussed a scheme in which the data collected from the multiple sensor motes is analysed by a central station and is displayed at strategic points to assist the user.

In this paper, we propose and implement a car parking management system using wireless sensor networks such that overall efficiency and flexibility of the facility management system is improved. The system is highly cost-effective as each mote is equipped with only one passive ambient light sensor, to detect the presence or absence of a car. Apart from detecting the car the sensor mote also provides additional information like the amount of time the car has been parked and also its health status. The system designed is also energy efficient, since the radio module at the mote is allowed to "sleep" at regular intervals. In addition the power consumed by the radio module is reduced by the use of repeaters. The proposed system is completely automated and does not require the presence of a human at the entry or exit point. The system not only displays the availability status at strategic locations but also sends the information such as slot allotted, time parked, billing information and directional details to the user's mobile phone via SMS (Short Message Service). By introducing the SMS feature we are basically targeting everyone as the number of mobile subscribers in the world are very high, which is expected to increase further tremendously. Furthermore, including the SMS feature helps us avoid the usage of paper or plastic cards that are currently used for the purpose of parking/billing.

This paper is organized as follows: Section II decribes the system architecture and its operation. Section III discusses about the implementation details of the system. Section IV describes the experimental setup. Sections V and VI present the future work and conclusion.

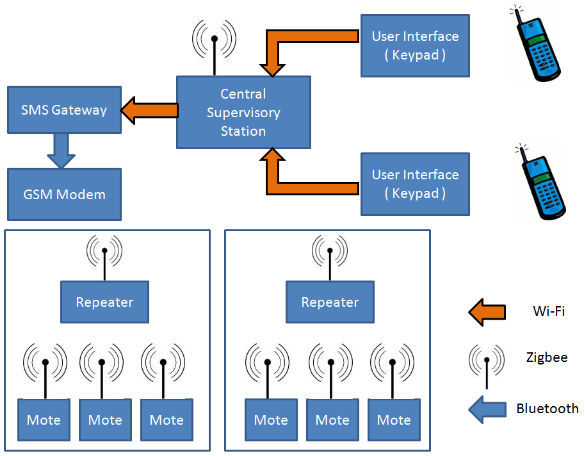


Figure 1. System Architecture

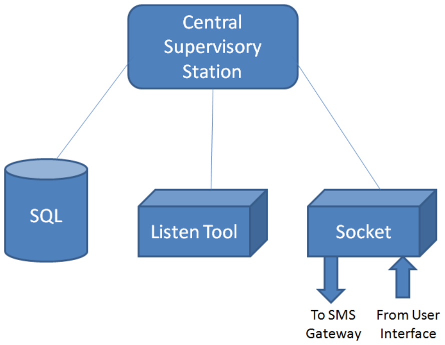


Figure 2. Central Supervisory Station Components

II. System Architecture

The block diagram of the system is shown in Figure 1 and the CSS comprises of components as shown in Figure 2. The operation of the system is as follows:

- A user enters the parking facility. At the entrance, there will be a keypad and a display as shown is Figure 9. The driver enters his mobile phone number using the keypad.
- On successful entry of the phone number, ID of the nearest empty parking slot, time of entry and route direction information will be displayed on the monitor. The same information will also be sent to the user's mobile phone via SMS.
- In order to incorporate the SMS feature, a GSM modem is connected to the CSS. A java based SMS gateway at the CSS provides the essential functionality using AT commands to the send the SMS.
- When the driver parks the car in the designated slot, a timer is started in the mote present in that slot. The mote will inform the CSS that the slot has been currently occupied.
- The CSS will update the database of the motes with the occupancy information along with the corresponding user mobile phone number. This is done to uniquely identify the parking slot with the vehicle.
- When the driver leaves the parking slot area, immediately the timer in that slot's mote stops. The timer data is communicated to the CSS.
- The CSS consults its database, extracts the mobile phone number field corresponding to the mote ID it received, and sends an SMS to the user. The SMS will contain information of how long the vehicle was parked and the

billing amount.

- By the time the driver reaches the exit of the parking facility, he/she would have received the SMS containing the billing information.

Each parking slot will be equipped with an MIB510 Iris sensor [6] mote running TinyOS [7] operating system and MDA100 ambient light sensor to detect the presence/absence of a vehicle. The mote keeps track of how long the vehicle has been parked and sends this timing information to a repeater, which in turn sends it to the CSS. The IEEE 802.15.4 (Zigbee Protocol) [8] is the protocol used for communication between the motes. The CSS calculates the billing information based on the data received from the sensor motes. The mote also sends the availability status to the CSS which is displayed on a suitable GUI as shown in Figure 10. The CSS monitors the status of mote health.

To overcome the energy problems encountered in multi-hop routing strategy, in the proposed system, the motes forward their data to a repeater mote placed at a convenient distance from both the CSS as well as the individual motes in the parking area. The repeater simply retransmits the data it receives from the motes to the CSS. The repeater is powered from the AC mains.

III. Implementation Details

The entire working flow of the system at CSS and the sensor mote are shown in Figure 5 and Figure 6, respectively.

A. *Parking bay occupancy detection and Data packet creation*

The method employs measurement of the amount of ambient light captured by the light sensor to detect the presence/absence of the vehicle over the parking slot. The sensor mote is placed at the center of the parking slot few inches below the ground level with a suitable physical casing to ensure only light incident in normal direction is used for evaluation purpose. The principle behind detection is that, when a vehicle occupies a slot, there is a significant dip in the light intensity. By comparing this light intensity with a suitable threshold a decision on the occupancy status is taken. The threshold set is adaptive in nature based on the present lighting conditions.

Each sensor mote creates a data packet of 14 bytes with the following fields; mote id, timing information and slot availability status. Each field occupies two bytes. The packet header is as shown in Figure 3:

Misc Field
Mote ID (2 Bytes)
Slot Availability Status (2 Bytes)
Timing Information (2 Bytes)

Figure 3. Occupancy Status Packet Header

This packet generated by the mote is transmitted to the CSS. At the CSS, a java based packet sniffer program is used to detect the packets sent to the CSS. This java program has been modified to not only detect the packets but also parse the packet to obtain just the last 6 bytes of data from the packet. These last 6 bytes having the Mote ID, the slot status and the timing information is then stored at the CSS.

B. *Dead or Alive (DoA) signalling*

The mote periodically transmits DoA signal to the CSS, which enables the CSS to keep track of the health of the motes. The sensor mote sends the DoA signal typically every five seconds. The DoA signal is nothing but a data packet containing only mote id as the payload. The CSS maintains a table with all the mote IDs as its entries. Whenever a DoA signal is received, a count corresponding to that mote is incremented. Then, a decision as to whether a mote is dead/alive is taken after averaging over three consecutive readings from all the motes for experimental purpose. The number of consecutive readings can be increased to obtain a higher probablity of assurance, whether a mote is dead or alive, but at the cost of increase in decision taking time. The packet header here is of 10 bytes and is shown in Figure 4:

Misc Field
Mote ID (2 Bytes)

Figure 4. DOA Packet Header

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-207-3

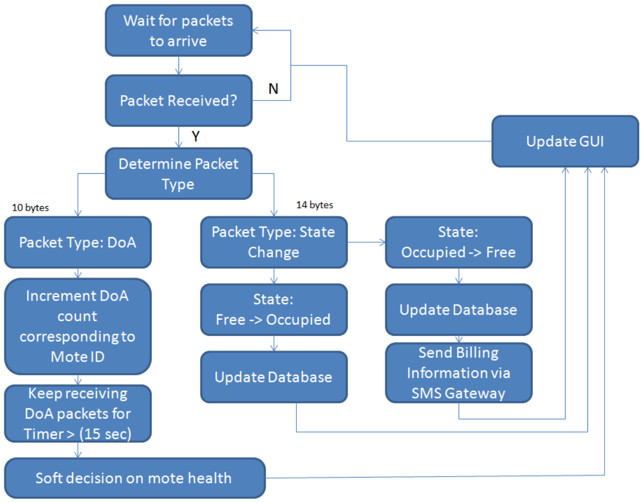

Wait for packets to arrive
Packet Received?
N
Y
Determine Packet Type
10 bytes
14 bytes
Packet Type: DoA
Increment DoA count corresponding to Mote ID
Keep receiving DoA packets for Timer > (15 sec)
Soft decision on mote health
Packet Type: State Change
State: Free -> Occupied
Update Database
State: Occupied -> Free
Update Database
Send Billing Information via SMS Gateway
Update GUI

Figure 5. Flow Diagram at CSS

C. Data Packet Transmission Optimization

Radio Transmission consumes a lot of battery power. In order to reduce the battery drain due to repeated transmissions of data packet, the data packet indicating the slot status is transmitted only when there is a state transition. The state transition happens when the vehicle arrives at or leaves the parking bay. Activating the radio module to communicate information only during these transitions reduces the power consumption to a great extent.

However, to further reduce the power consumption by the mote, it can be put to sleep state. Since the DoA signal is sent every 5 seconds by the motes, according to the calculation it takes approximately 40 microseconds to transmit the 10 bytes long DoA signal at speed of 250Kbps. After the DoA packet is sent, the mote's transceiver module is put to sleep state. The mote is then awakened after every 5th second or if there is a state transition.

It has been found that the radio module of the mote in active state draws 12mA. On the other hand, during the sleep state, it draws only 1µA. This shows that by putting the mote to sleep state considerable power savings can be achieved.

D. Adaptive Thresholding

The lighting conditions do not remain constant at all times. In order to adapt to the ever changing lighting conditions, an adaptive thresholding algorithm has been implemented. The present lighting conditions are monitored and readings are captured periodically for a fixed interval of time. These values are averaged out and after comparing it with the previous threshold, the new threshold is set. Hence the system is made intelligent to adapt itself automatically to varying light conditions.

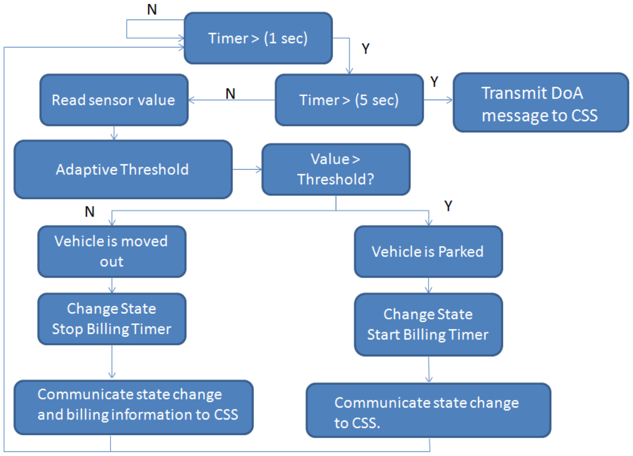

N
Timer > (1 sec)
Y
Read sensor value
N
Timer > (5 sec)
Y
Transmit DoA message to CSS
Adaptive Threshold
Value > Threshold?
N
Y
Vehicle is moved out
Change State Stop Billing Timer
Communicate state change and billing information to CSS
Vehicle is Parked
Change State Start Billing Timer
Communicate state change to CSS.

Figure 6. Flow Diagram at Sensor Mote

IV. Experimental setup

The project uses Crossbow IRIS – XM210 mote with MDA100CB sensor board. The MDA100CB sensor board is equipped with a CdSe photocell which acts as a light sensor. When there is light, the nominal circuit output is near VCC or full-scale, and when it is dark the nominal output is near GND or zero. This is used in the decision making process. Each mote runs on two AA batteries. The CSS mote is connected to the pc via USB cable for monitoring and display purposes. Figure 10 shows the display running at the CSS.

The adaptive threshold for the sensor mote was tested at an indoor parking facility at a mall. The communication range of the radio module was tested in the outdoor parking lot. The health status of the batteries in the motes is also monitored for maintenance purposes. According to therotical calculations, the battery life is estimated to be 1.58 years without considering the power consumed by the processor in idle state. The deployment of the sensor mote for the experiment is as shown in Figure 7 and Figure 8 shows a closer view of the same. In order to avoid the mote being over run by a car, the mote could be placed few inches below the ground surface during the actual deployment at the parking bay.

Attribute	IRIS - XM120
Program Flash Memory	128K bytes
Serial Flash	512K bytes
RAM	8K bytes
Serial Communication	UART
Analog to Digital Convertor	10 bit ADC, 8 Channel, 0-3V input
Current Drawn (Processor)	8mA (Active), 8uA (Sleep)
Frequency Band	2405 MHz – 2480 MHz
Transmit Data Rate	250 kbps
Outdoor Range	>300m
Indoor Range	>50m
Current Drawn (Radio module)	16mA (Rx mode), 10mA (Tx mode)
Battery	2 AA batteries

Table I
Hardware Specifications

Device	Software/Operating System
IRIS - XM120	TinyOS 2.1.1
Base Station - PC with Intel processor	Ubuntu 10.04
	Listen tool, GUI using JDK 1.6
	SMS gateway using JDK 1.6
	Socket Programming using JDK 1.6
	MySQL Database

Table II
Software Specifications

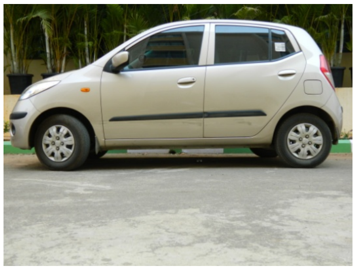


Figure 7. Deployment

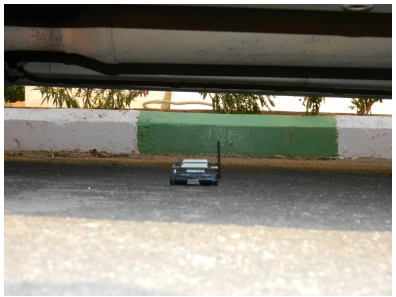
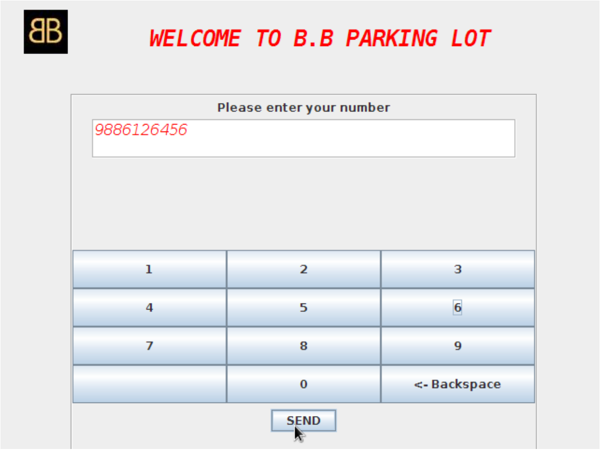


Figure 8. Deployment Under Car

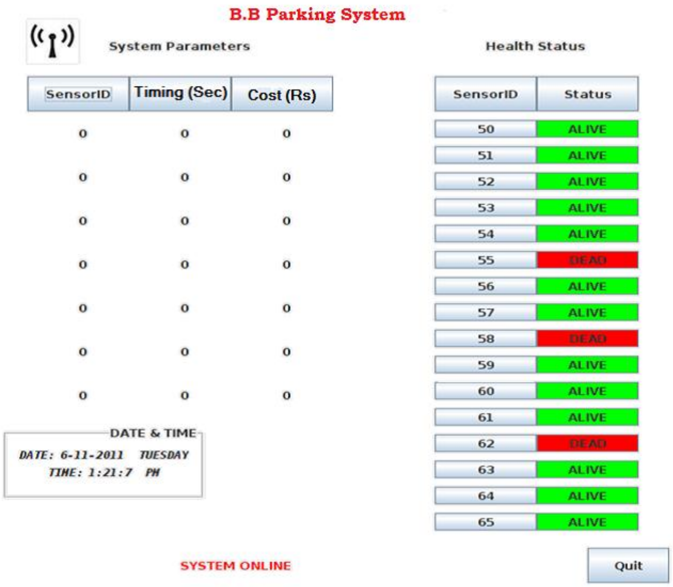


WELCOME TO B.B PARKING LOT

Please enter your number

9886126456

Figure 9. End User UI



B.B Parking System

System Parameters

SensorID	Timing (Sec)	Cost (Rs)
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Health Status

SensorID	Status
50	ALIVE
51	ALIVE
52	ALIVE
53	ALIVE
54	ALIVE
55	DEAD
56	ALIVE
57	ALIVE
58	DEAD
59	ALIVE
60	ALIVE
61	ALIVE
62	DEAD
63	ALIVE
64	ALIVE
65	ALIVE

DATE & TIME
DATE: 6-11-2011 TUESDAY
TIME: 1:21:7 PM

SYSTEM ONLINE

Figure 10. Central Supervisory Station UI

V. Future Work

In the future work, we plan to use additional sensors which will allow usage of the system in outdoors environment, albeit at increased cost. An active sensor may be added to validate the light sensor information. Cost may be further reduced by using a low cost processor mote such as custom built mote using the 8051 microcontroller. The system can be further extended to include the IMS network by providing information on SIP enabled devices. Another possible extension would be to provide location based services via mobile apps.

VI. Conclusion

The project successfully demonstrated the possibility of using WSN for multi floor parking facility. The system developed is fully automated, highly energy efficient and cost effective as only one sensor is utilized. The utilization of only one light sensor with the adaptive threshold algorithm was reliable in detecting the presence/absence of the car. As the application does not demand any complex routing mechanisms, our system implements the broadcast technique to communicate to the CSS. This is both simple in implementation and power efficient as opposed to unicast transmission which has a higher power profile due to the additional overhead to the packets. Finally, the system also proposed a novel approach of associating the slot, timing, direction and billing information with the users mobile number via SMS.

References

[1] J. Benson, "Car-Park Management using Wireless Sensor Networks," tech. rep., University College Cork (UCC), Cork, Ireland, November 2006.

[2] Andre Barroso, Jonathan Benson, Tina Murphy, Utz Roedig, Cormac Sreenan, John Barton, Stephen Bellis, Brendan Flynn, and Kieran Delaney, "The DSYS25 Sensor Platform," tech. rep., University College, Cork, Ireland and National Microelectronic Research Center, Cork, Ireland, 2004.

[3] Vanessa W.S Tang, Yuan Zheng, and Jiannong Cao, "An Intelligent Car Park Management System based on Wireless Sensor Networks," tech. rep., Internet and Mobile Computing Lab, Department of Computing, The Hong Kong Polytechnic University, P.R.China, August 2006.

[4] Seong-eun Yoo, Poh Kit Chong, Taehong Kim, Jonggu Kang, and Daeyoung Kim, "PGS: Parking Guidance System based on Wireless Sensor Network," tech. rep., Information and Communications University Daejeon, Korea, May 2008.

[5] B. Yan Zhong, S. Li Min, Z. Hong Song, Y. Ting Xin, and L. Zheng Jun, "A Parking Management System Based on Wireless Sensor Network," tech. rep., Institute of Software, Graduate School of Chinese Academy of Sciences, Beijing, November 2006.

[6] Sangwon Lee, Dukhee Yoona, and Amitabha Ghosh, "Intelligent Parking Lot Application Using Wireless Sensor Networks," tech. rep., Autonomous Networks Research Group, Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, May 2008.

[7] Itziar Marin, Eduardo Arceredillo, Aitzol Zuloaga, and Jagoba Arias, "Wireless Sensor Networks: A Survey on Ultra-Low Power-Aware Design," tech. rep., World Academy of Science, Engineering and Technology, August 2005.

[8] ZigBee/IEEE 802.15.4 Summary, "Sinem Coleri Ergen," tech. rep., EECS Berkely, September 2004.

[9] David Gay and Philp Levis, *The nesC Language: A Holistic Approach to Network Embedded System.*

[10] David Gay and Philp Levis, *Software Design Patterns for TinyOS.*

[11] David Gay and Philp Levis, *nesC 1.1 Language Reference Manual.*

[12] Wikipedia The Free Encyclopedia, "Tiny OS Documentation Wiki." http://docs.tinyos.net/, August 2011. Accessed: 25/6/2012.

[13] "Tiny OS 2.0.2 Documentation." http://www.tinyos.net/tinyos-2.x/doc/, July 2007. Accessed: 25/6/2012.

[14] "Data Sheet for Memsic IRIS Motes." http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html. Accessed: 28/7/2012.

ISBN: 978-1-61208-207-3

Experiments for Fire Detection Using a Wireless Sensor Network

Ronald Beaubrun
Department of Computer Science and Software Engineering
Université Laval
Quebec, Qc, Canada
e-mail: ronald.beaubrun@ift.ulaval.ca

Yacine Kraimia
Department of Computer Science and Software Engineering
Université Laval
Quebec, Qc, Canada
e-mail: yacine.kraimia.1@ulaval.ca

***Abstract*— Recent works have shown the needs for using *Wireless Sensor Networks* (WSN) for rural and forest fire detection. In this context, a number of approaches have been proposed. However, such approaches do not take the wind direction into account, which is not realistic and does not provide accurate information on how the actual fire can be spreading out. This paper proposes a set of real-time experiments that take into account the wind direction in the context of fire detection using a WSN. Results analysis shows that the wind direction has an impact on the temperatures recorded by the sensors and can be an important parameter for on-time fire detection.**

Keywords-Experiment; fire detection; wind direction; wireless sensor network (WSN).

I. INTRODUCTION

Nowadays, the necessity for observing and controlling hostile environments through *Wireless Sensor Networks* (WSNs) becomes essential for many military and scientific applications [1]. Recent works have shown the needs for using such networks for rural and forest fire detection [2]-[8]. The basic principle for detecting fire using WSNs is to deploy a number of wireless sensors to cover a target area. Such sensors should be able to collect several parameters, such as temperature, humidity level and light intensity [2], [4]. In order to properly evaluate those parameters, three types of sensors are generally used [8]. The first type is responsible for collecting information about the environment where these sensors are deployed. Such information is transmitted via a radio link to the second type of sensors, *i.e.*, the sinks. This type of sensors processes the data coming from the wireless sensors and transmits such data to a gateway. The latter is directly connected to a host computer via a USB (*Universal Serial Bus*) port, which enables to directly transfer the data received from the sinks to the host computer. In this context, the host computer displays the processed information to home users.

Many applications of WSNs have been implemented for fire detection [1], [9], [10]. The role of these applications is to provide accurate information on how the actual fire is spreading out. In this context, a number of equipments, such as GPS (*Global Positioning System*), cameras and base stations, can also be used in order to have more details and to better locate the fire. For example, Lloret *et al*. [11] used cameras in a wireless sensor network for image and video caption of a real fire, which is useful to validate the presence of fire and prevent false alarms. Also, the sensor network implemented by Osman *et al*. [12] does not take the wind into account, which does not properly detect the presence of fire on time. In the same vein, Yu *et al*. [13] propose a method for detecting forest fire, using WSNs for collecting information on temperatures, relative humidity and wind speed. However, in the paradigm proposed in [13], the wind direction is not taken into account, which is not realistic and arises some questions. How can the presence of wind in a site have an impact on the temperatures recorded? Does the wind direction affect the process of fire detection?

This paper proposes a set of real-time experiments that enable to assess the impact of wind direction on the temperatures collected by a WSN. It is organized as follows. Section 2 presents the hardware platform, as well as the application implementation of the proposed approach. Section 3 describes the experiments and discusses the results analysis. Section 4 gives some concluding remarks.

II. THE PROPOSED APPROACH

In this research, we consider a realistic model which takes into account the wind direction in the context of fire detection using a WSN. In this section, we present the hardware platform and the implementation of the WSN applications.

A. Hardware platform

The proposed network consists of a sensor that collects information about its environment, and a sensor gateway that transfers the collected data to a host computer. In order to ensure interaction between the environment and the hardware system, a primary application for data collection is implemented, whereas a second application for graphical display is installed on the host computer.

ISBN: 978-1-61208-207-3

For the experiments, *Tmote Sky* wireless sensors from Moteiv were used [14]. Each sensor is equipped with integrated detectors for light, temperature and humidity detection [15]. The sensors are controlled by a Texas Instruments MSP 430 microcontroller which has 48 KB of flash, 10 KB of RAM, and contains an internal digitally controller directed oscillator (DCO) that runs at 8 Mhz. In the same vein, *TinyOS* was used as a development environment [16]. It is an operating system designed for networks with limited resources. TinyOS libraries and applications are written in *nesC*, a version of *C* that was designed for programming embedded systems. In nesC, programs are built from components that are connected together to form an entire program.

According to [15], the units for the temperatures obtained are ADC (ADC_{counts}). Conversion to °C is done as follows:

- Find the internal voltage V_{int} as follows :

$$V_{int} = (ADC_{counts}) * (V_{ref}/4096) \qquad (1)$$

where $V_{ref} = 1.5$

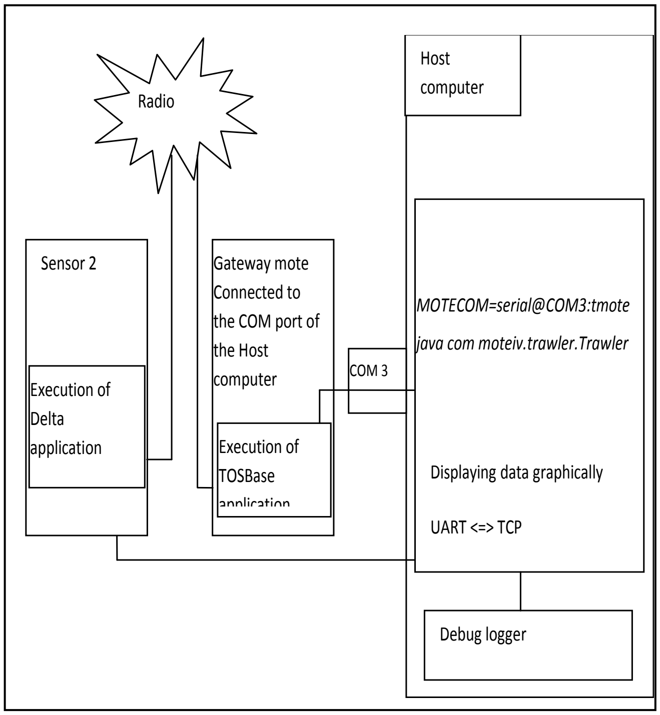
- From (1), it is possible to calculate the temperature in °C as follows:

$$Temperature\ (°C) = (V_{int} – 0.986)/0.00355 \qquad (2)$$

B. Implementation of the WSN applications

A WSN application is a distributed program which consists of several modules that are executed by different computers. Such modules are illustrated in the deployment diagram presented in Figure 1, where the nodes represent the software packages and the lines correspond to the data flowing between the nodes. The application execution is controlled by a host computer that is connected with the sensor gateway. Then, the communication between the sensor gateway and the host computer is established with a USB cable, using the application coming from the base station (*apps/TOSBase part*) of TinyOS distribution.

In the context of this research, the needs for implementing a number of applications are specific. First of all, it is necessary to be able to modify the parameters of the experiments: the fire intensity, the wind direction, as well as the number of sensors and their coordinates. Secondly, as the deployment is expensive, it is necessary to place the sensors in specific locations, so that it is possible to download all the samples recorded and to detect missing data. Furthermore, it is difficult to control the wind direction. In order to do so, a controlled wind will be produced by using a fan. Finally, an electric oven will be used in order to generate and simulate the heat.


Radio
Host computer
Sensor 2
Execution of Delta application
Gateway mote Connected to the COM port of the Host computer
Execution of TOSBase annlication
COM 3
Displaying data graphically
UART <=> TCP
Debug logger

MOTECOM=serial@COM3:tmote
java com moteiv.trawler.Trawler

Figure 1. Deployment diagram.

Moreover, the main application used in the experiments is called *Delta*. Its installation requires first to compile it in order to obtain a binary image in TinyOS, then to install the binary image on each sensor. The commands for compiling and installing Delta application are the following:

```
cd /opt/moteiv/apps/Delta
make tmote
make tmote reinstall,1
```

Once Delta application is installed, each sensor node can sample its internal temperature and communicate it to the other sensors. Then, another application called *Trawler* that comes with TinyOS can be used. Trawler enables to start the process of creating an ad hoc network and displaying the network topology, as well as the messages received by the PC on the screen. It is responsible for sending the sensor commands via a USB port and the gateway. Also, it is responsible for graphical representation of incoming data, and for displaying the current temperatures obtained from the sensor network and information about the sensor status. The command for running Trawler is:

```
MOTECOM = serial@COM4:tmote java com
moteiv.trawler.Trawler
```

This command enables to send requests to the sensor nodes through port COM4 that connects the gateway with the PC.

Moreover, the following application *java net.tinyos.sf.SerialForwarder* is used as a java applet that creates a TCP socket for enabling data sharing with other applications on the host computer. It forwards the requested packets to the *Universal Asynchronous Receiver Transmitter* (UART), as the sensor attached to the PC sends the messages on the radio link. In addition, it takes the messages coming from the UART, and forwards them to other users via an Internet connection, which enables other applications to use such information.

Note that, in the proposed experiments, only one sensor sends information to the sensor gateway using the radio link. Then, the sensor gateway sends such information to the host computer which displays it on the screen. In this context, the sensor gateway runs an application that listens to the radio link. When a message coming from the sensors is received, it is transmitted via a serial interface, in order to make it available to the *serial forwarder*, and finally to the screen.

III. RESULTS AND ANALYSIS

The first experiment aims at collecting the temperatures recorded and transmitted over time by the sensor. Such an experiment is mainly carried out for testing the sensor capability to collect exact temperature values. For that, an oven is used to produce the heat, as the sensor is placed at 6 cm from the heat and the wind speed is set to zero. At the beginning of the experiment, the sensor indicates 17,079° C as ambient temperature. Then, after turning on the oven, the sensor node indicates that the temperatures are increasing, as illustrated in Figure 2. We then realize that the temperature variation is linear from the 103rd to the 1203rd second, then becomes logarithmic. The maximum temperature is 54.215° C obtained after 2303 seconds.

Such results can be interpreted as follows. At the beginning of the fire, the temperatures increase quickly, which justifies the linearity between 103 and 1203 seconds. After a while, the growth rate of the temperatures decreases until it converges to a nearly constant value. This corresponds to the case of a real fire, where temperatures are used to continuously increase until a fixed value [13]. In other words, in the context of fire monitoring, a 50° C threshold can be set. Therefore, when the temperatures collected by the sensors exceed such a threshold, an alarm can indicate the possible presence of fire.

Moreover, in the context of a real wildfire, the wind can play an important role in the fire propagation. The purpose of the following experiments is to evaluate the impact of the wind direction on the temperatures recorded by the sensors. It helps answer the following question: will the temperatures collected by the sensors remain the same if the tests are repeated under the same conditions, but with changes in the wind direction? For this, a wind generated by a fan at a constant speed of 8 km per hour is established. The distance between the sensor and the heat is set to 6 cm. The wind direction is controlled by positioning the fan in specific angles relative to the axis that passes through the heat and the sensor as follows: 0 degree, 45 degrees, 90 degrees, 135 degrees and 180 degrees, which is illustrated in Figure 3. More specifically, angle of 0 degree means that the wind blows in the same direction as the heat and the sensor, *i.e.*, the wind first passes through the heat before it reaches the sensor. On the other hand, angle of 180 degrees means that the wind first passes through the sensor before it reaches the heat. Angles of 45 degrees, 90 degrees and 135 degrees are other directions that the fan can take with respect to the heat and the sensor. Note that for all the experiments, the temperature values are registered at the same time (2200th second) for each orientation. Also, note that we have not considered any gust factor in the experiments.

Figure 4 illustrates the temperature values obtained for this experiment. We realize that the highest temperature is registered when the wind direction makes an angle of 0 degree. Every time the wind direction is changed, the sensor collects lower temperature values. Specifically, at 135 degrees, the lowest temperature is registered. Indeed, when the wind blows at 135 degrees, it redirects the heat from the fire to other directions. As a result, it prevents the sensors from collecting the correct temperature values. Theoretically, the sensor should collect the lowest temperature at 180 degrees. However, for unexplained reasons, it was not possible to obtain such a temperature for this condition of the experiment.

In general, the distance between the sensor nodes and the heat can play an important role in the early detection of the fire. As a result, the sensor nodes must be as close as possible to the heat, so that they can report the presence of fire on time. In the next experiment, we propose to evaluate the impact of wind direction on the temperatures collected, while changing the distance between the sensor and the heat to 20 cm, and keeping the other conditions the same as in the previous experiment. Figure 5 illustrates the results obtained for this experiment. As for the previous experiment, the highest temperature is registered for wind direction of 0 degree, whereas the lowest temperature is registered for wind direction of 135 degrees.

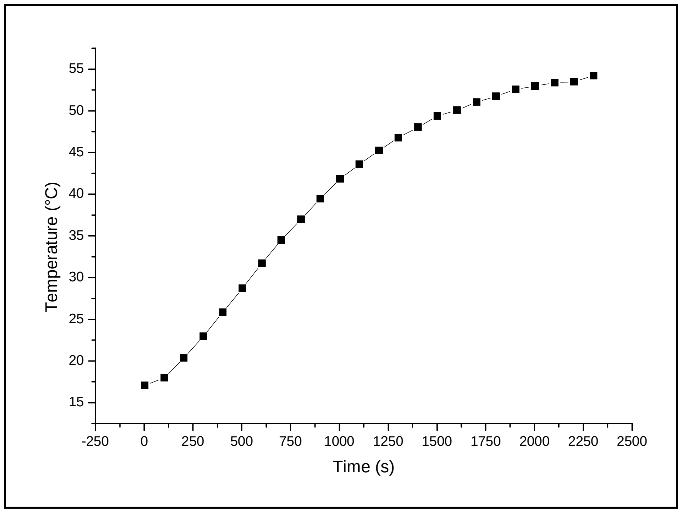


Figure 2. Temperature variation indicated by the sensor node.

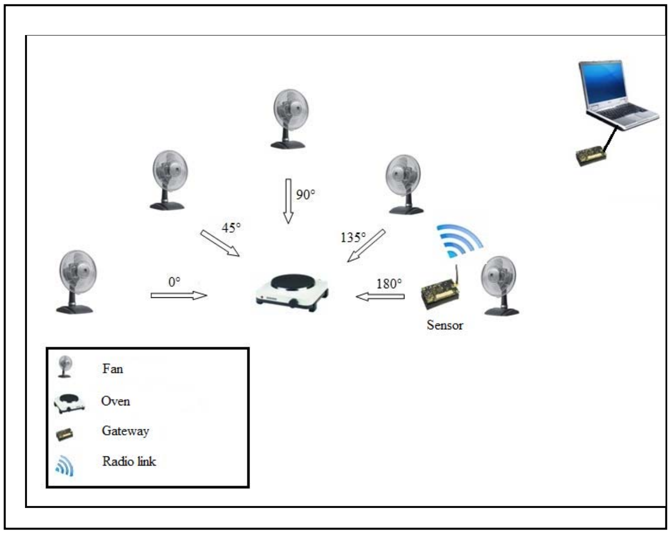


Figure 3. Variation of wind direction in the proposed approach.

Figure 6 compares the results obtained from both experiments. In general, for the same wind direction, the temperatures collected by the sensor at 20 cm from the heat are lower than those collected at 6 cm. For instance, the highest and lowest temperatures recorded by the sensor at 6 cm are 51.5 °C and 41.2 °C respectively, whereas such temperatures respectively reach 43.3 °C and 34.39 °C at 20 cm. Such comparison confirms that the maximum temperature is recorded at 0 degree, whereas the minimum temperature is recorded at 135 degrees. In other words, if a WSN is implemented for fire detection under windy conditions, the time for fire detection will be the fastest for wind direction of 0 degree and the slowest for wind direction of 135 degrees. For wind directions of 45 or 90 degrees, the collected temperatures depend on the distance between the sensor node and the heat. At 6 cm, the temperature registered for wind direction of 45 degrees is higher than that registered for wind direction of 90 degrees, which is not the case at 20 cm. As a result, for wind directions of 45 or 90 degrees, the time for fire detection depends on the distance between the heat and the WSN. An important lesson learned is that the wind direction has an impact on the temperatures collected by the sensors and can be an important parameter for detecting fire on time using a WSN. Note that at 180 degrees, the registered temperatures are not the lowest for both experiments.

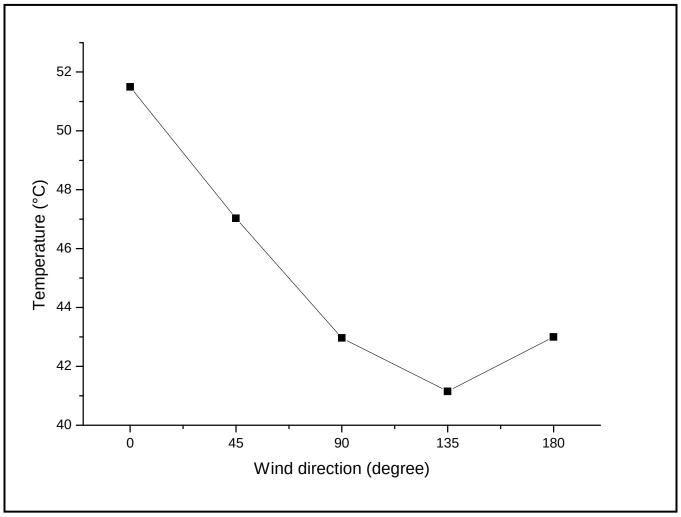


Figure 4. Temperature variation in function of the wind direction at 6 cm.

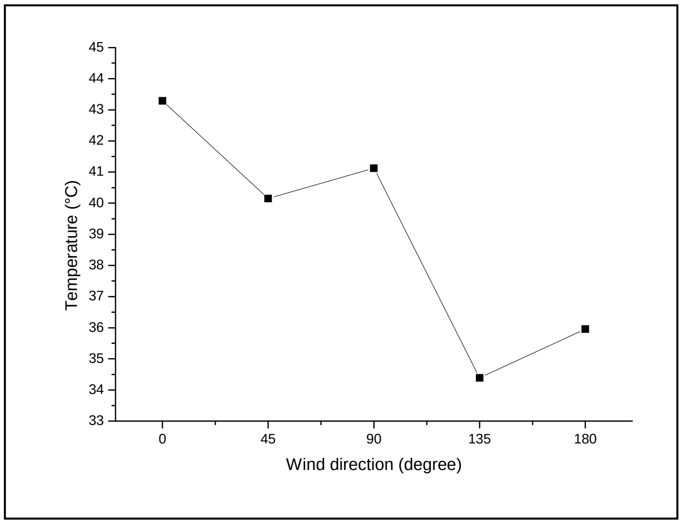


Figure 5. Temperature variation in function of the wind direction at 20 cm.

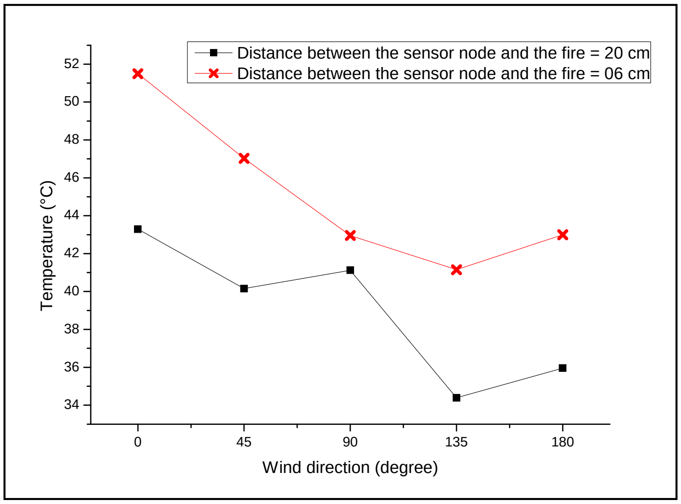


Figure 6. Comparison of temperatures recorded for two different distances.

IV. CONCLUSION AND FUTURE WORK

In this paper, a set of experiments were presented for assessing the impact of wind direction on the temperatures recorded by a WSN. In this context, a real heat was generated using an oven, whereas the presence of wind was simulated using a fan that had different orientations. Several applications were implemented in the sensor nodes for recording environmental temperatures and sending such information to a sensor gateway. Results analysis shows that the wind direction has an impact on the recorded temperatures. More specifically, if a WSN is implemented for fire detection under windy conditions, the time for fire detection can be the fastest for wind direction of 0 degree and the slowest for wind direction of 135 degrees. Future work should focus on methods and algorithms that consider the impact of both wind direction and wind speed on fire detection using a WSN.

REFERENCES

[1] D. Puccinelli and M. Haenggi. “Wireless Sensor Networks: Applications and Challenges of Ubiquitous Sensing,” IEEE Circuits and Systems Magazine, vol. 5, pp. 19-31, 2005.

[2] D. M. Doolin and N. Sitar, “Wireless Sensor for Wildfire Monitoring,” Smart Structures and Materials 2005: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems, SPIE, Volume 5765, pp. 477-484, San Diego, USA, 2005.

[3] J. Glasa and L. Halada, “Envelope Theory and its Application for a Forest Fire Front Evolution,” Journal of Applied Mathematics, Statistics and Informatics, No. 1, pp. 27-37, 2007.

[4] M. I. Asensio and L. Ferragut, “On a Wildland Fire Model with Radiation,” Int. Journal for Numerical Methods in Engineering, vol. 54, pp. 137-157, Feb. 2002.

[5] Z. Nauman, S. Iqbal, M. I. Khan, and M. Tahir, “WSN-Based Fire Detection and Escape System with Multi-modal Feedback,” MCSS 2011, CCIS, vol. 149, pp. 251–260, 2011.

[6] P. Roy, S. Bhattacharjee, S. Ghosh, S. Misra, and M. S. Obaidat, “Fire Monitoring in Coal Mines Using Wireless Sensor Networks,” Performance Evaluation of Computer & Telecommunication Systems (SPECTS), the Hague, Netherlands, Jun. 2011.

[7] E. S. Manolakos, E. Logaras, and F. Paschos, “Wireless Sensor Network Application for Fire Hazard Detection and Monitoring,” SENSAPPEAL, vol. 29, pp. 1–15, Athens, Greece, Sep. 2009.

[8] L. Shixing, X. Wujun, and Z. Yongming, “Research and Implementation of WSN in Fire Safety Applications,” IEEE 6th International Conference on Wireless Communications, Networking and Mobile Computing, Chengdu, China, Sep. 2010.

[9] R. Beckwith and P. Bowen, “Report from the Field : Results from an Agricultural Wireless Sensor Network”, IEEE 29th Conference on Local Computer Networks, pp. 471-478, Tampa, Florida, Nov. 2004.

[10] G. D. Richards, “An Elliptical Growth Model of Forest Fire Fronts and its Numerical Solution”, International Journal for Numerical Methods in Engineering, vol. 30, pp. 1163-1179, 1990.

[11] J. Lloret, M. Garcia, and S. Sendra, “A Wireless Sensor Network Deployment for Rural and Forest Fire Detection and Verification”, Special Issue State-of-the-Art Sensors Technology in Spain, vol. 9, pp. 8722-8747, Valencia, Spain, 2009.

[12] S. Osman, J. E. Penha, and F. Nakamura, “Fusing Light and Temperature Data for Fire Detection,” IEEE Symposium on Computers and Communications, Riccione, Italy, pp. 107-112, Jun. 2010.

[13] L. Yu, N. Wang, and X. Meng, “Real-time Forest Fire Detection with Wireless Sensor Networks,” International Conference on Wireless Communications, Networking and Mobile Computing (WIMOB), vol. 2, pp. 1214–1217, 2005.

[14] S. Smolau, “Evaluation of the Received Signal Strength Indicator for Node localization in Wireless Sensor Networks”, M.Sc. thesis, Laval University, Quebec, 2008.

[15] Moteiv. “Tmote Sky Ultra Low Power IEEE 802.15.4 Compliant Wireless Sensor Module Datasheet,” 2006.

[16] TinyOS: An open-source OS for the wirless embedded sensor networks. http://www.tinyos.net. [Retrieve: July, 2012].

Linking Sensor Data to a Cloud-based Storage Server Using a Smartphone Bridge

David D. Rowlands[a], Jason R. Ride[a], Mitchell W. McCarthy[a], Liisa Laakso[b], Daniel A. James[a]
[a]Centre for Wireless Monitoring and Applications, Nathan campus
[b]Applied Physiotherapy and Exercise Science, Gold Coast campus
Griffith University
Brisbane, Queensland, Australia
d.rowlands@griffith.edu.au, j.ride@griffith.edu.au, mitchell.mccarthy@griffithuni.edu.au,
l.laakso@griffith.edu.au, d.james@griffith.edu.au

Abstract – **There is a need for systems that unify the processes of collecting, storing and analysing long-term sensor data in multi-user contexts. The ubiquity and communication features of smartphones make them suitable as data aggregation platforms for standalone sensors. Cloud-based servers are advantageous for their centralisation and scalability in the storage and processing of data. This paper conceptualises a method to link sensors with a cloud server by detailing a smartphone application design (for collecting, aggregating and transmitting data from connected sensors), communication protocol (to preserve security and integrity of data during transmission) and storage methodology (to protect data during processing and storage within the cloud server). Overall, this paper presents design concepts to link sensors with cloud-based storage and provides details of a health-based implementation that applies them.**

Keywords – sensor; smartphone; security; cloud; data.

I. Introduction

The development of ultra-low power transmission protocols for wireless sensor devices has led to an increase in the number of physiological measurement systems that typify Body Area Networks (BAN) [1][2]. These networks typically collect physiological information from multiple sensors and transmit it to a central receiver for storage, analysis and/or feedback. The process of linking many sensors to a central server can be performed by an aggregation platform tasked with consolidating and securely re-transmitting all data collected from a single body. Current generation smartphones have shown to be suitable for this purpose, in both short-term single-user feedback [3] and longer-term multi-user feedback [4][5] contexts. However, many of the existing multi-user systems tend to focus on direct distribution of data (without analysis) [4] or are tailored to specific targets and lack versatility [5]. There is a need for a more generalised system that integrates collection, aggregation, secure storage and analysis with the potential to incorporate future data formats and processing methods by expanding on rather than replacing the fundamental system design.

This paper outlines the proposed design of a system that uses a smartphone as an aggregation platform to link standalone sensors with a cloud-based server. The cloud server builds on previous work by the authors: a server platform designed to closely integrate storage and analysis of data within a scalable distributed architecture [6]. Section II outlines important design factors in using a smartphone as a sensor bridge. Section III details a practicable strategy for securely and reliably transmitting data to the cloud server, and the methodology employed within the server to protect data during processing and storage. Section IV describes the preliminary development track for a health-based implementation based around the proposed techniques.

II. Smartphone Bridge

Traditional methods of collecting data from external sensors such as pedometers that relied on manual user input have been shown to require intrusive “compliance-enhancing features” to sustain a high level of data entry and integrity [7]. As a promising alternative, smartphones can facilitate data collection from sensors by acting as an automated aggregation, storage and communication platform. This allows them to operate as a standalone aggregation platform that provides basic analysis and feedback of data to the user, a bridging device that transfers the data directly to an external system, or a combination of the two. The concept of a bridging device is particularly useful when large amounts of data are being collected, either from one or many users, or there is computationally heavy analysis to be performed that would benefit from the use of dedicated tools.

A. Application Design

When designing a persistent application such as a sensor bridge, the developer must consider that smartphones are capable of performing many concurrent tasks that may frequently and randomly require control of the user interface. Fortunately, smartphone operating systems (OSs) support background states where an application may operate almost invisible to the user without disrupting other applications. This is an ideal method for implementing a sensor bridge, as the majority of its lifecycle is spent collecting data from external devices. In using this method, consideration of user notification, power consumption, and server connectivity factors is essential.

B. User Interaction

While it was mentioned that user reliance in data collection resulted in poor compliance, automated methods may occasionally require some administration in order to operate correctly. A sensor collection application, for instance, will require a means of modifying variable settings, such as a list of sensors, remote server address or user credentials. There are a variety of methods to achieve such tasks, such as a downloaded configuration file or a simple interface that appears when either the application is installed.

C. *User Notification*

In the absence of a graphical interface, notification tools such as icons and pop-up messages provide a mechanism of either reassuring the user that the service is operating as intended or to alert them to issues that require their attention. In the context of a sensor bridge, regular events such as sensor status updates and successful data transfers to a remote system do not require user involvement, merely notifying that the system is operating correctly. Non-intrusive, non-blocking methods (e.g., Android's Toast notifications) that simply appear on the screen before fading after a short time are suitable for these events.

Critical events such as sensor or communications hardware failure that may result in data loss demand immediate attention. Therefore, intrusive methods incorporating features such as vibration, audio tones and interface-blocking visual alerts requiring acknowledgement provide the best chance of notifying the user.

D. *Power Management*

Background applications typically do not require a user interface; therefore power consumption caused by screen lighting is largely insignificant. However, the constant use of communications hardware for a sensor bridge will have a noticeable impact on battery life. To alleviate this, methods are required that help to minimize power use where possible.

It is well established that location tracking using Global Positioning Systems (GPS) can consume considerable power [8] and have a high Time To First Fix (TTFF), making it impractical in applications where location data is requested infrequently. Fortunately, OSs allow location tracking to be achieved without the use of GPS, instead using the known coordinates of cell towers and wireless access points (WiFi) to triangulate the user's position. This can be achieved in a fraction of the time, with a reduced accuracy depending on the availability of network infrastructure and with a significant reduction in power consumption.

Smartphones typically have an idle state of activity where the internal processor enters a low power mode until a wake up event occurs. However, the use of a background application for constant sensor communication will prevent the OS from entering this state, regardless of phone or user activity. By monitoring the incoming data, it may be possible to detect periods of user inactivity where collection can be temporarily suspended (e.g., during sleep).

E. *Server Connectivity*

A simple implementation of a bridging application may involve a persistent connection with an external system with collected data streamed immediately across. However, long-term collection from multiple sensor devices can generate significant amounts of data that may consume the limited mobile phone data allowance available to the user. Use of WiFi can aid this situation by only uploading when a network connection is available and reserving the cellular network for emergency use. Applying a method such as this will involve the use of temporary storage of data on the mobile phone, so storage capacity must be considered when deciding how long data may be allowed to accumulate before utilizing the cellular network.

In order to ensure that data transfer to an external system maintains a high level of security and integrity, a robust communications protocol must be devised. The following section proposes a protocol that meets these requirements.

III. Centralised Server

The data storage methodology used in this system employs and extends an existing system that integrates the collection, storage, analysis and visualisation of multi-channel multi-sensor data within a cloud context [6]. The key elements explored in this section include securely communicating sensor data from multiple mobile clients to a central server, as well as handling reception, manipulation, storage and protection of data within the server.

A. *Authentication and Communication*

In this system, each connection from client to server represents a link between an identifiable user (with a unique username and secret password) and an arbiter within the server that controls access to the storage and processing tools. The functionality provided by the server is intended for use with private data; this means that all connections must be authenticated so that eavesdropping, modifying or otherwise intercepting communication is not foreseeably possible. Protocols such as HTTPS [9] and WEP/WPA2 (WiFi security) provide some level of protection, but neither have been proven to be infallible [10][11] and are conceptually vulnerable to man-in-the-middle style attacks.

The authentication procedure used in this system aims to prevent man-in-the-middle attacks without the need for a trusted certification authority by using a private key that is never communicated over a network. Figure 1 shows a proposed methodology that implements this authentication procedure, consisting of three distinct parts – login (A), data transfer/processing (B) and logout (C).

The authentication process begins when the client software receives a unique username and confidential password from the user. The two credentials are encrypted using a hashing function to generate a secret cipher key that is stored internally for later use. When a session is required, the client requests the right to be authenticated (Figure 1 A1), which is reciprocated with a pseudo-random unique

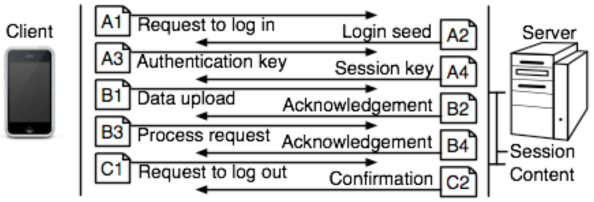


Figure 1. Communication protocol for uploading data

login seed (A2) from the server. The client then uses a second hashing function to produce an authentication key from the secret cipher key and the received login seed. This authentication key (A3) is sent to the server for comparison with an equivalent (generated internally using the same two hashing steps). If the two match, the server replies with a private session key (encrypted using the secret cipher key) (A4) for use on all messages within the session (B1-B4).

Since the authentication key (A3) is dependent on the login seed that preceded it, it cannot be re-used to authenticate illegitimately. By limiting the frequency of login requests (A1), the risk of dictionary-based attacks can be minimised. The length of the session key (A4) controls the relative probability of brute force decryption [11], and is based upon the maximum number of packets expected within any given session. Because the session key is private, symmetric-key encryption techniques such as Integrity Aware Cipher Block Chaining (IACBC) can be used; this particular technique combines confidentiality and verification of correctness into a single computationally efficient algorithm [12]. Replay attacks can be prevented within the server.

Messages sent within the context of an authenticated session may include, but are not limited to, data uploads (B1) and requests to run data processing scripts (B3), both of which are reciprocated with an acknowledgement of success (B2, B4). In data uploads, specific details such as packet size, format, timing and error handling are defined at the discretion of each client, but each upload produces a single data file on the server. Data processing involves collating these packets, optionally synchronising and filtering the content and storing the result. Such requests may occur until the client logs out (C1, C2) or the session is terminated.

This protocol will be further analysed in a future validation study, with the null hypothesis being that socially engineered attacks and weaknesses in the smartphone OS represent the greatest threat to the security of the system. The performance implications have not yet been analysed.

B. *Data Reception*

Figure 2 details the internal modules within the server relating to reception, processing and storage in the context of uploading data from external clients. Requests from clients are managed by the Input/Output (I/O) handler, which communicates with the File Handler, Script Handler and Session Handler as necessary, depending on the type of request received. The Session Handler also communicates authentication details to the Rule Parser, which protects the Storage Database from unauthorised access.

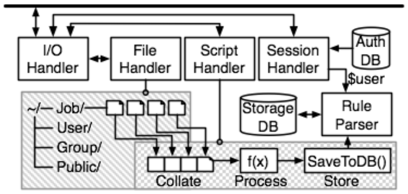


Figure 2. Server modules for receiving, processing and storing data

Each data packet that is uploaded to the server is stored as a single unencrypted file within a virtual scratch space. This space is generated by the File Handler and consists of symbolic links to folders within the file system that an authenticated session has read/write access to, with access to all other folders disabled. These folders include, but are not limited to, private storage (User/), shared storage belonging to the user's group (Group/), a public folder readable by any entity (Public/), and a cache folder representing the context in which actions triggered by communication-based requests takes place (Job/). Data packets would usually be uploaded to the *Job/* folder.

In this system, file-based storage is only intended for short-term use, with permanent data storage handled by a Structured Query Language (SQL) database system. Use of query-based systems such as this shifts the responsibility of identifying, evaluating and collating the requested data away from the user level to the database system framework. In doing so, the system can rely on well-established data acquisition techniques used within SQL frameworks to optimise performance. This structure also allows security features to be implemented by intercepting and manipulating query syntax, as is performed within the Rule Parser.

C. *Data Processing and Storage*

The transition from temporary file storage to long-term database storage is achieved through a process of collating the uploaded data packets into a contiguous data set, processing and filtering the data as necessary, and storing the result into the database. The exact procedure is context dependent and intended to be versatile, and as such, is suited to a scripting environment such as MATLAB [6]. For protection of the server [13], the code execution environment is sandboxed, with resource access controlled by the Script Handler. In scripts used to upload data, provisional access is given to files within the Job/ folder for input and the Storage Database (via the Rule Parser) for output.

D. *Data Protection*

The purpose of the Rule Parser is to prevent unauthorised access to data by validating and amending the syntax of all SQL queries before they reach the Storage Database. The current implementation focuses on asserting a straightforward permission directive – the content of a table row may only be updated by the same user who created it, although other users may also be given permission to read it. To enforce this, all tables within the SQL database contain two hidden fields (_owner and _readers), which are used to determine the access rights of the currently authenticated user ($user) specified by the Session Handler (see Figure 2).

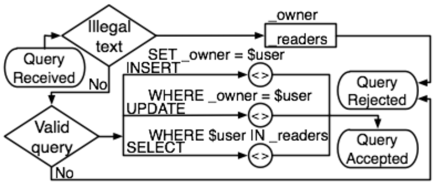


Figure 3. Query evaluation flowchart; simplified Rule Parser

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-207-3

A simplified representation of the algorithm used to enforce access rights is shown in Figure 3. Upon receiving a query, the Rule Parser verifies that the query text contains no mention of the words *_owner* or *_readers*, since either could constitute an attempt to manipulate or otherwise access hidden fields. If either keyword is found, the query is rejected. The query is also rejected if it is not an attempt to *INSERT* (create), *UPDATE* (modify) or *SELECT* (read) one or more rows. In the case of an *INSERT* statement, the query text is modified to define the *_owner* as being the *$user*. In an *UPDATE* statement, the query text is modified to ensure that *$user* is the *_owner* in all rows to be updated. In a *SELECT* statement, the query text is modified to ensure that *$user* exists within a list of *_readers*. It is noted that the full implementation will comprehensively evaluate syntax (including wildcards) to prevent misuse and provide functionality to add and remove users from the *_readers* list.

IV. Current Development

A health-based application is being developed using a smartphone (running Android OS) containing WiFi, location tracking, Bluetooth and an ANT receiver for communication with compatible wireless sensors. The initial development of this application will involve acquisition of data using an ANT compatible footpod. The sensor determines stride, cadence and distance using a triaxial accelerometer and transmits this information to the phone at a rate of 2Hz.

The application is implemented as a background service that commences during the boot sequence of the OS. This service initialises the ANT communications channel with the sensor and stores the received data into a temporary file. Location tracking provided by the phone is also utilised to obtain geographical position intermittently, with the option of GPS or network services depending on application requirements. In this system, network services are heavily favoured over GPS, trading off reduced accuracy for significantly reduced power consumption and time to first fix, as the hardware is only required to be active long enough to triangulate the user's position before switching off. The use of GPS allows its impact on battery life to be measured, as well as providing a relative measurement of accuracy.

Data upload to the server occurs based on the availability of WiFi and 3G networks. While a WiFi connection is present, data latency is kept low through frequent uploads. If WiFi is unavailable, data is accumulated in local storage until a connection becomes available. Once a local storage limit is reached, accumulated data is uploaded using the 3G network. If neither network is available, there is little choice but to notify the user and request immediate action.

Due to the dependence on network availability, data packets uploaded to the server may be unpredictable in size and content. At least once a day, the client will request that a script be run to collate, process and store all data currently held within uploaded packets. To do this, the script creates a contiguous stream of time domain samples, with each sample stored as an independent entry into the database. Each sample has an owner, timestamp, source (e.g., ANT or GPS) and a set of numeric fields used to describe sample content. An ANT sample will contain stride, cadence and distance, while a GPS sample will contain latitude, longitude, accuracy and time to first fix. In future development, the cloud server will process this data to produce useful statistics to users and their supervising practitioners.

V. Conclusions

This paper has outlined the design of a sensor bridge application and a communication and storage protocol to allow long-term multi-user sensor data to be securely uploaded to a remote centralised server. Further work will involve a validation of the prototype implementation to expand the capabilities of both the data collection and analysis aspects of the system. Due to the generalised design of the overall system, it can be applied to various real world applications in areas such as health and sport.

References

[1] C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov. "System architecture of a wireless body area sensor network for ubiquitous health monitoring", Journal of Mobile Multimedia, 1(4):307--326, 2006.

[2] E. Jovanov, K. Frith, F. Anderson, M. Milosevic and M. C. Shrove. "Real-time Monitoring of Occupational Stress of Nurses", Proc. of the 33rd Annual Conf. of the IEEE Eng. in Medicine and Biology Society 2011;2011:3640-3

[3] T. McNab, D. A. James, D. Rowlands, "iPhone sensor platforms: Applications to sports monitoring," Procedia Engineering, Volume 13, 2011, pp. 507-512

[4] C.L. Borgman, J. C. Wallis, M. S. Mayernik, A. Pepe. "Drowning in data: digital library architecture to support scientific use of embedded sensor networks", Proc. of the 7th ACM/IEEE JCDL, 2007, 269-277

[5] A. Bourouis, M. Feham and A. Bouchachia, "Ubiquitous Mobile Health Monitoring System for elderly (UMHMSE)", International Journal of Computer Science & Information Technology (IJCSIT), Volume 3, Issue 3, June 2011.

[6] J. R. Ride, D. A. James, J. B. Lee and D. D. Rowlands, "A distributed architecture for storing and processing multi-channel multi-sensor athlete performance data," Proc. of the 9th Conf. of the Intl. Sports Engineering Association, in press.

[7] A. A. Stone, S. Shiffman, J. E. Schwartz, J. E. Broderick and Michael R Hufford, "Patient compliance with paper and electronic diaries," Controlled Clinical Trials, Volume 24, Issue 2, April 2003, Pages 182-199

[8] Aaron Carroll , Gernot Heiser, An analysis of power consumption in a smartphone, Proceedings of the 2010 USENIX conference on USENIX annual technical conference, p.21-21, June 23-25, 2010, Boston, MA

[9] T. Dierks and C. Allen, "The TLS Protocol Version 1.0," IETF RFC 2246, 1999.

[10] F. Callegati, W. Cerroni and M. Ramilli, "Man-in-the-Middle Attack to the HTTPS Protocol," IEEE Security & Privacy, IEEE, 2009, pp. 78-81.

[11] V. Kumkar, A. Tiwari, P. Tiwari, A. Gupta and S. Shrawne, "Vulnerabilities of Wireless Security protocols (WEP and WPA2)", Intl. Journal of Advanced Research in Computer Engineering & Technology, Volume 1, Issue 2, April 2012.

[12] C. Jutla, "Encryption Modes with Almost Free Message Integrity," Advances in Cryptology - EUROCRYPT 2001, B. Pfitzmann (ed.), LNCS 2045, Springer Verlag, 2001.

[13] K. Gama and D. Donsez, "A self-healing component sandbox for untrustworthy third party code execution," Proc. of the 13th Intl. Symp. on Component-Based Software Engineering, Springer, 2010, pp. 130–149.

Configurations effects over Swarm Underwater Acoustic Network Performance

Marco Tabacchiera, Eva Marchetti, Silvello Betti
Department Of Electronics Engineering
University of Rome "Tor Vergata"
Rome, Italy
{tabacchiera, betti}@ing.uniroma2.it

Samuela Persia
Fondazione Ugo Bordoni
FUB
Rome, Italy
spersia@fub.it

Abstract— **An underwater swarm network for monitoring and exploration applications is proposed. Specifically, the behavior for different application scenarios with the corresponding performance analysis is presented. The design of an underwater swarm is discussed considering that the main requirement is maintaining the nodes power consumption as low as possible, without any increase of latency in the network. Results show that both average power and latency can be preserved by considering a reasonable number of hops needed to forward information from source to destination. It is verified that the results are influenced by the nodes motion related to the swarm configuration. This study aims to provide some guidelines to develop an autonomous underwater vehicles swarm for underwater applications.**

Keywords – swarm; underwater communications.

I. Introduction

Over the past few years, a growing interest has been showed for the Underwater Acoustic Networks (UANs) [1] [2]. Several studies have been performed to overcome limitations due to underwater environment, being the UANs affected by different factors influencing performance requirements with respect to the terrestrial ones. Mainly, the temporal and spatial variability, in combination with poor bandwidth availability of the underwater acoustic channel lead to consider the communication range limited and dramatically dependent on both range and frequency [3]. First underwater experimental systems were performed to demonstrate the hardware capabilities and the possibility of underwater communication architecture [4][5].

Our study is directed to a particular UAN, the Swarm Underwater Acoustic Network (SUAN), that is a network, in which the communication paradigm is not a trivial challenge, because the traditional peer-to-peer communication among Autonomous Underwater Vehicle (AUV) devices, as described in [6], is not applicable. Moreover, the particular scenario that is investigated, such as surveillance scenarios with self-organizing capabilities of the network, leads to consider both control and data traffic within the swarm.

Some studies have been performed for this scope, where Ad-Hoc networks [7], or cellular-type [8] networks have been considered to define the underwater network communication. Other studies consider the underwater network as a sensor network. In [9], for instance Fazel et al. consider the performance of a sensor network in terms of energy efficiency, even if their conclusions are not applicable to our case because they assumed a network composed of a large number of fixed nodes anchored to the bottom of the ocean for long period monitoring. The swarm configuration is instead characterized by an aggregate motion due to different modes operability, in which they are involved.

The outcome of this study is the basis for swarm nodes design, in order to define an underwater network for both monitoring and alarm detection applications. A preliminary study has been performed in [10], where the challenges needed at the protocol stack of the AUVs members of the swarm were analyzed. In this work a deeper investigation on swarm nodes motion has been carried out, and how this could impact over network performance has been further investigated in terms of number of hops involved during the forwarding information activities. It is mandatory to maintain this parameter as low as possible to obtain twofold enhancements

- to preserve the *lifetime* of the network: few nodes are involved in the forwarding and consequently their corresponding battery levels are conserved;
- to reduce *latency* in the network: few hops are needed and then less time is spent for sending packets in the very slow underwater channel.

The paper is organized as follows: a brief introduction of the considered application scenarios is provided in Section II, the system model is described in Section III, the test cases are described in Section IV, and the main results are showed in Section V. Finally, in Section VI, the main conclusions are drawn.

II. UNDERWATER SWARM DEFINITION

An underwater swarm network is characterized by nodes very close one to each other, with mobility capability. The structure of the network is that of a distributed network, in which the nodes, through the exchange of control information, will take decisions in collaborative manner.

This system will be able to work in two different modes, which correspond to two different application scenarios:

- Environment Monitoring (**Broadcast Scenario**), in which the nodes perform measurements of proper parameters, measurements of shallow water in the port area and short range communications are considered. From a communication point of view, it means that each node is able to communicate with its closest neighbor and to forward information towards a collecting node, which usually is a bottom node with could have also a radio equipment to communicate with a platform at the surface.

- Alarm detection (**Pipeline Scenario**): the swarm detects an alarm occurrence, for instance a measured value of a specific parameter (e.g., oil in the water) is higher than a given threshold in a specific region, and thus, it will be ready to coordinate itself and move towards the area, in which the anomalies have been detected. From a communication point of view, it means that each node is connected only to one next node and all the nodes are allocated in a linear manner. In this case a heavy data transmission is assumed in a directional way.

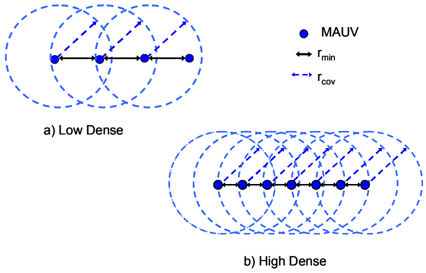
We remind that, a swarm is characterized by a more complex communication protocol than a peer-to-peer paradigm often applied to AUV devices, and thus the performance of the network will be strictly related to the solutions taken into account at each design level.

III. SYSTEM MODEL

We consider an underwater network of 10 swarm nodes located initially in random positions within a volume of 1000 m×1000 m×150 m. Each node has mobility capability. All the nodes within a coverage range r_{cov} from a given node are considered as its own neighbors. Generally, a network design is based on a suitable trade-off between the coverage radius of each component, and the available transmission power. In addition, to take into account real time data transmission (e.g., submarine video), also the latency constraints need to be considered. Furthermore, the particular mobility availabilities of the swarm nodes need to be analyzed. They have to move, in some situations, very close each to the other, and so interference has to be maintained as low as possible. From these considerations, we have found that all the possible solutions impact, at different levels, over the network performance.

A. Mobility

An important aspect to be considered is the nodes mobility, which determines the different applicative scenarios under test. A 3D model is considered: it is assumed that the position information are stored in all the nodes, and thus a perfect knowledge of exact neighbors' positions is available at any time. The two proposed scenarios imply different algorithms to model nodes motion.



a) Low Dense

b) High Dense

Figure 1. Pipeline scenarios: low (a) and high dense (b) configurations.

Pipeline Scenario - In this scenario the nodes move along the linear conjunction source to destination node. They may assume a dense or a sparse configuration, as depicted in Fig. 1.

Broadcast Scenario – In this case, a control algorithm is implemented able to elaborate an ellipsoid dimension, in which each node has to move inside. The ellipsoid dimensions are dependent on two parameters: the number of nodes, *N* and the maximum "dimension" of the swarm (e.g., the more distance nodes possible within the swarm), *D*.

Afterwards, one of the nodes will be elected as the "center" of the ellipsoid (this choice could depend on the target of the swarm, as well as the final configuration selection, i.e., ellipsoid or sphere) and then, others nodes, located outside the ellipsoid, will have to move inside it.

Each of those nodes will move towards the direction of the center of the ellipsoid as far as a distance r_{min} from the node to the surface of the ellipsoid has been reached. r_{min} represents the minimum physical distance, at which the AUV devices have to stay to avoid to crash one to each other. Even in this case, the nodes could stay very close one to each other (high dense configuration) or not (low dense configuration).

A possible sequence of the algorithm states is depicted in Fig. 2. Note that, when the nodes have reached the ellipsoid (Fig. 2b), they form three sub-clusters around the two focuses and the center of the ellipsoid, respectively (each node will choose the closest of these points). To save power, a node stops when has reached a connection to one of the possible sub-clusters (Fig. 2c). After that, sub-clusters will join together by moving their extreme nodes towards the other sub-clusters. During this operation, if a node loses connection from its sub-cluster of origin, another node of its group will "follow" it. Iterating this operation will force the swarm to occupy the maximum possible area of the ellipsoid (Fig. 2d), without the use of a central control.

B. Physical Layer

For the Physical Layer technology, a system based on underwater acoustic signals propagation is considered.

Compared to traditional AUNs, in SUAN the communication range is very short and can be vary from 3 to 100 meters.

In [10], to obtain a good trade-off between bandwidth and efficiency, an isotropic transducer operating at 300 kHz was considered. This value is higher than those used in traditional UAN, but at the same time implies a less harmful multipath effect. The modulation format adopted for acoustic shallow water channel is a Multi-Frequency Shift Keying (M-FSK) with M=4, 8 and 16. This modulation format is more prone to contrast the multipath effects and can allow a low cost modem implementation. Generally, the Signal-to-Noise Ratio (*SNR*) per bit can be evaluated as [7]

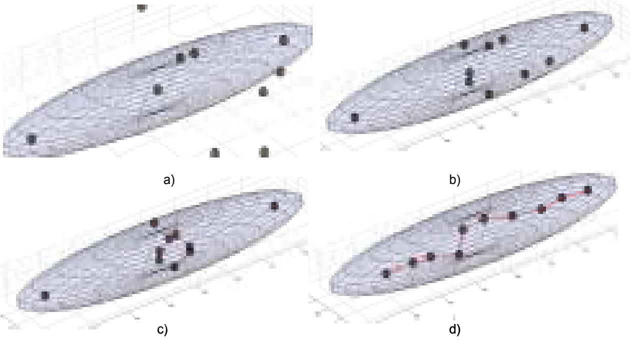


Figure 2. Mobility algorithm: a sequence of a possible configuration: (a) initial nodes' random positions; (b) the nodes enter in the ellipsoid; (c) the formation of the three sub-clusters; (d) final configuration.

$$SNR(d,f) = \frac{P_{TX}(d)}{A(d,f) \cdot N_{Tot}(f)} \tag{1}$$

where P_{TX} is the transmitted power, $A(d,f)$ the attenuation over the link between the transmitting node T_x and the receiving node R_x. $N_{Tot}(f)$ is the overall ambient noise due to turbulence, shipping, waves and thermal noise.

The attenuation is given by [7]

$$A(d,f) = d^k \cdot a^d \tag{2}$$

where k is the energy spreading factor (k is 1 for cylindrical, and 2 for spherical spreading) and

$$a = 10^{\frac{\alpha(f)}{10}} \tag{3}$$

is a frequency-dependent term depending on the absorption coefficient $\alpha(f)$. The absorption coefficient for the frequency range of interest is calculated according to Thorp's expression [11], expressed in dB/km and f in kHz.

A packet is correctly received if the *SNR* (1) exceeds a proper threshold, which can be evaluated by considering the sensitivity level of the receiver. This value is obtained by the data sheets specifications of the transducers. Specifically, for our simulations, an ITC-1089D transducer has been considered. Its specifications are: Receiving Voltage Sensitivity (RVS) of -218 dB re1V/1μPa and Transmit Voltage Response (TVR) exceeding 148 dB re 1μPa/V@1m.

C. Medium Access Control Layer

Different solutions have been proposed in literature for Medium Access Control (MAC) Layer design [12]. For low data rate, with rare collision events, a very simple solution such as ALOHA scheme or its improved version, can be considered. In particular, we have assumed an ALOHA without interference phenomena. Future studies will be necessary to evaluate the effects of the MAC over the system reliability. Hence, a Collision Probability, P_{coll}, will be included in the proposed model.

D. Network Layer

To design a reasonable swarm, two aspects need to be verified: energy saving and latency reduction. These requirements suggest to consider at network level a multi-hop paradigm to forward data among the swarm in order to obtain a reasonable trade-off between the above mentioned opposite factors. The network performance can be thus expressed by the End-to-End Frame Error Probability (*FEP*) for a multi-hop route, which depends on the number of hops needed to forward information from source to destination.

FEP can be evaluated as in [7], resulting

$$FEP = 1 - (1 - P_b)^{L \cdot n_h} \tag{4}$$

where P_b is the bit error probability for a single node-to-node link, L the frame size in bits, and n_h is the number of hops needed to forward data within the swarm. Obviously, n_h strictly depends in what configuration the swarm is, and then by its geometrical characteristics. The bit error probability depends on both the modulation format and the propagation channel. An M-FSK modulation format over a Rayleigh fading channel is considered. Therefore, the P_b of an M-ary orthogonal signal can be expressed as [13]

$$P_b = \frac{M/2}{M-1} \cdot \sum_{m=1}^{M-1} \frac{(-1)^{(m+1)} \cdot \binom{M-1}{m}}{1 + m + m\gamma} \tag{5}$$

where M is the level of the M-FSK modulation format, and γ the linear expression of the *SNR* resulting from (1).

IV. TEST CASES

Performance analysis has been repeated for both the proposed scenarios. Different considerations have been carried out for each case, strictly related to the different mobility models that swarm can assume. Specifically,

- **Pipeline Scenario -** AUVs place themselves along the linear conjunction between source to destination to forward information in efficient manner.
- **Broadcast Scenario** - The swarm can assume two different geometrical shapes: Sphere or Ellipsoid.

For each case, it is possible consider two configurations:

- *Low dense configuration:* AUVs assume a uniform distribution inside the limited area to maximize the swarm coverage area.

- *High dense configuration:* AUVs are located very close each to other. They are at the minimum distance permitted by their physical dimension.

V. Performance Analysis And Trade-Offs

Performance analysis has been carried out by Matlab [14] and system performance has been evaluated and compared with different test cases described in Section IV. More deeply, we have investigated the performance of the swarm network in terms of *FEP* according to [4]. We have firstly evaluated the *FEP* by considering the number of hops needed to reach destination by theoretical considerations in different test cases, taking into account the main system constraints: the minimum distance, r_{min} , and the coverage radius of each node, r_{cov}. After we have evaluated n_h by considering the outcome by both mobility, and physical simulations.

Specifically:

- the outcome by the mobility model will permit to evaluate the distance of each node in different configurations;
- the outcome by the physical model will permit to evaluate the corresponding *SNR* (at each distance calculated by the mobility schemes) of the transmitted signal for each node. The average SNR value for each test case will be considered for the *FEP* evaluation.

These evaluations have been performed for each M-ary modulation format. We proposed some equations to define a number of hops for each scenario as described in the following subsections.

A. Pipeline Scenario

We have evaluated the number of hops taking into account geometric configurations of the swarm for both high and low dense case. We have reported only theoretical considerations because no difference has been found by simulation results.

- **Low Dense Pipeline** – In this case the nodes are located at the maximum distance allowable for a fully connected swarm, r_{cov} (Fig. 1a), and the number of hops is given by

$$n_{h_{LP}} = N - 1 . \tag{6}$$

- **High Dense Pipeline –** In this case the nodes are located at the minimum distance allowable, r_{min} (Fig. 1b) and thus the number of hops to cover the distance D is obtained as the ratio D to r_{cov}

$$n_{h_{HP}} = \frac{D}{r_{\text{cov}}} = \frac{(N-1)\cdot r_{\min}}{r_{\text{cov}}} = (N-1)\cdot \frac{r_{\min}}{r_{\text{cov}}} . \tag{7}$$

B. Broadcast Scenario

For this scenario we have firstly evaluated n_h with theoretical considerations, after we have repeated the same analysis by considering the outcome of the mobility simulator.

1) **Theoretical Model** – The model is based on geometrical considerations of the different configurations that the swarm assumes. For each test case, the maximum number of hops needed to forward information from source to destination has been evaluated.

a) High Dense Sphere Configuration - The maximum number of nodes can be evaluated according to (8), as the ratio between two sphere volumes: the Physical Sphere, V_{min} and the Coverage Sphere V_{cov}. This assumption is justified by considering that each node is very close to each other, and thus the Coverage Sphere of each node overlaps with the other ones (Fig. 3b),

$$n_{h_{DS}} = \frac{V_{\min}}{V_{\text{cov}}} = \frac{N\frac{4}{3}\pi\cdot r_{\min}^3}{\frac{4}{3}\pi\cdot r_{\text{cov}}^3} = N\cdot\left(\frac{r_{\min}}{r_{\text{cov}}}\right)^3 . \tag{8}$$

b) Low Dense Sphere Configuration – The number of hops can be evaluated according to (9), where D is the diameter of the sphere, in which the swarm can stay and d is the maximum distance between two neighbors nodes, which corresponds, in the worst case, to r_{cov} (Fig. 3a). The diameter D can be calculated as a function of the volume of the sphere that (in the worst case) contains N spheres of r_{cov} /2 radius. In this case, the maximum hops number is

$$n_{h_{LS}} = \frac{D}{d} = \frac{\sqrt[3]{\frac{6}{\pi}V}}{r_{\text{cov}}} = \frac{\sqrt[3]{\frac{6}{\pi}N\frac{4}{3}\pi\left(\frac{r_{\text{cov}}}{2}\right)^3}}{r_{\text{cov}}} = \sqrt[3]{N} . \tag{9}$$

c) High Dense Ellipsoid Configuration – As in the high dense sphere case, the nodes are very close one to each other within a “thin” sphere that converges to an ellipsoid; if the ellipsoid is very thin means that the swarm assumes a quasi-linear configuration and thus the number of hops can be calculated by (7), whereas if the ellipsoid has its geometrical parameters of the same value (i.e., a=b=c), the ellipsoid converges to the sphere case and the number of hops is expressed by (8). For all the other cases, the number of hops is given by

$$N\cdot\left(\frac{r_{\min}}{r_{\text{cov}}}\right)^3 < n_{h_{HE}} < (N-1)\cdot\frac{r_{\min}}{r_{\text{cov}}} . \tag{10}$$

d) Low Dense Ellipsoid Configuration – As in the low sphere configuration, the maximum distance allowable between two nodes is determined by the minimum *SNR* required by $r_{cov.}$ If the ellipsoid is very thin, it collapses to a linear distance such as the pipeline, and the number of hops is equal to *N*-1, while if the ellipsoid collapses to a sphere, the number of hops can be expressed by (9). Even in this case the average configuration with a canonic ellipsoid leads to a number of hop as:

$$\sqrt[3]{N} < n_{h_{LE}} < N - 1. \qquad (11)$$

2) **Numerical Results** – The results obtained by using theoretical approaches have been confirmed by simulation outcomes. We have simulated the swarm mobility models and after evaluated the number of hops needed for each configuration. In our simulations, we assume 10 AUVs in the swarm, with r_{cov}=80 m and r_{phy} =3 m. The output of the mobility algorithm is the *Hop Matrix*, M_{Hop}, in which the i-th column represents the number of hops needed to the i-th node to reach the j-th node. We have verified that:

a) High Dense Sphere Configuration – Each node is directly connected to all the others, and then the $\mathrm{M}_{\mathrm{Hop}}$ will be composed of all 1, i.e., only one hop is needed to reach the destination.

b) Low Dense Sphere Configuration – In this case the details of the M_{Hop} are shown in Fig. 4, in which we can note that the maximum number of hops is 6, due to the spherical symmetry of the nodes configuration.

c) High Dense Ellipsoid configuration – Even in this case the M_{Hop} matrix is fulfilled as the High Dense Sphere one.

d) Low Dense Ellipsoid Configuration – The number of hops increases significantly with respect to the previous case, due to the increase of the swarm area.

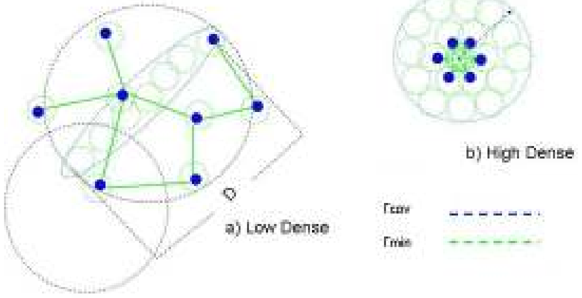


Figure 3. Sphere scenarios: low (a) and high (b) dense configurations.

$$\begin{pmatrix} 1&4&2&1&3&5&1&2&5&3\\ 4&1&5&3&1&1&4&2&1&2\\ 2&5&1&2&4&6&1&3&6&4\\ 1&3&2&1&2&4&1&1&4&2\\ 3&1&4&2&1&2&3&1&2&1\\ 5&1&6&4&2&1&5&3&1&3\\ 1&4&1&1&3&5&1&2&5&3\\ 2&2&3&1&1&3&2&1&3&1\\ 5&1&6&4&2&1&5&3&1&3\\ 3&2&4&2&1&3&3&1&3&1 \end{pmatrix} \quad \begin{pmatrix} 1&4&2&6&2&7&3&1&1&5\\ 4&1&6&2&2&3&1&5&3&1\\ 2&6&1&8&4&9&5&1&3&7\\ 6&2&8&1&4&1&3&7&5&1\\ 2&2&4&4&1&5&1&3&1&3\\ 7&3&9&1&5&1&4&8&6&2\\ 3&1&5&3&1&4&1&4&2&2\\ 1&5&1&7&3&8&4&1&2&6\\ 1&3&3&5&1&6&2&2&1&4\\ 5&1&7&1&3&2&2&6&4&1 \end{pmatrix}$$

Figure 4. Hop Matrix – Low Dense Sphere Configuration (left) and Low Dense Ellipsoid Configuration (right).

In this case, the maximum number of hops reaches the Pipeline Low case, i.e., 9 hops. It means that all the nodes are involved in the forwarding activities (Fig. 4).

Figures 5-8 show the performance in terms of *FEP* for each case under test for both theoretical and numerical results, respectively. The evaluations have been repeated for M-FSK with *M*=4,8, and 16. In the figures only 4-FSK and 16-FSK have been reported because the 8-FSK results appeared not significantly different from the 4-FSK ones.

Specifically, Fig. 5 shows the theoretical *FEP* evaluation for pipeline scenarios. Figures 6-7 show *FEP* evaluations for low dense sphere and ellipsoid, respectively by considering both theoretical and numerical models. Fig. 8 depicts high dense sphere and ellipsoid cases. In the latter case theoretical and numerical results are coincident, and thus only one (theoretical) has been reported. We have verified that 16-FSK requires a lower *SNR* value than 4-FSK to achieve the same *FEP* level. This attitude becomes more evident in the high dense cases. The sphere configurations are characterized by better performance with respect to the other ones, while the ellipsoid scenarios assume middle performance between the sphere and the pipeline cases. All the theoretical considerations are confirmed by numerical results, showing a good confidence level for the proposed mobility algorithm.

VI. CONLCUSIONS AND FUTURE WORK

Underwater swarm networks have been considered to define the requirements to be satisfied in different application scenarios. Performance evaluations by comparing different operability modes for a swarm network have been carried out. By numerical results we have verified that the performance strictly depends on the different swarm configurations. The power consumption and the latency have to be taken into account, which are the main constraints for alert applications. A good trade-off has to be achieved to obtain solutions suitable to real underwater context. Future studies will be to consider a more complex physical layer model, which includes environment information and its variability (i.e., temperature and salinity profiles), remembering that in underwater network design the upper layers of nodes are highly constrained by the physical layer parameters.

ACKNOWLEDGMENT

This work has been supported by ”HARNESS” (Human telecontrolled Adaptive Robotic Network of SensorS) National Project.

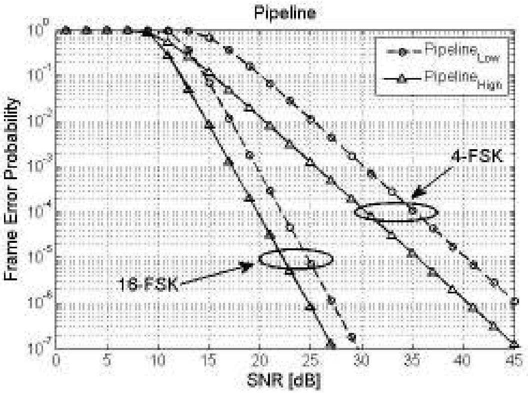


Figure 5. FEP comparison (theoretical results) for different pipeline scenarios.

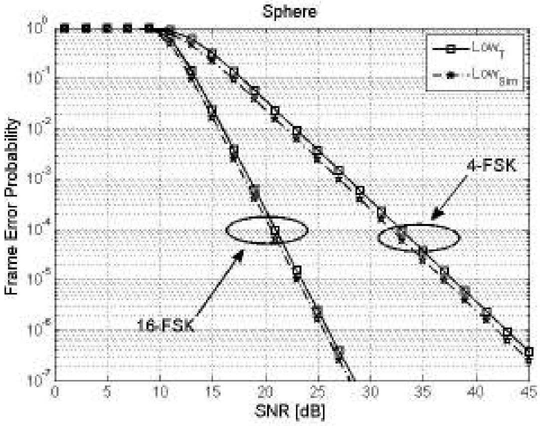


Figure 6. FEP comparison (theoretical and numerical results) for different sphere low dense configurations.

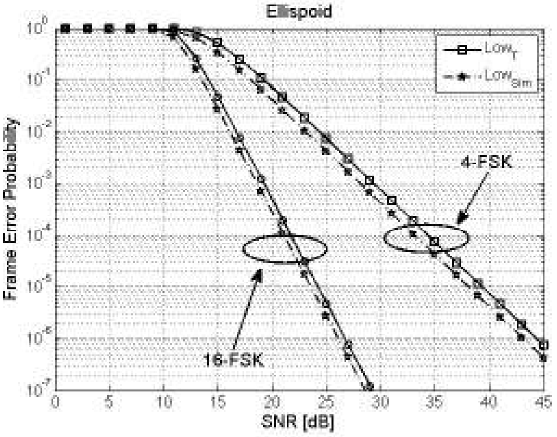


Figure 7. FEP comparison (theoretical and numerical results) for different ellipsoid low dense configurations.

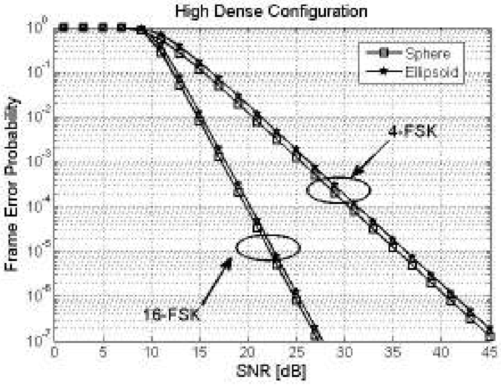


Figure 8. FEP comparison (theoretical results) for different high dense configurations.

REFERENCES

[1] E. M. Sozer, M. Stojanovic, and J. G. Proakis, ”Underwater acoustic networks,” Oceanic Engineering, IEEE Journal of, 2000, vol. 25, pp. 72-83.

[2] J. G. Proakis, E. M. Sozer, J. A. Rice and M. Stojanovic, “Shallow Water Acoustic Networks,” IEEE Communications Magazine, November 2001, pp. 114-119.

[3] I. F. Akyildiz, D. Pompili, and T. Melodia, ”Challenges for Efficient Communication in Underwater Acoustic Sensor Networks,” ACM Sigbed Review, vol. 1, no. 2, July 2004.

[4] J. Rice, ”SeaWeb acoustic communication and navigation networks,” in Proc. Int. Conf. Underwater Acoust. Meas.: Technol. Results, Crete, Grece, Jun. 2005, pp. 2007 - 2017.

[5] M. Grund, L. Freitag, J. Preisig, and K. Ball, ”The PLUSNet underwater communications system: Acoustic telemetry for undensea surveillance,” in Proc. IEEE OCEANS’06 Conf. on, Boston, MA, Sep. 2006.

[6] S. M. Smith, J. C. Park, and A. Neel, ”A peer-to-peer communication protocol for underwater acoustic communication,” in OCEANS’97, Halifax, Nova Scotia, Oct. 1997, pp. 268-272.

[7] A. Stefanov and M. Stojanovic, ”Design and Performance Analysis of Underwater Acoustic Networks,” IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Advances in Military Communications and Networking, Vol. 29, Issue 10, pp. 2012-2021.

[8] M. Stojanovic, ”Design and Capacity Analysis of Cellular Type Underwater Acoustic Networks,” IEEE Journal of Oceanic Engineering, Vol.33, No.2, Apr. 2008, pp. 171-181.

[9] F. Fazel, M. Fazel, and M. Stojanovic, ”Random Access Compressed Sensing for Energy-Efficient Underwater Sensor Networks,” IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Energy-Efficient Wireless Communications, Vol.29, n. 8, Sep. 2011.

[10] M. Tabacchiera, S. Persia, C. Lodovisi, and S. Betti, ”Performance Analysis of Underwater Swarm Sensor Networks,” ’Broadband and Wireless Computing, Communication and Applications (BWCCA), 2011 Int. Conf. on, Barcellona, Spain Oct. 2011, pp. 333-338.

[11] L. Berkhovskikh and Y. Lysanov, ”Fundamentals of Ocean Acoustics,” New York, Springer, 1982.

[12] C. Petrioli, R. Petroccia, and M. Stojanovic, ”A Comparative Performance Evaluation of MAC Protocols for Underwater Sensor Networks,” in Proc. IEEE Oceans’08 Conf. on, Quebec City, Canada, Sep. 2008.

[13] J. G. Proakis, “Digital Communications” 2nd Ed., McGraw-Hill, 1989.

[14] http://www.mathworks.com

The Localization Problem for Harness: A Multipurpose Robotic Swarm

Ramiro dell'Erba
Italian National Agency for New Technologies, Energy and Sustainable Economic Development Robotic Laboratory
ENEA
Rome, Italy
dellerba@enea.it

Claudio Moriconi
Italian National Agency for New Technologies, Energy and Sustainable Economic Development Robotic Laboratory
ENEA
Rome, Italy
moriconi@enea.it

***Abstract*—This paper deals with the localization problem of an underwater robotic swarm in the context of the HARNESS project (Human telecontrolled Adaptive Robotic Network of SensorS) currently in progress in our laboratory. This system is based on cheap autonomous underwater vehicles (AUV) organized with swarm rules and conceived to perform tasks, ranging from environmental monitoring to terrorism attack surveillance. The key aims of the HARNESS project are: the development of a novel underwater acoustic channel with very high performances in routing and data throughput capacities; and the design of a reliable swarm rule-based control system with an interface dealing with a supervisor operator acting as a priority definition arbiter. A method to determine the shape of the swarm, based on trilateration calculation, is proposed.**

Keywords-swarm; underwater; robot; localization.

I. INTRODUCTION

This paper deals with the localization problem of an underwater robotic swarm.

The HARNESS project [1] seeks to realize an underwater multi AUV robotic system, arranged in a swarm organization where the classical flocking rules and the Communication Network protocol are merged in a novel higher level control. It is expected to improve the performance of classical AUV technology exploiting the large occupied volume and the short distances among the vessels. The speed of surface monitoring and the transmission bandpass among the vessels and towards the surface should be some of the most important results. The availability of a suitable robotic swarm could be relevant in many operations: surveillance of sensitive sites, fast exploration of relatively wide areas of interest, detailed analysis of objects (i.e., archeological artifacts) without removing them from their underwater sites.

The most interesting areas to explore and protect are those in proximity to coasts and with depths ranging between 50 to 200 m. Professional and expensive divers can operate only to a maximum of 70-80 meters in recovery operations, and simple and fast explorations cannot be performed by humans beyond 100 m.

On the other hand the use of rovers has proven to be useful in many cases, but generally expensive mainly because they require the support of an equipped ship.

The use of robotic technologies in ocean surveys, inspections, pipe and cable tracking, has been well established in the field of marine engineering for many years [2] with an important increase in performance in recent years [3] as many autonomous underwater vehicle systems moved from the prototype stage to scientific, commercial, and military uses.

An AUV must be considered as a real cost alternative to other available technologies, such as manned submersibles, remotely operated vehicles (ROVs) and towed instruments led by ships. However, many problems are still to be solved to make AUV competitive especially for the issues relevant to power availability, information processing, navigation, and control.

The goal of this paper is to give a presentation of the novel concepts in Harness. We will discuss advantages and drawbacks of the project concepts and explain the prototype under construction. Finally we will focus on the crucial problem of localization of the swarm with some proposal for an efficient solution suitable to some scenarios.

II. RELATED WORK

The expected features of Harness could limit the use of expensive surface ships to the deployment phase; moreover it takes advantage of the parallel exploration of many cheap AUVs [4] to reduce work time.

The concept of robot swarms has been a research theme of the scientific community for several years. The realization of swarms of different numbers of cooperating robots has been successfully attempted, but in the underwater environment it is still a challenge. Many of the difficulties are relevant to the lack of fast and reliable communication links. Swarm research has been inspired by biological behaviors, like the one of bees [5][6] to take advantage of social activity concepts [7], labor division, task cooperation and information sharing. A single-AUV approach is affected by operational limits the lack of effective communication and the limitation of available sensing that may reduce its effectiveness. On the contrary, a multi-robot approach can benefit from its parallel operation and from redundancy and greater robustness allowed by the use of multiple agents.

Another theoretical advantage of the swarm, considered as a whole entity that we intend to investigate lies in the possibility of performing parallel computation from realizing a distributed perception. As an example, landmark recognition is a task that can be shared among the different nodes taking advantage of the different viewpoints of the same target and applying a voting process to increase recognition performance. An adequate communication network must be available, in order to allow at least the exchange of the target identification features, but the network under design, based on multiple channels with carrier frequencies among 0,3 to 2 MHz promises to be adequate.

This form of distributed perception is a novel attempt that cannot be effectively demonstrated in a simulated scenario apart for the implementation aspects. The real experiment is therefore expected at the end of the project, in a couple of years' time, when a basic swarm of at least ten vessels with relevant sensors and communication network will be available

Another project strength is the capability to adopt a flexible geometrical distribution of the members depending on the task and environment characteristics in particular for communication. In the underwater world the physical medium makes the acoustical channel the most convenient one, since electromagnetic waves are very rapidly dampened. Acoustical propagation can offer better performance, but a number of effects must equally be taken into account. High frequency carriers are dampened too, even if less dramatically then their E.M. analogue. Moreover when the frequency increases, the propagation within the medium shows an extremely narrow direction pattern that prevents the easy dispatch of a message among the many nodes with mobile and imprecisely identified positions.

The solution of this puzzle takes advantage of the swarm organization and of the possibility of a mutual interaction between the spatial distribution of transmission nodes (the swarm members themselves) regulated by the flocking rules and the transmission protocol. In Figure 1 and Figure 2 two examples of adaptive localization of the swarm, for different tasks, are shown. The exploitation of ultra-high frequencies is enabled by the shortening of distances with positive effects also on the multi-path effect and the consequent intersymbolic interferences. On the other hand a task request for a wider swarm configuration can stimulate the flexible protocol to select lower frequencies and slow down the communication rate.

One of the aims of the project is the study and implementation of different behaviors in the swarm, to generate a collective shaping as a response to environmental stimula and to modify the communication parameters in order to maximize the performance of the system [8].

In this case the swarm control must balance the different requests of the operator (e.g., modify the mission task), the swarm needs and the single member's management (e.g., obstacle avoidance, loss of communication link).

The result is the selection of collective behaviors that must be compatible with all the aforementioned conditions. One of the peculiar approaches of Harness is the aim of controlling a swarm as a whole by a human supervisor. Because of the unique control concept of the swarm this action has to be carried out by means of a non-conventional procedure. Human commands, given in intrinsically fuzzy channels (gesture recognition / voice commands) are converted into flocking rule modifications. In this first phase the research is limited to the change of the rule parameters; the next step will be a selection from amongst a different set of rules; and the final goal (if the field tests clearly show the need) will be the automatic synthesis of special rules.

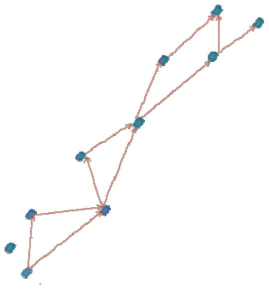


Figure 1. "Pipe", configuration indicates when the communication is the main objective of the geometrical shape to transport data on long distances at the maximum allowed speed.

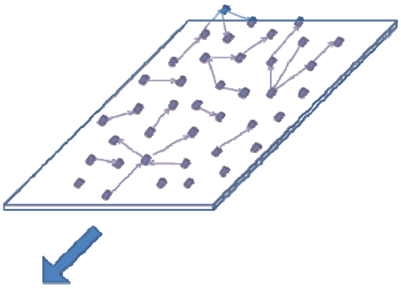


Figure 2. Planar distribution to carry out fast and parallel monitor operations.

The communication between swarm and human requires a special function. An element of the swarm has to emerge, navigate to the sea surface and activate an RF link with the console while it continues to maintain the link with the other elements.

III. The swarm Vessel Prototype

In Figure 3 the Venus prototype, a torpedo type realized in our laboratory, is shown. Its characteristics are the following:

Max depth 100m; Max speed 2 Knots; Weight about 20 Kg; Autonomy 3hrs; Dimensions 1.20 m length 0.20 m diameter.

Standard sensors include a stereoscopic camera, sonar, accelerometer, compass, depth meter, hydrophones side-scan sonar.

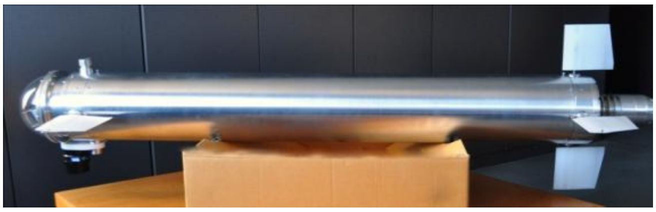


Figure 3. Low cost Venus AUV

We are dealing with a system thought to be a component of a swarm of about 20 objects. The distances between robots ranging between 3 and 50 meters. Therefore, the maximum distance possible between two robots is about 1000 meters, as a very particular alignment case; the average value of the distances is about 10 meters. Maximum speed should be about 2 knots.

An optical, high power, transmission device will be used for a number of different experimental approaches integrating the acoustical data channel and the direct vision sensing.

Optical methods are very powerful but their performances are affected by many strongly variable parameters like salinity, turbidity, the presence of dissolved substances that change the color and the transparency in different optical bands and the amount of solar radiation that heavily affect the signal to noise ratio.

The current approach uses a strategy based on the variable exploitation of the optical channel depending on the environmental conditions. In favorable conditions the transmission protocol will freely decide which channel to adopt depending on the priority, distance-to-cover and dimension of the message itself. In less favorable conditions the optical channel will be limited to the fundamental synchronization task, generating a light lamp that will optimize the message passing through the optical channel (and several other non strictly communicating functions). In poor optical conditions strong but very short lamps, will ensure references for a safer visual navigation.

IV. THE LOCALIZATION PROBLEM

Localization and mapping is the key of a successful navigation in autonomous mobile platform technology and is a fundamental task in order to achieve high levels of robustness in vehicle positioning and values of the collected data. Robot localization and mapping is commonly related to cartography, not only for geographic purposes, but for biological and ecological studies, for geophysical researches and for security surveillance. It combines science, technique and computation to build an environment representation to correlate spatial information with the data collected [5][6].

Compared to a single AUV, a multibody system has the need to know and manage its own configuration. This can be an advantage if we are able to profit from the large number of data points to reduce errors in positioning. The localization problem of a swarm can be divided into three tasks:

1. Absolute localization, with the meaning of localization of a member or of a geometrical locus with respect to a fixed reference system (AL)
2. Relative localization of (RL) i.e., the swarm configuration.
3. Relative localization of one member with respect to neighbors; we call it immediate relative localization (IRL) and its meaning will be clarified later.

Different methodologies are required to solve the three tasks. In line with the swarm philosophy each element must be able, if connected with the others, to perform the localization task. However, the reader’s attention is drawn to this last point. We do not mean that each element must always do all of the tasks; often there is no need to know all of the machines’ positions, but each vessel must be capable of doing so using all the internal data, the external data communicated by the other elements and the external data measured by the robots, including that deduced by environmental observation.

The usefulness of IRL is in achieving a rapid response to environmental modifications. If the system has to wait for a thorough knowledge of its internal structure before deciding the type of response the reaction times could become too slow, compared to the dynamics of environmental phenomena.

Schools of fishes change their position in a very quick time just looking the movement of the neighbors. In principle also IRL is more easily achieved with simple and robust signals based on physical fast transient, and possibly using a rapid propagation of fields such as optical. Later, if necessary, the swarm can re-compute the relative position of each member.

The knowledge of the whole configuration can be useful, for instance, in some approaches to the geographic identification of a site: the possibility to collect information from many “points of view”, by a swarm, allows a “distributed site recognition” (a task that we are starting to study). This technique leads to landmark recognition with less ambiguity. We distinguish the problems because they are separate: localization of a single entity is a problem relevant to understanding a metric of the space, possibly realized only by a sequence of reference points, and to placing the entity in relation to the established metrics or with to reference points. Localization of a swarm underlines the different techniques that can be used together with the different information that can be gained.

For many animals localization may be related to different types of metric linked to the search for something critical for their life like food. In fact this localization could relate to the identification of a source of chemical compound that is attractive for the biological or robotic system (which could be blood to indicate food or something else indicating pollution); the result is a physical field with gradients of arbitrarily complex shape. Our position is therefore referred to the smell, great or small, of our reference parameters. We can move toward the higher value of the field to reach the source. In this case the geometric or geographic location is meaningless for the success of the mission except when the mission itself is finished and the last task is the site identification for further actions. We are locating ourselves in the higher concentration food position.

The previous example of gradient localization is an example of topological localization. Its realization is similar to the force field potential applied in many obstacle avoidance techniques.

Two different environment representations can be used: metric and topological world models. A metric environment map, using a unique coordinate system, is theoretically easy to implement when you are using distance sensors. However, metric maps are not well suited to integrate non-metric information as required if you have to match patterns. Moreover, metrically consistent map building is a non-trivial problem.

Topological representation models of the world make use of graphs of nodes (distinctive places) and connecting edges (pathways). The advantage is simplified map building (path planning is reduced to direct graph search); moreover sensor fusion of non-metric and metric data is easier, because they are uniformly treated as attributes to nodes and/or edges.

However, topological world models, typically, are characterized by poor resolution, which is insufficient for any other purpose than navigating from one region to another.

Another example is the kinesthetic sense of human body; it works with a sensitivity to muscle tension and extension; the brain relates these variables to locate the position of the body in space.

It is important to note that such a kind of a map is a multidimensional set of associations of features (houses, streams, odors) with usually non-linear metrics that allow you to switch gradually from a certain place to another place via a route typically, but not necessarily, geographical.

Localization of a swarm encompasses different techniques that can be used together with the different information that can be gained by putting them together. This can become a great advantage.

The easiest localization system is the open loop estimation; this means that the estimate of position is based on expected results of motion commands. Therefore no contribution from the sensor is required and no feedback is calculated.

The information that the robot gathers can be divided into two kinds; idiothetic and allothetic sources. The distinction relates to internal or external sensor source data; as an example, if a robot is counting the number of wheel turns this is an internal source.

The allothetic source corresponds to the sensors of the robot, like a camera, a microphone, laser or sonar. A typical problem of this last method is "perceptual aliasing"; this means that two different places can be perceived as the same. For example, in a building, it may be impossible to determine your location (sometimes also for humans relying solely on visual information, because all the corridors may look the same). Without an external reference, like for example, acoustic beacons at known positions, the vehicle has to rely on proprioceptive information obtained through a compass, a Doppler Velocity Logger (DVL) or an Inertial Navigation System (INS) [6]. To this, totally internal to the robot family belongs one of commonest methods, the dead reckoning [9.

A common dead reckoning sensor is the INS. An INS measures the linear acceleration and the angular velocity of the vehicle using three accelerometers and three gyroscopes. Typical underwater external sensors used to correct accumulated errors from the integration of the INS measurements, are Doppler Velocity Log Sensors (DVL), Ultra Short Baseline (USBL) and Differential Global Position Systems (DGPS/GPS); the latter only in the case that the vehicle is operating in shallow waters and can come out of the water to fix (and eventually communicate) the position.

Independent of the quality of the sensors used, the error in the position estimate based on dead-reckoning information grows without upper limit. Typical navigation errors are about some per cent of distance traveled for vehicles traveling within 100 meters of sea. Lower errors can be obtained with large and expensive INS systems, but for vehicles relying only on a compass and a speed estimate these can be higher than 10%, after 100 meters. The error can be reset if the AUV comes to the surface by GPS, but sometimes this is impossible (under ice for example) or undesirable (security operation) [8]. The use of beacons to form a Long Baseline (LBL) array limits the operation area to a few square kilometers and requires a substantial deployment effort, to position the beacons, before operations, especially in deep water. This reduces the advantages of AUV and requires an expensive ship to support the operation. A swarm could be advantageous compared to a single vessel, if a high rate of communication is available, to reduce the dead reckoning errors. It can collect together all the data of all the vessels to minimize the errors in estimating position.

Other methods employ the use of landmarks. If we use external references in the localization problem, like humans, we have to deal with their definitions and position on a map (metric or not). Any kind of landmark is subject to classification and identification based on its attributes. Unfortunately the identification and positioning of a landmark often suffers from ambiguity, owing to the multiple solutions of the associate equations. That has the meaning of more than one landmark is identified by the same features.

A set of features' location position estimates can basically be thought of as a map. The challenge is to combine INS/dead-reckoning and other information with sensor observations of features to build a map, locally or globally referenced.

A more modern idea consists of matching measurements of one or more geophysical properties, such as bathymetry, gravity, or magnetic field, to a known environment map. If there is sufficient spatial variation in the parameters being measured, there is potential to reduce navigation uncertainty. As an example the marine turtle's migration is monitored by magnetometers, measuring the earth's magnetic field variations. However, often, these techniques require a map of the environment that is not available. The marine turtle's migrations are monitored by a three axis magnetometer, but the method's resolution is only 35 nautical miles [10][11].

V. Our Proposal

If we have three or (better) four vessels on the surface they can be positioned by GPS [12]. Later, using the communication system, we can obtained (i.e., from the clock time of each communications between the elements) the distance between one or more vessels and the "constellation" whose position is known. Using the same mathematical calculation of the GPS system we can get the Absolute Localization of the whole swarm. The vessels on the surface (that can be substituted by little boats) could also have the advantage to carry a high band pass using laser communication system, owing to the easy transmission vertical channel; in fact the collimation problem (typical in laser communication systems) between vessels, in this case, can be partially avoided .

If we do not have surface vessels we can calculate the Relative Localization (i.e., the configuration of the swarm) solving the distance equations and using a constellation composed by some vessels; unfortunately we have multiple possible solutions and to select between them we need some more information. A flash, for example, can be used to discriminate direction from which the signal has arrived. A more clever system is to use a whatever (but known) movement of the swarm and to repeat the calculation for multiple solutions corresponding to the new possible configutation. Now, using the preceding configuration and appliyng a coordinate transformation (from the known movement), we obtain only one configuration matching both the old and the new. At this point we can obtain the configuration of these elements of the swarm taking one of them as the coordinate origin. Later, when the distances of other vessels are available, we can add more elements to build the whole swarm configuration.

VI. Equipment

The equipment to perform these solutions divide into "base requirements", which is the minimum instrumentation we anticipate having on the single machine and "desirable requirements" for enhanced instrumentation and better performance. Quantitative considerations are not considered here due to lack of space.

Each machine is characterized by six degrees of freedom, but only two of them (depth and heading) are very easy to measure, by means of a depthmeter and a compass. If the machine has cylindrical symmetry one is uninfluenced. Considering the yaw to be unimportant (we imagine navigation in one plane other than for a few moments) we understand that the real difficultly is to determine the coordinate x-y, of the center of mass; the x-y plane being that parallel to the sea bottom.

The base equipment of all the machines is composed of: Network communication, GPS, Depth meter, Inclinometer, Compass, Flash, Photodiode, Webcamera, Livery on the vessel surface and an electric or magnet device. All these components are cheap and available.

The network is a requirement that exists not only for communications but must also be used for data exchange and we are interested in its use in sonar ranging for RL. Of course a snapshot of any situation suffered in a delay, we presume between tenths of a second and one second, should be sent, together with the estimated robot speed for the correction. The network should be able to shift the working frequency from 100 to 1000 Khz (at least two frequencies). This number comes out from the consideration of data rate and the use of the net as an emergency ping or localization signal. The distance we want to cover (maximum 50 meters) and the data rate should be between 10 and 100 Kbytes/sec.

All the data available will be fused and weighted with all the data coming from other instruments so as to be more precise in localization. More than one algorithm is desirable, for example one which is more complex that uses all the available data and one which is quicker using only a subset of data, depending on the operative conditions and on the kind of localization required.

GPS is used for AL, when a scout (single robot that has this task) is on the surface.

At least a commercial depth meter must be present for AL in one dimension.

A couple of inclinometers is useful to measure the angle of position with respect of the land. Another degree of freedom can be removed by compass; care must be used in case of the presence of magnetic disturbances.

A photo-diode is used to receive flash lamp sequences for light communication; for example a codified sequence could send the heading of one machine to the other (close neighbors) so as to adapt themselves. Therefore, to transmit a simply codified message by flash, we can transmit a sequence; working on color or flash time is more complicated, standing the use of a cheaper flash unit. In some cases we can use an optical modem. This is a very cheap and light instrument, but does not give information on the position of the light source.

A webcamera must be used for image recording. Moreover together with an optical flash lamp. Using omnidirectional vision it is possible, with synchronized flash (or triggered by the first flash and using cumulative vision over a few seconds) to get qualitative information on the density of the machines. Of course it is not metric information but the single machine can get information if it is far or too far to the left (for example) from the swarm. Moreover flash could be useful in rescuing a single machine in difficulty, together with a switch of the network toward lower frequency working so as to increase the range.

The camera can also be used in the livery lecture for IRL for fast reaction movement, but it requires a computational job that must be simplified; image analysis is much too heavy. This last task could be done as a batch as for Simultaneous Localization and Mapping (SLAM), distributing the computational cost on a parallelized machine (the swarm itself if the network is adequate).

The single machine can be equipped with a strong electromagnet. The advantage of a static magnetic field is the possibility of transmiting (like a flash sequence) some information to the other machines. Moreover the hope is, contrary to the light source, to get some quantitative information on the RL by means of the magnetic field vector.

Electromagnetic transmission in sea water has made some progress over the last few years. Some opportunities are under investigation [13][14].

Magnetic (and electric) methods require some further consideration. An attempt to localize Rfid by magnetic field has been performed in air [13]. We start by considering the Earth's magnetic intensity field. Its scalar value is about 20 microTesla at the equator and 70 microTesla at the poles. We can consider it constant in our area of operation with some exceptions. We can generate a perturbation in magnetic Earth field to get information on the perturbation position (distance or heading direction or some other) from these numbers. We have calculated that cheap magnetometers are able to do this (also taking in account natural anomalies of the magnetic earth). So far, we should now be able to detect the spike in the magnetic field (we have calculated it in some conditions) superimposed on the Earth's field that we have generated in the sea. We get no information from the transient of the field but we do measure a change in the magnetic Earth fields. We now have two opportunities; one is a slow modulation of the field (to reduce attenuation) carrying some codified information, such as flash lamp. The second is to make an attempt to calculate the position of the field generator. This has been done for two objects in open space. It is a greater challenge to do this for a multisystem in the sea. We are investigating this possibility. Some other equipment such as Acoustic pinger, LBL USBL device, DVL, AHRS, Electromagnetic devices, Gradient localization can increase the performance of the system. A harbor could be equipped at not so high a cost, compared to the normal cost of surveillance. DVL can be mounted to integrate its data with cheaper AHRS (like XSense for example) for AL and RL (we avoid INS for the cost). A quantitative measure of absorption of electromagnetic waves in the water led to the possibility of using Radio frequency modems in sea water. Anyway some new electromagnetic underwater modems may promise something here and we are investigating. Recently, electromagnetic devices for marine application have shown an improvement in their performance leading to commercial products [15]. These new devices need to be investigated and also whether their dimensions are too large for Harness project. Electric and static magnetic fields also are under investigation.

VII. Conclusion and Future Work

In this work, we have explained that the Harness project could be useful for multipurpose use with particular attention to the localization problem. We propose a swarm of underwater cooperating robots.

The advantages lie in the economy of the method, the parallelization of the task and the robustness of the system. The disadvantage lies in the major control difficulty of the swarm, owing to the presence of a new layer named as "Swarm control" which has different rules from the individual machine control. Many difficulties remain to be studied, especially in the communication between swarm elements owing to the unfriendly environment that limits the communication channel also if different methods are used together.

The localization problem is divided into three different tasks related to different work conditions and some classical and alternative methods are under investigation. The configuration problem of the swarm is solved using the time clock of every message exchanged between the elements of the swarm and its known movement. Actual work is addressed to overcome the principal problems we encountered in the realization of the project that are communication, control, localization of the swarm within the sea and telepresence of the human operator.

Future work will concern the realization of a demonstration by using three prototypes.

VIII. Acknowledgments

The Harness project is funded by the Italian Institute Technology (IIT).

IX. References

[1] «HARNESS (Human telecontrolled Adaptive Robotic NEtwork of SensorS)». [Online]. Available: http://robotica.casaccia.enea.it/index.php?option=com_content&view=article&id=63&Itemid=82&lang=en. [retrieved: Lug, 2012].

[2] J. J. Leonard, A. A. Bennett, C. M. Smith, and H. Feder, «Autonomous underwater vehicle navigation», in *IEEE ICRA Workshop on Navigation of Outdoor Autonomous Vehicles*, 1998.

[3] S. Nawaz, M. Hussain, S. Watson, N. Trigoni, and P. N. Green, «An Underwater Robotic Network for Monitoring Nuclear Waste Storage Pools», *1st International ICST Conference on Sensor Systems and Software (SCUBE)*, pp. 236–255, 2009.

[4] J. Yuh, «Design and Control of Autonomous Underwater Robots: A Survey», *Autonomous robot*, pp. 7–24, 2000.

[5] J. Callmer, M. Skoglund, and F. Gustafsson, «Silent Localization of Underwater Sensors Using Magnetometers», *EURASIP Journal on Advances in Signal Processing*, vol. 2010, n°. 709318, pp. 1687–6172, 2010.

[6] M. Dunbabin, P. Corke, I. Vasilescu, and D. Rus, «Data muffling over underwater wireless sensor networks using an autonomous underwater vehicle», in *Proc. 2006 IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2006, pp. 2091–2098.

[7] O. Khatib, V. Kumar, and D. Rus, *Experimental Robotics: The 10th International Symposium on Experimental Robotics*. Springer Verlag, 2008.

[8] C. M. S. J. . Leonard and A. A. B. . Shaw, «Concurrent Mapping and Localization for Autonomous Underwater Vehicles», presented at the Proc. Int. Conf. Field and Service Robotics, 1997.

[9] K. S. Chong and L. Kleeman, «Accurate odometry and error modelling for a mobile robot», in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, 1997, vol. 4, pp. 2783–2788 vol. 4.

[10] «Archival Fish Tag with RF Transmitting Capabilities». [Online]. Available: http://www.tempsensornews.com/biomed/archival-fish-tag-with-rf-transmitting-capabilities/. [Accessed: Mar, 2011].

[11] «Desert Star Systems, LLC - Underwater and Defense Systems». [Online]. Available: http://www.desertstar.com/. [retrieved: Mar, 2011].

[12] E. W. Grafarend and J. Shan, «GPS solutions: closed forms, critical and special configurations of P4P», *GPS Solutions*, vol. 5, n°. 3, pp. 29–41, 2002.

[13] T. Nara, S. Suzuki, and S. Ando, «A Closed-Form Formula for Magnetic Dipole Localization by Measurement of Its Magnetic Field and Spatial Gradients», *IEEE Trans. Magn.*, vol. 42, n°. 10, pp. 3291–3293, Ott 2006.

[14] «Specifications «□SeaSPY «□Products «□Marine Magnetics

[Marine magnetometers that do their job smarter.]». [Online]. Available: http://www.marinemagnetics.com/products/seaspy/seaspy-specifications. [retrieved: Mar, 2011].

[15] V. I. Danilov and M. Ianovici, «Magnetic field of thick finite dc solenoids», *Nuclear Instruments and Methods*, vol. 94, n°. 3, pp. 541–550, 1971.

Self-Powered Wireless Ocean Monitoring Systems

Sea-Hee Hwangbo, Jun-Ho Jeon and Sung-Joon Park
Digital Communication Lab.
Department of Electronic Engineering, Gangneung-Wonju National University,
7 Jukheon-gil, Gangneung, Gangwon, Republic of Korea
Email: psj@ieee.org

***Abstract*—Recently, underwater wireless sensor network (UWSN) has been emerged as one of the important research topics from the need for the conservation and exploitation of the ocean. Since underwater sensor nodes suffer from limited power source, in this paper, we have planned to design and implement self-powered ocean monitoring systems which generate renewable marine energy from a device at sea surface and share the energy with underwater nodes. As a first step of this objective, we have developed an acoustic modem for wireless communication and made experiments in a pond to verify the performance of the modem. In addition, we have supplied the power to the modem by harvesting solar energy with solar panels. Efficient energy management, movement of underwater nodes and energy transfer will be investigated as succeeding work.**

***Keywords*-Underwater Wireless Sensor Network (UWSN); Energy Harvesting; Acoustic Modem.**

I. Introduction

During the last few years, there has been a growing interest in observing underwater environments for scientific exploration and monitoring ocean currents and winds (Tsunamis) and some researchers have studied and developed the prototype of underwater wireless sensor network (UWSN) systems [1], [2]. The study on acoustic modem has also conducted by many research groups. Especially, Yan et al. [3] implemented a modem adopting orthogonal frequency division multiplexing and verified one-way communication in oceanic environments.

One of the open problems for UWSN is the limited power, since the devices of UWSN typically rely on batteries. Until now, researchers tried to resolve this problem by utilizing a finite energy efficiently as much as possible [4], [5]. Meanwhile, in case of autonomous unmanned vehicle (AUV), there was a trial for the vehicle to be equipped with solar energy harvesting module at the Beacon Institute for Rivers and Estuaries [6]. When the vehicle floats to the sea surface, it starts to generate energy by using solar panels on its back.

Recently, initial studies on wireless energy sharing between devices are investigated in various forms. Magnetic resonant coupling with coils has lots of attention due to its efficiency and working range. Photonics and modern electro-magnetics group at MIT has conducted theoretical analysis and experiments with coils [7]. According to the results, wireless energy transfer in the air was possible between two self-resonant coils having the radius of 30 cm. The efficiency for 60 watts transfer was 70 % and 40 % at 1 m and 2 m distance, respectively. Besides, magnetic induction, RF and optics are the candidates of wireless energy transfer in air in the spotlight.

Based on the thorough survey of the abovementioned existing work, in this paper, we suggest the concept of self-powered wireless ocean monitoring system which generates solar or wind energy from a device at the sea surface and transfers this energy to other devices in a wireless manner. It would be possible to observe oceanographic data and predict natural disasters such as tsunami and sea shock permanently with this system. As a first step of the target system, we design and implement an acoustic communication system operated by the renewable solar energy and make experiments in a pond.

The remainder of this paper is organized as follows. We first describe the big picture of the target system in Section II. The design of solar harvesting and acoustic transceiver are discussed in detail in Section III. In order to examine solar harvesting modules and estimate the performance of the developed modem, experimental results are provided in Section IV. Finally, in Section V, we give a brief summary and comments on further work of this paper.

II. Objectives

Fig. 1 shows the blueprint of a self-powered wireless ocean monitoring system. This system is comprised of four components for monitoring of underwater environments: sensor node, sink node, gateway and an onshore user. The basic operation of the system is similar to the normal UWSN. That is, when the user requires underwater sensing data, it queries to a specific sensor node via the gateway in downlink transmission and gets the wanted data in uplink data transmission.

The main differentiated points of the proposed system are power management, energy harvesting, actuator module of node and wireless power transfer module. The power management part on the gateway controls energy generation and consumption continuously for smart energy management. Also, renewable energy is generated from the gateway located at the sea surface and the nodes goes to the gateway and recharge the battery when it is needed. Also, Fig. 2 illustrates the block diagrams of node and gateway in detail. In the figure, the black dotted lines represent the data flow and the red solid lines means the power transfer.

The specific procedures for energy sharing of the system are as follows. First, solar module on the gateway keeps generating energy from the sun and charging a battery. Meanwhile, in the underwater, each smart battery charge controller continuously checks the residual battery power in the node. When detected a lack of power of its node, the controller reports the present

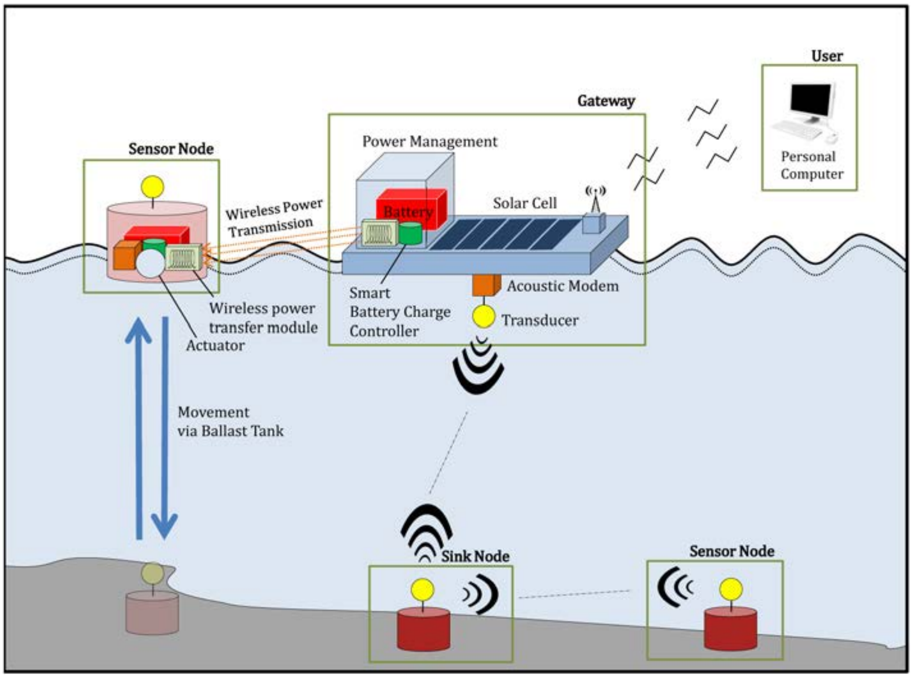


Fig. 1. Blueprint of self-powered wireless ocean monitoring system.

state to the gateway and sends a signal to a actuator module to move to the gateway. The actuator module consists of thruster and ballast which take in charge of the horizontal and vertical movements. After being arrived near the gateway, the battery in the node is charged wirelessly by two strongly coupled induction coils in wireless power transfer module. Since two resonant objects having the same magnetic resonant frequency have a tendency to exchange energy, these two coils make it possible to transfer a power. When the recharging is completed, the node goes back in underwater with the help of actuator module and starts acoustic communication again.

To realize this system, the following devices should be investigated.

1) Acoustic modem
2) Smart battery charge controller
3) Marine energy harvesting module
4) Actuators with thruster and ballast module
5) Wireless power transfer module

Among the five items, we have designed and implemented three devices: acoustic modem, smart battery charge controller and marine energy harvesting module. In further works, we will develop remaining two devices such as actuator and wireless power transfer.

III. A Prototype of Self-Powered Wireless Ocean Monitoring

A. *Transceiver*

In our previous work, several transceiver modules taking in charge of underwater wireless data transmission and reception have been implemented [8], [9]. Compared to the previous transceiver, amplification blocks in transmitter and receiver are reinforced and software coding is optimized for the performance enhancement in terms of working range and data rate. As shown in Fig. 3, the acoustic transceiver consists of digital board, analog transmission board, analog reception board and transducer. The detailed specification of the transceiver is summarized in Table I.

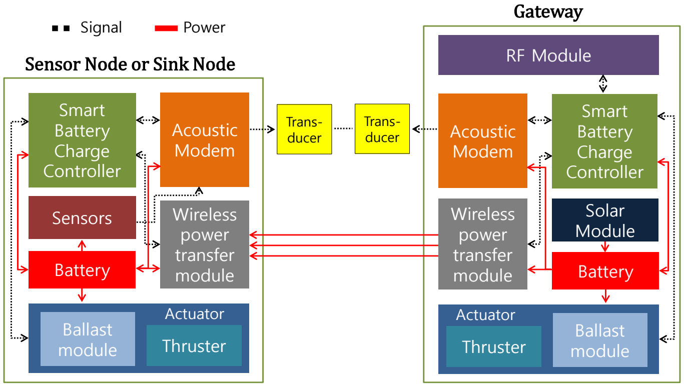


Fig. 2. Block diagram of gateway and node.

B. *Smart Battery Charge Controller*

As shown in Fig. 2, smart battery charge controller at gateway is connected to five components: RF module, acoustic modem, actuator, wireless power transfer module and battery. In this work, we have implemented controller communicating with RF module via RS-232, acoustic modem via SPI and battery.

The software processed at the controller of gateway is depicted in Fig. 4. MCU measures battery voltage level

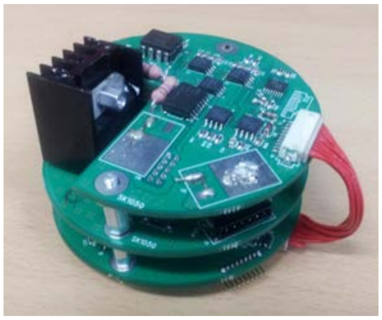


Fig. 3. Implemented acoustic transceiver.

TABLE I
SPECIFICATION OF ACOUSTIC TRANSCEIVER

Feature	Description
MCU	ARM9 (Cortex-M3)
Transducer	Omnidirectional
Resonant frequency	74 kHz
Interface	UART
Data rate	1 kbps
Power Consumption	3 W
Modem size	70 x 35 mm (ϕ×H)

periodically and compares it with a predetermined threshold T_h. If the voltage level is less than T_h, it computes sensing period to save power consumption and sends alarm command to the user. Then, the gateway requests data to nodes and receives sensing data from nodes. After forwarding the data to the user, it returns to the original state again.

C. Solar Energy Harvesting Module

The capacity of solar energy harvester should be determined by the following values:

1) the amount of power consumption of a device per day
2) the required amount of solar energy per day
3) the total required amount of solar energy per day considering a natural loss factor.

Since most of battery power is consumed at acoustic modem and actuator among several modules loaded in the sensor or sink node, the two modules are only considered for the capacity calculation of an energy harvester without loss of generality.

First, the amount of power consumption at acoustic modem per day is simply calculated by multiplying the power consumption of acoustic modem by hours in use per day. With the assumption that the acoustic modem operates during 20% in time in order to send data in underwater, the amount of power consumption per day becomes

$$3 \cdot (24 \cdot 0.2) \quad = \quad 14.4 \ WH. \qquad (1)$$

For actuator, if we assume that it consumes 60 watts and operates 5 minutes per day, the amount of power consumption at actuator is given by

$$60 \cdot \frac{5}{60} \quad = \quad 5.0 \ WH. \qquad (2)$$

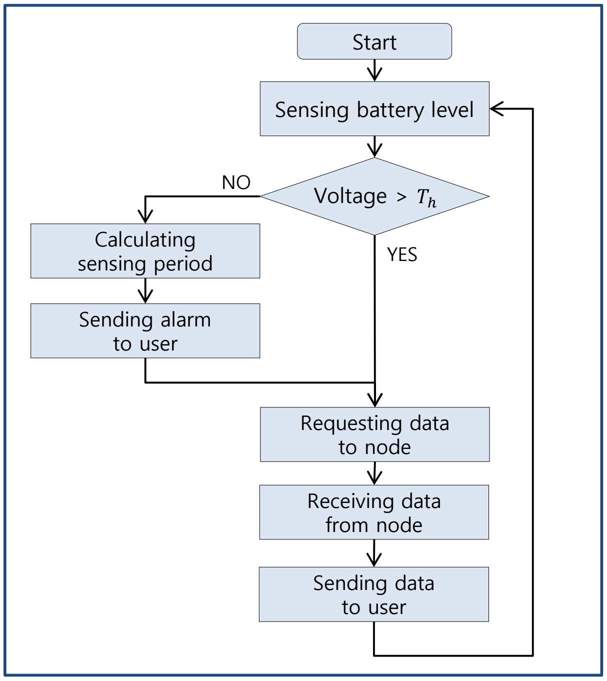


Fig. 4. Flow chart of software of smart battery charge controller at gateway.

TABLE II
THE AMOUNT OF POWER CONSUMPTION AND HARVEST

Comsumption		Harvest
Acoustic modem	Actuator	
3 W	60 W	20 W

From (1) and (2), total power consumption per day at each node is equal to 19.4 *WH*. Here, the result provided in [10] is used for the reference of power consumption at an actuator of wireless remotely-operated vehicle.

Since the hours exposed to sunshine is 3.5 hours in average, the required amount of solar energy per day can be predicted by dividing the amount of power consumption per day at a node by the sunshine hours, which is written by

$$19.4/3.5 \quad = \quad 5.5 \ W. \qquad (3)$$

Also, by considering the typical loss factor of 1.2, the total required amount of solar energy per day becomes

$$5.5 \cdot 1.2 \quad = \quad 6.6 \ W. \qquad (4)$$

In order to recharge the three nodes simultaneously in the worst scenario in Fig. 1, the solar panel should harvest at least 19.8 watts. From these reasons, solar energy harvesting module which generates 20 watts is selected for the implementation of the system. Table II shows the consuming power at a node and the harvested power at the gateway of this system.

IV. PRELIMINARY EXPERIMENTS

The developed acoustic modem, battery controller and solar energy harvester are verified the functionality in a pond. Fig. 5 illustrates the test scenario at the pond. As shown in the

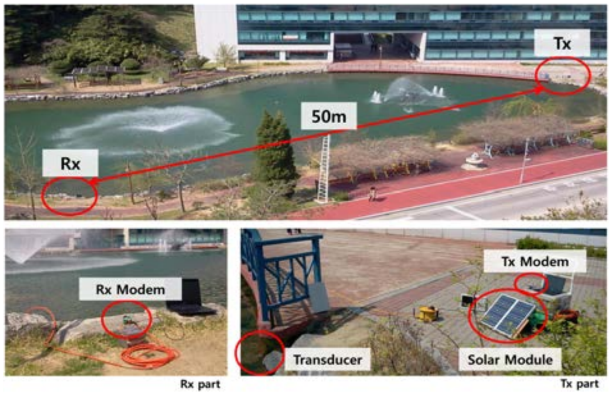


Fig. 5. Experimental environment.

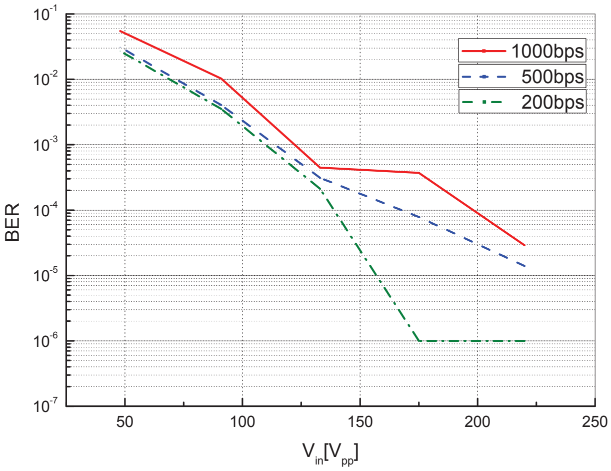


Fig. 6. Bit error rate with respect to input voltage and data rate.

figure, solar panel is connected to the transmitting modem and generates power. The receiving modem is placed on the other side at the distance of 50 meters from the transmitter. For the ease of experiments, only transducers are submerged to the pond. Predetermined data pattern is transmitted sequentially at the transmitter and the bit error rate (BER) is measured at the receiver.

Fig. 6 represents the bit error rate with respect to the input voltage of the transmitting transducer V_{in} and the data rate. According to the result, BER is improved by the increase of V_{in} at a given data rate. This is because if V_{in} is increased, the signal strength of received signal is also increased which make it easy for MCU to extract the transmitted signal from the corrupted signal by noise and multipath. Also, it is observed that BER degrades by increasing data rate which means both noise and multipath affect more and more for shorter symbol duration.

V. Conclusions and Further Work

In this work, we have proposed the concept of self-powered wireless ocean monitoring system which could be operated semi-permanently. Five core technologies have been distinguished for the target system. Furthermore, a simple underwater communication system equipped with solar energy harvester and battery controller has been developed and tested in a pond.

As succeeding work, we are planning to address the following topics. First, we will make more experiments in various marine environments to deploy acoustic modem with solar energy harvesting module. And, we will also develop wireless power transfer module and actuator and verify the feasibility of the self-powered wireless ocean monitoring systems.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-0004624).

References

[1] I. F. Akyildiz, D. Pompili and T. Melodia, "Underwater acoustic sensor networks: a survey revisited," *Ad Hoc Networks (Elsevier)*, 2007.

[2] Underwater acoustic network (UAN), "Underwater Acoustic Network," http://www.ua-net.eu/, [Jun. 2012].

[3] H. Yan, L. Wan, S. Zhou; Z. Shi, J.-H. Cui, J. Huang and H. Zhou, "DSP based receiver implementation for OFDM acoustic modems," *Elsevier Journal on Physical Communication,* 2011; doi:10.1016/j.phycom.2011.09.001.

[4] J.-H. Cui, J. Kong, M. Gerla and S. Zhou, "Challenges: building scalable mobile underwater wireless sensor networks for aquatic applications," *IEEE Network on Wireless Sensor Networking*, May 2006.

[5] J. Heidemann, W. Ye, J. Wills, A. Syed and Y. Li, "Research challenges and applications for underwater sensor networking," in *Proc. Wireless Communications and Networking Conference*, Las Vegas, NV, 2006, pp. 228-235.

[6] Beacon institute for Rivers and Estuaries, "About REON," http://www.bire.org/approach/reonoverview.php, [Jun. 2012].

[7] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher and M. Soljacic, "Wireless power transfer via strongly coupled magnetic resonances," *Science*, vol. 317, pp. 83-86, 2007.

[8] Jun-Ho Jeon and Sung-Joon Park, "A low-power underwater acoustic modem and its applications," in *Proc. SENSORCOMM 2011*, Nice, France, Aug. 2011.

[9] Tae-Hee Won and Sung-Joon Park, "Design and implementation of an omni-directional underwater acoustic micro-modem based on a low-power micro-controller unit," *SENSORS*, Feb. 2012.

[10] Geol-Ju Kim and Sung-Joon Park, "A wireless remotely operated vehicle using acoustic communication," *MTS Journal*, May/June 2012.

Monitoring the Marine Environment using a low-cost Colorimetric Optical Sensor

Brendan Heery, Lorna Fitzsimons, Timothy Sullivan, James Chapman, Fiona Regan, Kim Lau and Dermot Brabazon
Marine and Environmental Sensing Technology Hub (MESTECH), National Centre for Sensor Research
Dublin City University
Dublin, Ireland

Brendan.heery3@mail.dcu.ie
Lorna.fitzsimons@dcu.ie
Timothy.sullivan2@mail.dcu.ie
James.chapman@dcu.ie
Fiona.regan@dcu.ie
Kimlau01@gmail.com
Dermot.brabazon@dcu.ie

Jung-Ho Kim and Dermot Diamond
CLARITY, Sensor for Web Technologies
National Centre for Sensor Research
Dublin City University
Dublin, Ireland

Kimjungho@gmail.com
Dermot.diamond@dcu.ie

Abstract— **Anthropogenic activities have led to increased stress on our marine and other aquatic environments. There is a pressing need to monitor, measure, understand and mitigate the causes of these pressures. This paper presents the development and preliminary testing of a low-cost colorimetric optical sensor to detect colour-linked events in the marine environment. Potential applications may include the detection of Harmful Algal Blooms (HAB), which due to the production of toxins have deleterious effects on marine ecosystems and can ultimately lead to human, fish, bird and mammal deaths. Preliminary results indicate the capability of the sensor to differentiate between the colour signatures of several environmental samples.**

Keywords-environmental monitoring; colorimetric sensor; marine sensing; optical sensor.

I. INTRODUCTION

Recognition of the incontrovertible facts that the marine environment (1) plays a vital role in sustaining life and (2) is rapidly deteriorating, has led to a greater focus on the health of the European marine environment [1]. European frameworks such as the Water Framework Directive [2], the Bathing Water Directive [3] and the Marine Strategy Framework Directive [4] acknowledge that the marine environment, and water in general, are heritages and "*must be protected, preserved and, where practicable, restored with the ultimate aim of maintaining biodiversity and providing diverse and dynamic oceans and seas which are clean, healthy and productive* [4]".

Several water quality monitoring requirements are identified in the aforementioned directives and current FP7 projects address several of these areas of concern. For example, the focus of EPOCA [5] is ocean acidification (the drop in ocean pH caused by absorption of atmospheric CO_2). MedSeA [6] is again concerned with ocean acidification but specifically in relation to the Mediterranean Sea. The overall objectives of Hypox [7] and Viroclime [8] are to continuously monitor oxygen depletion in aquatic systems and to investigate the impact of climate change on viral flux (i.e., the growth, death and transport of viruses) respectively. Despite this significant increase in marine monitoring activity, there remain several important research areas that require detailed investigation, for example:

1. Continuous real-time monitoring of harmful bacterial species;
2. Improvement in continuous, long-term marine monitoring capabilities in terms of both sensor development and deployment e.g. antifouling and sensor marinisation (i.e., the design of a system for survival in the harsh marine environment).

To put some of these contamination risks in context, harmful bacterial species in the marine environment such as microcystins can lead to severe human and animal intoxication [9]. Symptoms can range for vomiting to liver and neurological dysfunctions which, in severe cases, can ultimately result in the death of consumers.

Certain harmful bacterial species of interest have specific colour footprints. For example, cyanobacteria, which include the microcystins, are characterised by bright green algal blooms [10]. Other harmful algal blooms, which contain species such as *Karenia brevis* and *Alexandrium* are characterised by the colour red, commonly known as red tides [11, 12]. Therefore, the application of low-cost colorimetric optical sensors may potentially be very beneficial to detect colorimetric events such as harmful algal blooms and to serve as an early warning alarm and decision support tool for more sophisticated bacterial sensors.

Traditional approaches to monitor and detect harmful bacterial species have typically involved the intermittent

ISBN: 978-1-61208-207-3

collection of samples at the relevant monitoring location (grab sampling), the transportation of the samples to the laboratory, the analysis of samples using various lab-based analytical techniques and the evaluation of results. This approach is not always ideal for a number of reasons, including: (1) the possibility of missing events due to low sampling frequencies, (2) the potential for sample contamination during transport from the point of collection to the laboratory, and (3) the inherent lead time from sample collection to analytical results. This can be problematic with regards to delayed reaction protocols and cause traceability. Lead times can be considerable in the analysis of bacterial samples which require culturing in the laboratory. There is therefore a strong argument for continuous in-situ monitoring. However, the development of long-term, continuous bacterial sensors in the marine environment is far from trivial. The objective of this research is to develop a low-cost, continuous, long-term, optical colorimetric sensor to detect colorimetric events, and thus act as a decision support tool for more sophisticated bacterial sensors, for example, to inform sampling frequency.

Much research has been conducted in the area of monitoring the inherent optical properties (IOP) of water and its relation to biological content. This is mainly done using satellite imaging of the ocean surface. Woods Hole Oceanographic Institution has reported vertical profiling of attenuation and scattering characteristics of New England coastal waters using a Wetlabs AC-9 meter. [13]

The Wetlabs AC-S *in-situ* spectrometer [14] a descendent of the AC-9 is a highly sensitive device with high spectral resolution for the detection of IOP, using a flow through detection system.

The OCS is an inexpensive, imprecise sensor intended for detecting significant changes in the bulk optical properties of marine waters. The OCS has several important advantages. First, it is low cost (less than €500 per unit for parts). A second advantage is ease of deployment; the sensor requires no support structure, only a mooring. These features allow for multiple strategic deployments to allow both temporal and spatial monitoring.

II. DEVELOPMENT OF OPTICAL COLORIMETRIC SENSOR

The Optical Colorimetric System (OCS) was based on a first generation prototype developed by researchers in CLARITY, Centre for Sensor Web Technology, Dublin City University (Fig. 1).

Prototype 1 was essentially a laboratory version and features included;

- LED array light source (IR, red, amber, green, blue)
- Photodiode detectors (90˚ and 180˚ to the light source)
- Short-range wireless communications.

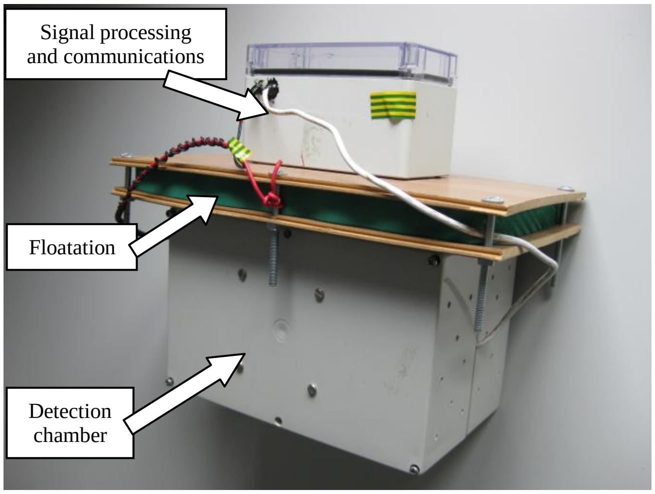


Figure 1. Prototype 1 showing measurement system below the water and communications above the water – separated by the float.

Prototype 2, the Optical Colorimetric Sensor (OCS), was developed to be a more robust field version of the original (Fig. 2). Additional features include:

- Robust marine deployable design
- Antifouling measures
- GSM communications.

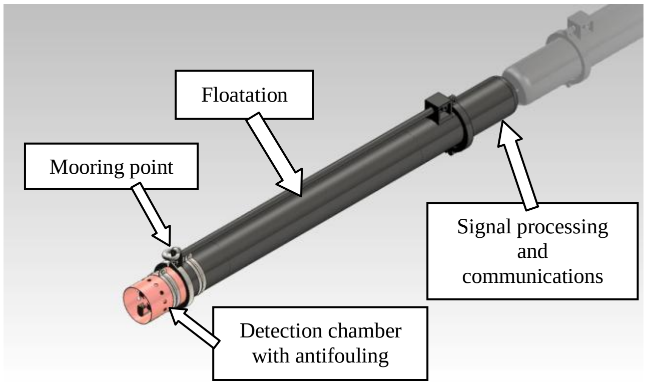

Floatation
Mooring point
Signal processing and communications
Detection chamber with antifouling

Figure 2. Prototype 2.

The Prototype 2 OCS is deployed using a single point mooring. It floats at the surface with the sensing elements submerged at 1 m. GSM telecommunications equipment is housed in the upper compartment above the water line. The OCS is designed for inshore use thus GSM communication is adequate in most Irish scenarios. Detection involves the use of 5 LEDs at differing wavelengths as a light source and photodiode detectors at 180 degrees and 90 degrees to the source. The detectors measure attenuation and reflectance respectively.

Optical detection in the marine is subject to interference such as scattering due to suspended particulate matter. Also significant drift in readings over time will be expected due to the formation of biofilms on optical windows. Antifouling research previously carried out by MESTECH has been incorporated into the sensor design to mitigate fouling

ISBN: 978-1-61208-207-3

effects. The objective of the OCS is to inform of shifts in colour or particle density with respect to a baseline, but not to quantify these changes accurately. Issues of sensor robustness will be addressed as part of future testing in the marine environment.

III. RESULTS TO DATE

Preliminary system evaluations were carried out using varying concentrations of food dye in tap water, which were chosen as being representative of typical environmental colorations. Fig. 3 and Fig. 4 show the system response (per LED) to changes in concentrations of red and green food dyes in tap water respectively. The percentage attenuation refers to the reduction in light transmission through the sample from a baseline of transmission in tap-water. Based on these initial results, the system demonstrates sensitivity to dye concentration changes in water, and can, therefore, distinguish between differing depths of colour. For example, the red dye (Ponceau 4R E124) absorbs light from 350 nm to 500 nm (i.e., UV to green). Thus in Fig. 3 the blue and green light is attenuated strongly by the red dye. In Fig. 4, the green dye (Tartrazine E102, Green S E142) absorbs light between 350 nm and 450 nm and also between 570 nm and 700 nm, thus the blue, amber and red lights are attenuated. In both Fig. 3 and Fig. 4, it can be seen that IR (850 nm) is not attenuated by either dye. However, it can be seen to attenuate strongly in environmental samples; see Fig. 5. This can be due to absorption in dissolved organic matter.

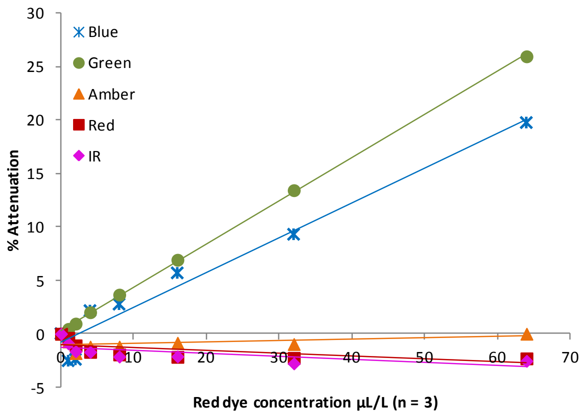


Figure 3. Response of the prototype 2 sensor to red dye.

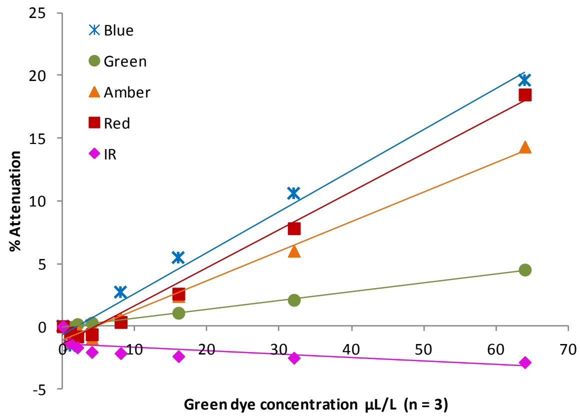


Figure 4. Response of the prototype 2 sensor to green dye.

A follow-on objective of the research was to establish the ability of the OCS to differentiate between the colours of samples taken from various aquatic environments. Samples from several locations throughout Dublin were analysed *in-situ* using the sensor. These samples included both riverine and estuarine waters. The percentage changes in light transmission through samples were recorded and were referenced against clean tap-water. The results are shown in Fig. 5, which clearly demonstrate the capability of the sensor to distinguish between the different absorbance signatures of the various water samples. These preliminary results are significant and point towards potential marine monitoring applications.

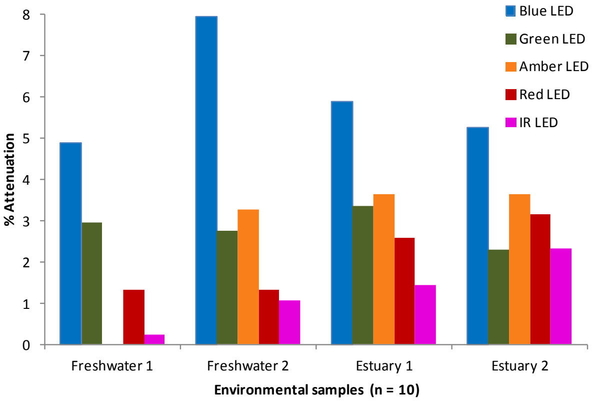


Figure 5. Field analysis of environmental samples using OCS.

IV. CONCLUSION AND FUTURE WORK

Based on preliminary tests and results, the OCS shows promise for marine environmental applications. The OCS system is robust and deployable in the aquatic environment. Of particular importance, the OCS shows potential to detect events in the environment such as a shift in water colour due to a pollution event.

ISBN: 978-1-61208-207-3

Future work includes longer-term deployment in the marine environment (SmartBay, the National Test and Demonstration Infrastructure, Galway). Events detected will be cross referenced against commercial sensing equipment (YSI sonde 6600) and standard laboratory analysis as part of an ongoing sampling programme. Future research will also involve the characterisation of the colorimetric response to simulated environmental events.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Marine and Environmental Sensing Technology Hub (MESTECH), which is funded under the Beaufort Marine Research Awards and carried out under the Sea Change Strategy and the Strategy for Science Technology and Innovation (2006-2013), with the support of the Marine Institute, funded under the Marine Research Sub-Programme of the National Development Plan 2007–2013.

In addition, the authors would like to acknowledge the system development work carried out by Prof. K.T. Lau (MESTECH) and Dr. J-H Kim in CLARITY, Centre for Sensor Web Technology, which is funded by Science Foundation Ireland under the CLARITY award (07/CE/ I1147).

The authors would also like to thank Mr Matthew Meagher for his assistance.

REFERENCES

[1] European Commission Fact sheet: protecting and conserving the marine environment, [on-line], http://ec.europa.eu/environment/pubs/pdf/factsheets/marine.pdf, (Accessed April 2012).

[2] European Water Framework Directive, DIRECTIVE 2000/60/EC.

[3] European Bathing Water Directive, DIRECTIVE 2006/7/EC.

[4] European Marine Strategy Framework Directive, DIRECTIVE 2008/56/EC.

[5] EPOCA website, [on-line], http://www.epoca-project.eu/, (Accessed April 2012).

[6] MedSeA website, [on-line], http://medsea-project.eu/, (Accessed April 2012).

[7] Hypox website, [on-line], http://www.hypox.net/, (Accessed April 2012).

[8] Viroclime website, [on-line], http://www.viroclime.org/, (Accessed April 2012).

[9] B. Byrne, E. Stack, N. Gilmartin, and R. O’Kennedy, 2012, “Antibody-based sensors: Principles, Problems and Potential for Detection of Pathogens and Associated Toxins”, Sensors, 9(6), 4407-4445.

[10] M.T. Madigan, J.M. Martinko, P.V. Dunlap, and D.P. Clark, *Brock Biology of Microorganisms*, 12th edition, 2009, Pearson, San Francisco, United States.

[11] J. T. Turner, V. Roncalli and P. Ciminiello, 2012, "Biogeographic Effects of the Gulf of Mexico Red Tide DinoflagellateKareniaBrevis on Mediterranean Copepods," Harmful Algae, 16(0) pp. 63-73.

[12] Woods Hole Oceanographic Institution website, [on-line], http://www.whoi.edu/oceanus/viewArticle.do?id=13406, (Accessed April 2012).

[13] Woods Hole Oceanographic Institution website, [on-line], http://www.whoi.edu/science/B/sosiklab/063.pdf, (Accessed, July, 2012)

[14] WetLabs AC-S specification sheet (Accessed July, 2012) http://www.wetlabs.com/products/pub/specsheets/acsssd.pdf

Sensor Based Risk Assessment for the Supply of Dangerous Products

Laurent Gomez
SAP Research France
Sophia Antipolis, France
Email:laurent.gomez@sap.com

Omar Gaci
ISEL
Le Havre, France
Email: omar.gaci@gmail.com

Jean Pierre Deutsch
LogPro Conseil
Paris, France
Email: jpdeutsch@logpro.fr

Elie El-Khoury
3Cap Technologies
Stuttgart, Germany
Email: elie.el-khoury@3cap.de

***Abstract*—As a consequence of globalisation, supply chain systems recently evolved toward a dynamic network of firms and industries. This complexification of supply chain processes raises several challenges in particular with respect to compliance to regulations (e.g., safety, security). With the multiplication of intermediate actors, a single non-compliant actor might jeopardize the safety of population and actors involved in supply chain process. There is, therefore, a clear need for risk management in order to mitigate the occurence of potential threats due to non compliance to regulation at the execution of the supply chain process. Traditionnal approaches tend to rely on human operators checks, or collection of contextual information from sensors locally (e.g., warehouse, truck). Therefore, there is risk of a disruption of regulation checks along the process execution. In this paper, we propose the delegation of risk assessment to sensor nodes, attached to the supplied products, for an automatic risk assessment all along the supply chain execution. Empowered with monitoring capabilities, sensor nodes are meant to trigger alert in case of contextual constraint violation along the supply chain. Our goal is to raise the awareness of the supply chain players with an early alerting service to enforce the regulations.**

***Index Terms*—Supply Chain Management, Sensor Networks, Security.**

I. Introduction

The globalization of trade has been accompanied by a growth of the number of intermediate partners involved in the supply chain. Those intermediate partners are mainly in charge of transportation or storage of products. As a consequence, the risk of disruption of the supply chain increases. For example, the supply of chemical substances raises risks of fire, explosion and environmental pollution. In order to mitigate those risks, and prevent any serious impact on population safety, safety regulations are in place, at international and national levels. Those risks might have a strong impact on population safety, or on the environment. Supply chain players have to be complaint with those regulations which impose them handling, transport and storage constraints on chemicals.

A. Disruption of compliance checks

There is, therefore, a clear need for compliance check at the execution of the supply chain. It is common practice in supply chain that each actor performs local regulation checks, in particular within storage unit, for example for SEVESO [1] classified sites. Measures for risk of fire, gas emission, detection of theft are already put in place locally. But, those measures only concern local checks since tracking and monitoring information are pushed to local supply chain systems only. We have therefore a disruption of regulation checks, as tracking and monitoring information are not forwarded to all the supply chain actors. For example, in the case of ambient temperature reaches chemical flash point during transportation, the storage unit receives the asset, without being informed of the inherent risk of explosion. Depending on the actors, those regulation checks, on temperature for example, are strongly dependent on the certification of the actors. For a site classified SEVESO II, like the K+N's [2] warehouse, strong measures on over heating, gas emission, or chemical leaking monitoring are put in place. But, this is not the case for all the actors involved in the supply chain. Therefore, we observe a disruption of the regulation checks, while each actor is focusing on its own compliance with regulations, with a different degree of implication, depending on their certification. To that extent, non compliance of a single actor might have a direct or indirect impact on the safety of the overall supply chain. For example, for a chemical, we might have a risk of explosion of a product which has been exposed to high temperature during transportation. When stored in the K+N's warehouse, the explosion risk remains for a while, until the temperature of the product goes back to normal. As a conclusion, it is difficult for supply chain management systems to evaluate continuously, without any disruption, the risk of incident occurrence.

B. Delegation of risk assessment to sensor nodes

Any disruption of regulation check might jeopardize the execution of the supply chain process. Therefore continuous risk assessment is critical for supply chain management systems. But, so far, risk assessment have been addressed only locally, within each unit of the supply chain. Meaning that non compliance of previous actor of the process can have a direct or non direct impact of the compliance with regulations. In order to cope with the disruption of risk assessment at the execution of the supply chain process, we propose to delegate risk assessment to sensor nodes attached to the products. Empowered with monitoring capabilities, wireless sensor nodes can evaluate continuously, and at runtime, the compliance with regulations. Sensor nodes are therefore capable of continuous evaluation of any mismatch between product's context and the constraints defined by regulations. To that extend, they support

ISBN: 978-1-61208-207-3

us with early detection of risks.

C. *Outline*

The remainder of this paper is organised as follows: in Section II, we describe a supply chain scenario motivating our approach. In Section III, we discuss related work with respect to regulation compliance at the execution of the supply chain. Section IV is dedicated to our approach: delegation of compliance checks to sensor nodes. In Section V, we evaluate our approach. Finally, we conclude in Section VI.

II. Importation of dangerous products from China to Europe

In order to illustrate our approach, we propose to use a supply chain scenario defined in the scope of the RESCUEIT [3] project. Related to the importation of dangerous products from China to Europe, this scenario has been elaborated and validated by end users such as Kuehne and Nagel (K+N) and the group Casino [4].

Chemicals are imported from a Chinese harbour toward the harbour of Le Havre, in France. Shipped products are household and gardening chemicals. These products are meant to be shipped by boat from a Chinese harbour. When received at the Le Havre harbour, the merchandise is checked by customs against REACH [5] regulations.

REACH is the European Community Regulation on chemicals and their safe use (EC 1907/2006) [5]. It deals with the registration, evaluation, authorisation and restriction of chemical substances. The aim of REACH is to provide an additional layer of protection for humans and the environment through the better and earlier identification of the intrinsic properties of chemical substances. To that extend, REACH introduces specific constraints on chemicals along the supply chain. They include the flash point, incompatibilities between products, and humidity conditions for chemicals.

At the Le Havre harbour, French customs with the support of an Approved Economic Operator [6] proceed to a merchandise integrity check. After a check of administrative document describing the content of the cargo, customs verify the quantity and quality of the products received.

Once quality checks have been performed at Le Havre harbour, and customs have verified that the merchandise is compliant with safety regulations, products are shipped by pickup trucks toward the warehouse located close to Savigny le Temple. This K+N warehouse,dedicated to the storage of dangerous products, is classified SEVESO II. This classification defines a set of safety management systems, emergency planning and land-use planning and a reinforcement of the provisions on inspections to be carried out by classified sites. In this case, specific safety measures are implemented on site, such as storage rules (e.g. limited quantity of chemical stored at the same place). Finally, household and gardening products are distributed to retailers (e.g., Casino supermarket).

A. *Identified products constraints*

In the scope of this scenario, we identify three gardening and household products ICPE-classified. ICPE [7] is a French nomenclature for ”Installation Classee pour la Protection de l'Environnement”. This classification defines a set a measures to be enforced for the handling, storage and transport of dangerous products. Each of the identified products has specific normative ICPE constraints: ICPE 1412, 1432, 1172.

Inflammable liquids are classified under ICPE 1412. In order to manipulate this product, gloves, glasses, a protective clothing, helmet and eye wash are mandatory. Products classified 1412 are a harmful and polluting products. Its flash point is 66 Celsius degrees. The flash point of a volatile liquid is the lowest temperature at which it can vaporize to form an ignitable mixture in air. In addition, this type of product must not be mixed with acids, bases or oxidizing. In addition, it is self flammable in large quantity at high temperatures. Therefore, in addition to risks of pollution along the supply chain, this product represents a significant risk of fire, if exposed to high temperature. In order to mitigate this risk, monitoring of ambient temperature is crucial.

ICPE 1432 products are liquefied gas inflammable. To manipulate this type of product, gloves, classes, protective clothing, wash eye, are mandatory. With respect to transport, they are classified UN 1950 or aerosol, with the mention of the restricted quantity, and a tag code 2.1-5F. Their flash point is between 13 and 13,4 Celsius degrees. It must not be in contact with acids and metal. Same as for ICPE 1412 products, in order to mitigate risk of fire, it is important to monitor ambient temperature.

Products classified as ICPE 1172 are dangerous for environment, extremely toxic for aquatic organisms. Gloves, classes, mask, and eye wash are required for the handling of Ronstar. Ronstar is classified UN 3007. In addition, this is irritant. Packaging of Ronstar is classified type III.

For transport, those dangerous products are classified UN 3082 [8], meaning dangerous products for the environment. UN code is four digit used for the transport of dangerous products. As this type of products is considered as slightly dangerous, packaging of type III is mandatory, with the mention of the restricted quantity. We therefore identified three additional constraints: shock, falling, opening. Shock and fall deal with any shock, falls occuring to the product, pallet or container, which might damage the product. Regarding opening, it refers to any attempt to product theft with the opening of container or packaging.

Table I summarizes identified constraints per ICPE classification. Those constraints are meant to be monitored by sensor nodes.

B. *Impacts*

In case of accidents along the supply chain, the impact on population safety, and on the environment can be disastrous. We identified three major impacts: fire, gas emission, dispersion of extinction waters.

Classification	Shock	Falling	Opening	Flash Point
ICPE 1412	X	X	X	13C
ICPE 1432	X	X	X	66C
ICPE 1172	X	X	X	-

TABLE I
IDENTIFIED CONSTRAINTS PER CLASSIFICATION

Depending on its intensity, fire can have more or less serious impact on individual health (e.g., slightly burning to death). In addition, merchandises and their packaging are combustible. They both have a strong calorific potential. In case of fire, the combustion of stored products would cause an important radiation of heating flux through the other storage areas in the warehouse. Toxic gases are also emitted in case of fire. Depending on the quantity of emitted gas, the effects on individuals can be lethal. In addition, under the effect of heat, dangerous products can cause the emission of toxic gas such as hydro-cyanic acid, oxides of sulphur. Fire fighters use specific products in order to extinguish fire. Those products (e.g., water plus chemical, powder, foam) contains chemical which aim at either decreasing the heat, or stifling the fire. Nevertheless those products drain polluting products which must not be thrown into the environment (e.g., river). Such incident may cause pollution of ground, underground or surface waters. It is therefore important to handle properly liquids used for fire extinstion in order to avoid them to be thrown outside of the building.

III. RELATED WORK

Automatic risk evaluation and propagation is currently an active topic in research institutions and industries alike. However, few works have been done over the automation of risk assessment for reasons related to human and technological limitations.

Aagedal et al. [9] introduced the CORAS project, which aims to provide methods and tools for precise, unambiguous, and efficient risk assessment of security critical systems. The focus of this project is on the tight integration of viewpoint-oriented modelling in the risk assessment process. Risk evaluation is done by determining the level of risk, categorize it, determine the interrelationships between risk themes, and prioritize the resulting risks themes. Although the CORAS addresses security-critical systems in general, it is interesting to note the risk assessment methodology used. In fact the authors use a model-based risk assessment.

Ivanov and Sokolov [10] focused their work on assessing and controlling the risks related to container supply chains (CSCs). However, due to the complexity of the risks in the chains, conventional quantitative risk assessment (QRA) methods may not be capable of providing sufficient safety management information, as achieving such a functionality requires enabling the possibility of conducting risk analysis in view of the challenges and uncertainties posed by the unavailability and incompleteness of historical failure data. Combining the fuzzy set theory (FST) and an evidential reasoning (ER) approach, the paper presents a subjective method to deal with the vulnerability-based risks, which are more ubiquitous and uncertain than the traditional hazard-based ones in the chains.

Wagner and Neshat [11] discussed the disruptions that occur more frequently and with more serious consequences. During and after supply chain disruptions, companies may lose revenue and incur high recovery costs. If the capability of measuring and managing supply chain vulnerability existed, they could reduce the number of disruptions and their impact. In this paper the authors developed an approach based on graph theory to quantify and therefore mitigate supply chain vulnerability. Although the approach seems promising, its applicability depends heavily on the availability of quantified data for the drivers of supply chain vulnerability. Without grounded data this method will not work. Additionally, the graph theoretical approach may not fully take into account the dynamic nature of supply chain vulnerability. Graphs are perhaps too static to be able to answer the dynamic changes in the supply chain at runtime.

IV. OUR APPROACH

A. Integration of Wireless Sensor Netwroks into Supply Chain

As depicted in Figure 2, whereas RFID are used for products tracking, sensor nodes can be used at different levels of the supply chain. Depending on the product value, sensor node can be used either at product level, packaging or pallet. In the scope of the RESCUEIT project [3], we have validated this assumption with end users of the project, K+N and the Casino Group. As we are addressing only low valuable products (e.g., household, gardening products), tagging RFID and sensor monitoring is done at pallet level.

Whereas RFID is rather focusing on identification of products (e.g., identification, classification), WSNs (Wireless Sensor Networks) are meant to monitor and control the supply chain environment. To some extent, RFID are not restricted to unique identification of products along the supply chain, but can be associated to information related to the classification, and dangerousness of products. Based on those classifications, and with regards to the regulations (e.g., safety, 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 violation of constraint is therefore reported to the supply chain management system as a risk of incident.

B. Terminology

Supply chain management systems are in charge of the delivery of products, or *assets*, to final customers. Depending on its classification, specific regulations define *constraints* along the supply chain, based on the *activity* on the assets (e.g., storage, transport, transformation). Therefore, regulations vary from one activity to another. A constraint on the stability of

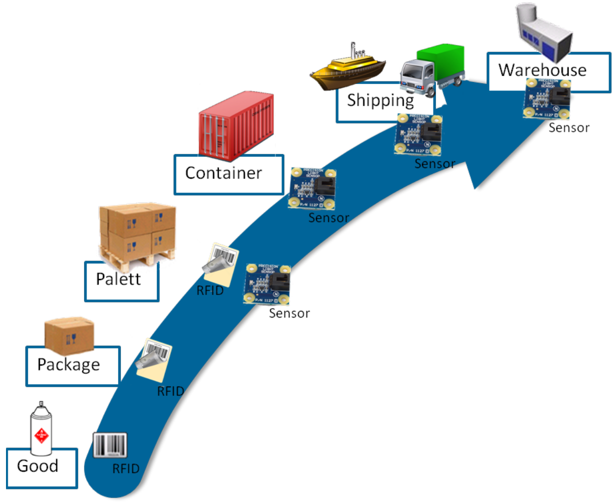


Fig. 2. Integration of RFID and WSN into Supply Chain

rack is only applicable in storage location for example. The non-compliance with regulations might lead to risky situation. For example, considering the flash point of product classified ICPE 1412, the following constraint is defined: *temperature must not exceed 13 Celsius degrees*. Whenever this constraint is fullfilled, then the risk of explosion increases. This example illustrates the relationship between the constraint and the *context*. *Context* is any kind of information which characterizes the environment of the asset. For the sake of clarity, we define the notation of *asset*, *classification*, *constraint*, *activity*, *context*, and *risk* as follows:

- *asset* is any product, or merchandise manipulated along the supply chain toward final customer. Assets are characterised by their classification.
- *constraint* is a representation of regulations over a specific asset, based on its classification.
- An *activity* is any steps in the supply chain, from production to delivery to final customer, including transformation, loading in palett, trucks, or rack.
- context is any type of information characterising the environment of an asset along the supply chain.
- *risk* is the probability that an incident occurs in the supply chain, due to a non compliance with regulations.

In Figure 1, we identify the following relationships: constraints, based on regulation, depend on asset classification and activity along the supply chain execution. For example, constraints on the stability of palett containing chemicals is to be fullfiled during storage activity. In addition, risk is depending on mismatch betwen constraints on products and their context in the supply chain. If a constraint on temperature is defined, the probability of incident occurence is depending on a violation with the context of the asset.

C. Methodology

As depicted in Figure 3, our approach is organised around the four following steps: (i) constraints extraction, (ii) node configuration, (iii) in-node risk evaluation, and (iv) node alerting. At (i) constraint extraction, constraints over product classification and supply chain activity are defined. For that purpose, regulations (e.g., safety, quality) are evaluated in order to extract per asset classification (e.g., chemicals, food), and per supply chain activity (e.g., transportation, storage) a set of constraints. For efficiency reasons, this task is meant to be performed outside of the node. At (ii) node configuration, the identified constraints are therefore pushed to sensor nodes attached to assets. Once pushed on nodes, those constraints are evaluated in real time by the sensor nodes during the (iii) in-node risk evaluation step. Whenever a sensor observes a mismatch between the current context and its set of constraints, it triggers an alert ((iv) node alerting).

D. Constraint definition

In Section II-A, we identify a set of constraints to be monitored depending on products' classification. Those constraints can be related to temperature, shock and container opening as depicted in Table I. While extracted from regulation, there are represented in an XML format. That XML representation is mapped product's classification, extracted from the regulations. In that context, we distinguish two types of constraints: monitoring and alerting ones. Monitoring constraint deals with regular monitoring of a given type of information, such as temperature.

```
<event>
  <name>Monitoring</name>
  <description>Monitoring Temperature every second</description>
  <monitoring>
  <sensorType>TEMPERATURE</sensorType>
  <sampleRate>00:00:01</sampleRate>
  </monitoring>
</event>
```

Alerting constraints define threshold over given sensor data type. Whenever that threshold is reached, an alert is triggered by the node. In addition, we define a notion of temporality on alert. An alert is triggered by the node only if the constraint is violated for a given time, for example light above threshold for 15 seconds in a raw.

```
<event>
  <name>CO</name>
  <description>Container opened.</description>
  <alert>
    <delayBetweenNotifications>00:00:30</delayBetweenNotifications>
    <constraintType>TEMPORAL</constraintType>
    <timePeriod>00:00:15</timePeriod>
    <expression>
      <constraint>
        <sensorType>LIGHT</sensorType>
        <compareOperator>GREATERTHAN</compareOperator>
        <value>1900</value>
      </constraint>
    </expression>
  </alert>
</event>
```

Finally, we enable combination of constraints. Combination of simple constraints enables an abstract of a constraint on the node. In the following example, we define a constraint, that if it is violated, trigger an alert for container overturn. This abstract constraint is based on acceleration monitoring.

```
<event>
  <name>PO</name>
  <description>Container has overturned.</description>
  <alert>
    <delayBetweenNotifications>00:00:30</delayBetweenNotifications>
    <constraintType>NONTEMPORAL</constraintType>
    <expression>
      <expression>
        <expression>
          <constraint>
            <sensorType>ACCELX</sensorType>
            <compareOperator>GREATERTHAN</compareOperator>
            <value>1000</value>
          </constraint>
        </expression>
        <binaryOperator>OR</binaryOperator>
        <expression>
```

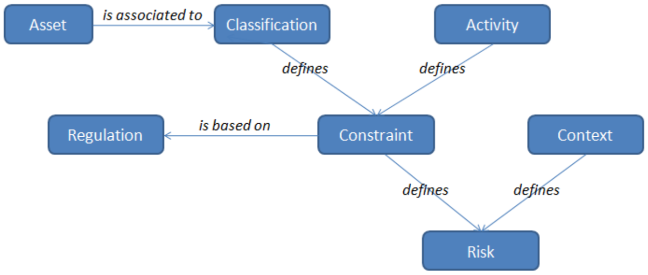


Fig. 1. Terminology

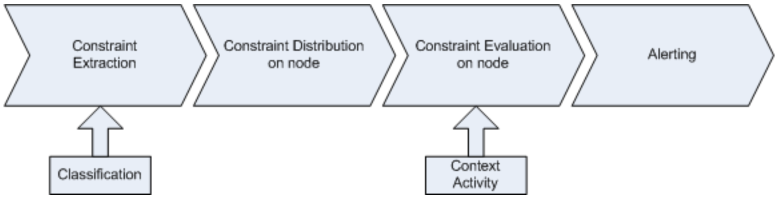

Constraint Extraction
Constraint Distribution on node
Constraint Evaluation on node
Alerting
Classification
Context Activity

Fig. 3. In-node risk evaluation process

```
          <constraint>
            <sensorType>ACCELX</sensorType>
            <compareOperator>LOWERTHAN</compareOperator>
            <value>-1000</value>
          </constraint>
        </expression>
      </expression>
  ...
  </alert>
</event>
```

E. Constraint evaluation

Sensor nodes have restricted resources available. XML processing is barely performed on such devices. For that reason, we need a specific representation of an constraint on the node. On the one hand, the representation must have a low memory cost and, on the other hand, its representation must be easily executable on an embedded device. A set of constraints is therefore represented on the node with:

- a set of simple constraint: Each constraint follows the template

  ```
  [SensorType]
  [Operator]
  [Value]
  [TypeofConstrainst]
  [<TemporalValue>]
  ```

  (e.g., A = ”Alert if Temperature GreaterThan 20”,Alert if Alert if Temperature GreaterThan 20”,Alert if Temperature GreaterThan 20”, B = ”Alert if Tilt LowerThan 50 for more than 5 seconds”).
- bytecode: that describe the execution of simple constraints.

The bytecode is inspired from the RPN (Reverted Polish Notation). Each operator follows their operands. (e.g., if A and B are simple constraint, the evaluation of ”A AND B” will be written as ”A B AND”). The interpretation is stack-based; that is, operands are pushed onto a stack, and when an operation is performed, its operands are popped from a stack and its result pushed back on. In the bytecode, we support three operands: AND, OR, and NOT.

On regular basis, the monitoring node collects all available ambient information (e.g., noise, temperature). It evaluates each simple constraint, and afterwards executes the loaded bytecode. If a violation in the combination of simple constraint is identified, an alert is sent to the SCM system.

F. Architecture

Figure 4 depicts our overall architecture. It is organised around three layers: supply chain management, mediation layer, and wireless sensor networks.

Supply chain management systems aim at monitoring assets along the execution of the supply chain. They have to be alerted in case of any incident which might disrupt the supply chain process. In our case, we use the container tracking system from SOGET [12].

As described previously, a WSN hosts a set of wireless nodes, attached to specific assets.

A mediation layer finally eases the integration of sensor nodes with supply chain. Within the mediation layer, we distinguish two services: the *sensor broker* and the *crossbow agent*. The sensor broker serves as a dispatcher for the subscription coming from the SCM system. The crossbow agent is in charge of the interface with the crossbow nodes [13] used for our evaluation.

As mediation layer, we use a mediation layer called the Middleware for Device Integration (MDI). MDI is a mediation layer developed by SAP Research for the integration of smart items (e.g., WSNs, RFID) into business applications. Based on an OSGi Service Platform, MDI is an agent-based middleware which enables both monitoring and controlling of smart items.

G. Message flow

In Figure 5, we depict the subscription to asset monitoring. Therefore, SCM systems have to subscribe to MDI for any

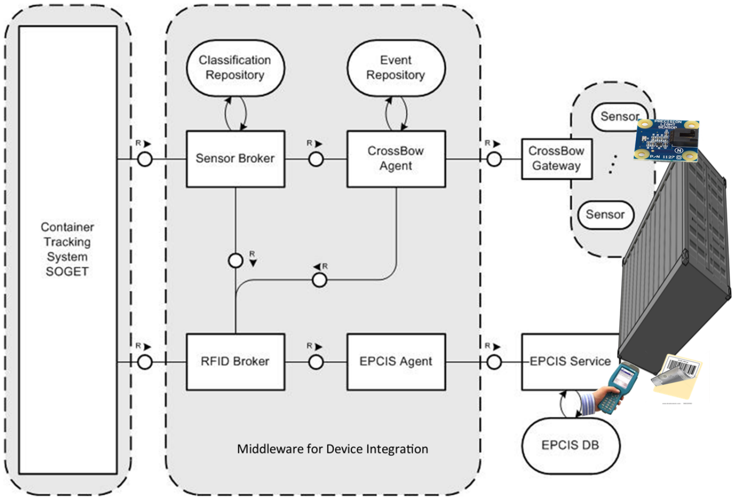


Fig. 4. Architecture

asset monitoring or occuring alerts (e.g., temperature exceeds flash point, shocks on the pallets).

As depicted in Figure 5, the SCM subscribes to monitoring or alerting for a given product, uniquely identified with its *productid*. In addition, the SCM provides the ICPE classification of the product to be monitored.

This classification is mapped to a set of contraints to be programmed on the nodes. This mapping is done at sensor broker level, with a dedicated database. The XML representation is pushed to the crossbow agent which generates a dedicated set of simple constraints and a bytecode to be executed on the crossbow node. The set of constraints and bytecode is specific to the type of sensor nodes used. Finally, the node is programmed over the air, with new constraints mapping monitored product's classification.

On regular basis, constraints and the bytecode are evaluated. If a constraint violation occurs, the sensor node triggers an alert. The MDI then notifies the subscriber.

V. Evaluation

In order to validate our approach, we propose an implementation of risk assessment on Crossbow sensor nodes [13]. Our goal is to evaluate the overhead on battery and memory introduced by our mechanism.

For the evaluation of our in node risk assessment approach, we used MICAz (MPR2400) processing unit equipped to a MTS310CAsensor board. Energy has been provided by two 1.2V rechargeable batteries with a capacity of 2200mAh. Each battery has been charged to a voltage of 2.65V before test start.

For the evaluation we propose four scenarios :

- Continuous packet sending every 30 seconds
- Monitoring of sensor data and continuous transmission every 30 seconds
- Monitoring of sensor data, evaluation of constraints violation and continuous transmission every 30 seconds
- Monitoring of sensor data, evaluation of constraints and Alerting only in case of constraint violation.

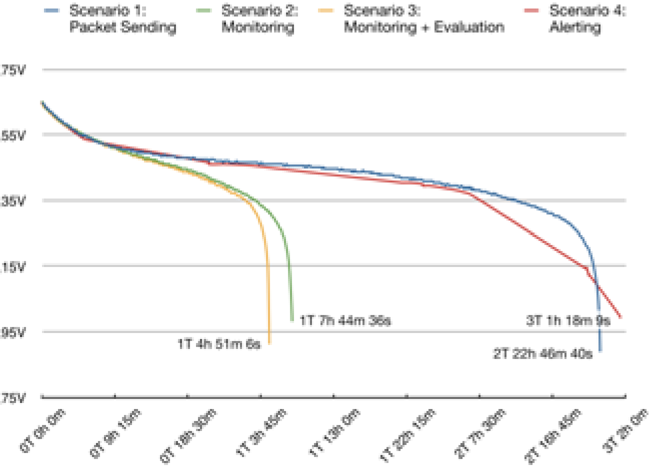


Fig. 6. Battery Evaluation

A. Battery overhead

Figure 6 depicts the consumption of energy for the four scenarios identified previously. We can observe three majors facts from that figure:

- Comparing Monitoring and Monitoring+Evaluation, we clearly demonstrated that the negligible overhead of evaluation of constraint violation.
- Comparing Packet Sending and Alerting, we observe that constraint evaluation do have a negligible overhead on energy consumption.
- Comparing Monitoring+Evaluation and Alerting, we confirm the fact that packet sending is main source of energy consumption. Following alerting strategies, we observe a gain in energy consumption of almost 60%.

B. Memory overhead

The sensors memory is limited. The used MICAz processing unit is equipped with an Atmel ATmega128L processor (8 bit

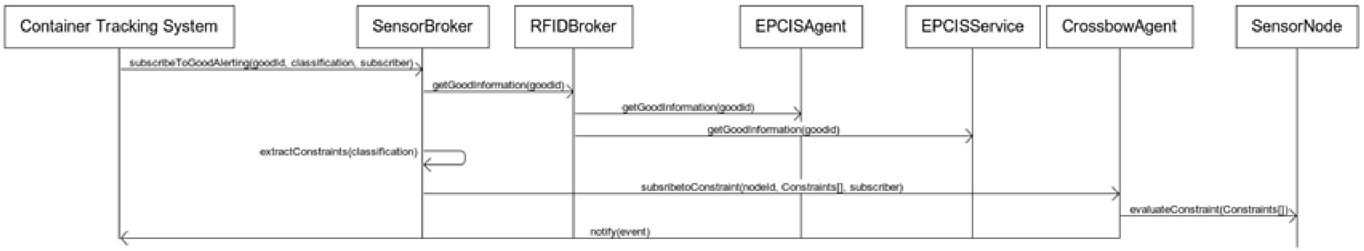

Container Tracking System
SensorBroker
RFIDBroker
EPCISAgent
EPCISService
CrossbowAgent
SensorNode
subscribeToGoodAlerting(goodId, classification, subscriber)
getGoodInformation(goodId)
getGoodInformation(goodId)
getGoodInformation(goodId)
extractConstraints(classification)
subsribetoConstraint(nodeId, Constraints[], subscriber)
evaluateConstraint(Constraints[])
notify(event)

Fig. 5. Message Flow

architecture), 128 K bytes of program is available in memory (ROM) and 4 K bytes for the runtime (RAM). The code in charge of constraint evaluation occupies 50.868 bytes in the ROM over 496.758 bytes available (10%), while using 3.737 bytes of RAM over 36.494 bytes available (10%). Those measurements are provided by the NesC, after source code compilation. Overall the memory consumption of our approach is relative limited.

VI. Conclusion and Outlook

With the shift towards a global open market in the recent decade, optimising production life-cycle of products is key to gaining the maximum profit. Ensuring safety of the products throughout the supply chain is a main aspect of the optimisation. The supply chain is typically made up of multiple companies who coordinate activities to set themselves apart from the competition.

Risk analysis techniques have emerged as a way to evaluate the potential risk inherent along the supply chain, and the identification of several different options in how to proceed. Often, these options are designed to minimize the risk while obtaining the most benefit, or at least finding ways to protect the product while taking the risk.

We discussed the current techniques for risk analysis. We show that current techniques lack the autonomy at execution time therefore, whenever an error occurred, a human intervention was always required to locate the problem and trigger a mitigation plan to prevent further propagation. we showed that this technique is not portable and scalable.

In this paper, we proposed to go a step further, with the delegation of risk assessment to sensor node attached to the supplied products. We identified a set of constraints mapped to the products classification. Those constraints represent the risk conditions of the item in question.

In addition, we propose the implementation and evaluation of our approach in the scope of the RESCUEIT [3] project. As future work we foresee the improving of the calculation of the risk value, to take into account the values from previous executions of the same supply chain. In addition, the issue related to the confidentiality of the generated alerts is still opened.

Acknowledgement

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 [3]. The authors take the responsibility for the content.

References

[1] ECD, “European council directive - control of major-accident hazards involving dangerous substances,” 1996. [Online]. Available: eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31996L0082

[2] Kuehne and Nagel, 2012. [Online]. Available: http://www.kn-portal.com/

[3] L. Gomez, M. Khalfaoui, E. El-Khoury, C. Ulmer, J. Deutsch, O. Chettouh, O. Gaci, H. Mathieu, E. El-Moustaine, M. Laurent, H. Schneider, C. Daras, and A. Schaad, “Rescueit : securisation de la chaine logistique orientee service depuis le monde des objets jusqu’a l’univers informatique,” *Workshop Interdisciplinaire sur la Securite Globale*, 2011.

[4] Casino, 2012. [Online]. Available: http://www.groupe-casino.fr/

[5] {European {Chemicals {Agency, “Guidance for identification and naming of substances under reach,” 2007.

[6] DGDDI, “Direction des douanes et droits indirects, approved economic operator,” 2005. [Online]. Available: http://www.douane.gouv.fr/page.asp?id=3421

[7] IPCE, “Installation classifiee pour la protection de l’environnement,” 2010. [Online]. Available: http://www.installationsclassees.developpement-durable.gouv.fr/

[8] UNECE, “United nations economic commission for europe, recommendations on the transport of dangerous goods - model regulations,” 2005. [Online]. Available: http://www.unece.org/trans/danger/publi/unrec/12e.html

[9] J. Aagedal, F. den Braber, T. Dimitrakos, B. Gran, D. Raptis, and K. Stolen, “Model-based risk assessment to improve enterprise security,” in *Enterprise Distributed Object Computing Conference, 2002. EDOC ’02. Proceedings. Sixth International*, 2002.

[10] D. Ivanov and B. Sokolov, “Handling uncertainty in supply chains,” in *Adaptive Supply Chain Management*. Springer London, 2010, pp. 81–91.

[11] S. M. Wagner and N. Neshat, “Assessing the vulnerability of supply chains using graph theory,” *International Journal of Production Economics*, vol. 126, no. 1, pp. 121–129, July 2010. [Online]. Available: http://dx.doi.org/10.1016/j.ijpe.2009.10.007

[12] SOGET, 2012. [Online]. Available: http://www.soget.fr/

[13] Crossbow, 2012. [Online]. Available: http://www.xbow.com/