International Journal on Advances in Networks and Services

Includes special issues on Wireless Sensor Networks and
a special issue on Peer-to-Peer Systems

2010 vol. 3 nr. 1&2

The *International Journal on Advances in Networks and Services* is published by IARIA.
ISSN: 1942-2644
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Networks and Services
Volume 3, Number 1 & 2, 2010

Networks and Services

- Claude Chaudet, ENST, France
- Michel Diaz, LAAS, France
- Geoffrey Fox, Indiana University, USA
- Francisco Javier Sanchez, Administrador de Infraestructuras Ferroviarias (ADIF), Spain
- Bernhard Neumair, University of Gottingen, Germany
- Gerard Parr, University of Ulster in Northern Ireland, UK
- Maurizio Pignolo, ITALTEL, Italy
- Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
- Feng Xia, Dalian University of Technology, China

Internet and Web Services

- Thomas Michael Bohnert, SAP Research, Switzerland
- Serge Chaumette, LaBRI, University Bordeaux 1, France
- Dickson K.W. Chiu, Dickson Computer Systems, Hong Kong
- Matthias Ehmann, University of Bayreuth, Germany
- Christian Emig, University of Karlsruhe, Germany
- Geoffrey Fox, Indiana University, USA
- Mario Freire, University of Beira Interior, Portugal
- Thomas Y Kwok, IBM T.J. Watson Research Center, USA
- Zoubir Mammeri, IRIT – Toulouse, France
- Bertrand Mathieu, Orange-ftgroup, France
- Mihhail Matskin, NTNU, Norway
- Guadalupe Ortiz Bellot, University of Extremadura Spain
- Dumitru Roman, STI, Austria
- Monika Solanki, Imperial College London, UK
- Vladimir Stantchev, Berlin Institute of Technology, Germany
- Pierre F. Tiako, Langston University, USA
- Weiliang Zhao, Macquarie University, Australia

Wireless and Mobile Communications

- Habib M. Ammari, Hofstra University - Hempstead, USA
- Thomas Michael Bohnert, SAP Research, Switzerland
- David Boyle, University of Limerick, Ireland
- Xiang Gui, Massey University-Palmerston North, New Zealand
- Qilian Liang, University of Texas at Arlington, USA
- Yves Louet, SUPELEC, France
- David Lozano, Telefonica Investigacion y Desarrollo (R&D), Spain
- D. Manivannan (Mani), University of Kentucky - Lexington, USA
- Jyrki Penttinen, Nokia Siemens Networks - Madrid, Spain / Helsinki University of Technology, Finland
- Radu Stoleru, Texas A&M University, USA

- Juong-Sik Lee, Nokia Research Center, USA
- Josef Noll, ConnectedLife@UNIK / UiO- Kjeller, Norway
- Willy Picard, The Poznan University of Economics, Poland
- Roman Y. Shtykh, Waseda University, Japan
- Weilian Su, Naval Postgraduate School - Monterey, USA

Multimedia

- Laszlo Boszormenyi, Klagenfurt University, Austria
- Dumitru Dan Burdescu, University of Craiova, Romania
- Noel Crespi, Institut TELECOM SudParis-Evry, France
- Mislav Grgic, University of Zagreb, Croatia
- Hermann Hellwagner, Klagenfurt University, Austria
- Polychronis Koutsakis, McMaster University, Canada
- Atsushi Koike, KDDI R&D Labs, Japan
- Chung-Sheng Li, IBM Thomas J. Watson Research Center, USA
- Parag S. Mogre, Technische Universitat Darmstadt, Germany
- Eric Pardede, La Trobe University, Australia
- Justin Zhan, Carnegie Mellon University, USA

Additional reviews

- Glen Sagers, Illinois State University, USA

International Journal on Advances in Networks and Services
Volume 3, Numbers 1 & 2, 2010

CONTENTS

Section from page 1 to page 195 is a special issue on Wireless Sensor Networks.
Section from page 196 to page 248 is a special issue on Peer-to-Peer Systems.

Introduction to Practical Deployments on Wireless Sensor Networks

Jaime Lloret
Department of Communications, Polytechnic University of Valencia
Camino Vera s/n, 46022, Valencia, Spain
jlloret@dcom.upv.es

***Abstract*—In the recent years there have been many published papers related on sensors. We can say that it is one of the hot topics of the last decades. But, the more we read, the more we realize that almost all are theoretical proposals or new theories. Very few works show real sensor networks deployments. While sensor network surveys show their benefits and the fields where they can be applied, they don't show who, how and where they have been deployed. In this paper, we are going to explain which motivation has roused us to create this special issue, and we will provide the main steps that a practical deployment should follow to be implemented. Then, we will introduce all the papers that have been selected to be published in this issue.**

Keywords-Wireless Sensor Networks, Practical Deployments, Real-world Implementations.

I. INTRODUCTION

Sensor has been a hot topic since the 90's [1]. The more parameters are needed to be measured from the environment, the more sensors have been deployed. Wireless Sensor Networks (WSNs) provide a self-configured and a self-powered wireless infrastructure for any type of environment. In order to build a WSN, the technology to join many of these sensors must be provided. In the late 90's, many theories and protocols were proposed for WSNs [2][3][4]. The main issues discussed in those works were the first three network layers. In the first layer, the physical layer, researchers proposed new wireless technologies for the sensors with the main objective of providing communication with low power consumption. In the second layer, where the medium access control protocols are placed, many researchers proposed new methods about how the wireless sensor devices have to see each other and establish a network topology [5][6]. In the third layer, many researchers focused their efforts on designing and developing new routing algorithms to route the information inside the wireless sensor network in an efficient manner and saving energy.

In all research fields related with WSNs, the main issues taken into account have been the low processing capacity, few computing resources and their energy limitation. Many more new theories appeared in the beginning of the 2000 [7][8]. Moreover, many researchers improved the WSN protocols and systems in existence. During the 2000, the technology advanced enough to create small devices with enough computing capacity and with low power consumption while being able to sense the environment [9][10]. We can see how WSN can benefit developing countries [11]. Moreover, the last researches have been focused on developing upper layer protocols for WSNs such as multimedia [12], transport layer protocols [13], etc.

Many articles show the wide range of application environments of the WSNs [14], but very few were about real implementations, compared to the amount of theoretic published works.

WSN lifecycle is formed by different differentiated steps. The first step is to know what the application of the WSN is. The second step is to design (or choose the appropriate) the hardware of the sensor node, its operative system, the system architecture, the wireless technology used by the nodes, the medium access protocol for all nodes and the routing protocol used in the sensor network. The third step is to implement the WSN in a testbench. WSN test is the fourth step. It should be performed in order to check its proper operation and its performance. The fifth step us the deployment of the WSN in the real world. Once, the WSN is running, its maintenance and continuous monitoring should be taken into account. Figure 1 shows the steps followed in the WSN. The peculiarity of the WSN lifecycle versus others is the number of disperses topics taken into account during the design step and the difficulty of the deployment step because of the number of devices and topics tackled.

In the beginning, there were some works published proposing algorithms and strategies for practical WSN deployment in terms of sensing coverage area [15][16], better detection and sensing range [17], the radio coverage [18][19], the optimal deployment based on the physical layer [20], the cost and lifetime [21], the topology control [22], energy saving issues [23], probability of target detection [24], and the scalability [25]. Later, some authors published a work with a theoretic analysis for the WSN deployment [26] and there was another work published about how to check the quality of the WSN deployments [27]. But, all of these papers are theoretic works.

We can also find in the literature some papers whose authors discuss the design and deployment issues [28], others that explain the WSN deployment procedure [29], and surveys about the steps that it should be made for a proper deployment [30]. Nowadays, there are studies about how the wireless technology affects to the deployment [31]. Others use the real world to test and validate their proposals [32], and even propose a deployment time validation consisting of techniques and procedures for WSN status assessment and verification [33]. Some authors state that the Self-Monitoring and Self-Configuration are the main keys for the WSN deployment success [34]. There is also a work that shows the development of robots to put the sensor nodes carefully in the appropriated place [35].

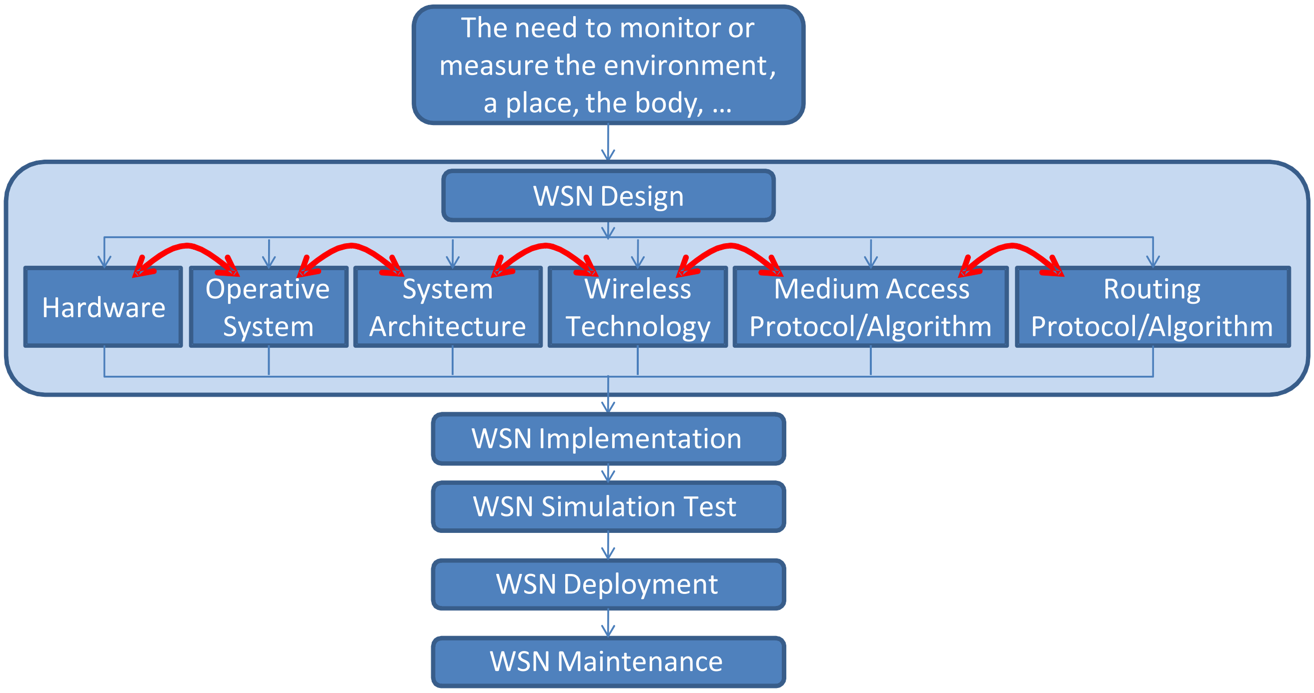


Figure 1. WSN lifecycle.

Finally, some software applications have been developed for designing WSNs, such as the one presented in [36], and for monitoring and control the WSN once it has been deployed [37][38].

The remainder of the paper is structured as follows. Section II shows some related work and explains our motivation to create this special issue. The papers selected for this special issue are described in Section III. Finally, Section IV provides the conclusions.

II. Related Work and Motivation

Along the years, there have been many conferences where one of the topics has been the deployment of the WSN. But, very few papers related with WSN deployments can be found between the papers published in their proceedings. Some examples of WSNs deployments found in some conferences are: for the agriculture [39], for a petroleum environment [40], for a Semiconductor Plant [41], for Coastal Marine Environments [42], for bridge monitoring [43], and for monitoring heritage buildings [44]. Although we can find one paper about the deployments performed by a research centre (in this case CSEM [45]), it is very rare to find such type of papers between the published papers. But, the Sensor-specialized conferences that have been differentiated because of the amount of papers published about practical deployments along the years, up to 2010, have been RealWSN (its first event is referenced in [46]) and, one of the biggest ones, SENSORCOMM (its first event is referenced in [47]), both having four past events. Others have disappeared or focused on more topics.

When we search for research books related with WSN deployments, we can see that there is no book focused exclusively on it, and, moreover, very few works about WSN deployments are published in books.

When we have a look at the research journals, we can find some sporadic paper publications. Some examples are: a deployment for structural monitoring [48], a deployment for management of irrigation [49] and a deployment of rural and forest fire detection system [50]. We have not seen any topic related with WSN deployments or real WSN experiences in any international journal. In the recent months, some call for papers for Special Issues of international journals have appeared with the topic WSN deployments such as the one called "Special Issue on Mobility in Wireless Sensor Networks" in [51], but it will be published in mid 2011. Moreover, we have found a Special Issue focused on with the main topic WSN Real-World Deployments and Deployment Experiences, but it will be published at the end of 2010. To the extent of our knowledge, there is not any Special Issue related with WSN deployments before the publication of this issue.

III. Papers Selected

This section describes the papers published in this special issue. We have selected 15th papers. They can be ordered in 4 parts. The first part shows WSN deployments in agricultural, rural and forest environments. Four papers have been selected to cover it. The second part shows four WSN deployments for healthcare and human assistance. The third part is formed by 2 papers about WSN deployments in oceanography and marine zones. The last part is formed by several types of WSN deployments. This part is ended by a survey of WSN deployments found in the related literature.

A. Part 1: Deployments in agricultural, rural and forest environments

In the first paper of this special issue, C. Anand et al. present "Wireless multi-sensor embedded system for Agro-industrial monitoring and control". The paper shows the development of a low cost multi sensor embedded system for measuring up to eight input analog sensor parameters along with reconfigurable automation, and communicating with host (wired and wireless), optimized by appropriate algorithms. The developed system has been deployed with resistance temperature detector (RTD) sensors, for temperature measurement, decision-making and their control in order to reduce the false alarms and unwanted process shut downs. Moreover, a mobile robotic platform is developed to test the multi-sensor embedded system for Agro-Industrial Applications (such as grain and fruit storage, vegetable storage smart pond automation for fresh aquaculture and so on). It has been also successfully tested to monitor the soil parameters in the farm and transmit the data to the central system using wireless protocol for precision agriculture. The data sent using wireless communication allows the system to be efficient and effective. It also allows making intelligent decisions based on the processed data. Furthermore, the average loss in signal is measured and received power is calculated and compared. The authors observed that for better transmission of signals via wireless communication, the low frequency along with low baud rate and line of sight range is required to minimize the signal loss.

The second paper, "Distributed Monitoring Systems for Agriculture based on Wireless Sensor Network Technology", authored by D. Di Palma et al., presents the design, optimization and development of an application to the agrofood chain monitoring and control. The authors try to reach the best architectural solutions with particular focus on hardware implementation and communication protocol design. The overall system was addressed in terms of the experienced platform, network issues related both to communication protocols between nodes and gateway operations up to the suitable remote user interface. They show the results given by several pilot sites in different vineyards throughout Italy and France. Finally, they present the commercial system "VineSense", based on WSN technology and oriented towards market and user applications. It was born from their previous experiments. At last, some important agronomic results achieved by the use of VineSense in different scenarios were sketched out, thus emphasizing the positive effects of the WSN technology in the agricultural environment.

The third paper is titled "Evaluation of Environmental Wireless Sensor Network - Case Foxhouse" and is authored by I. Hakala et al. This paper describes the implementation of an environmental monitoring system in a Foxhouse. They use a WSN in order to collect real-time data (luminosity, temperature and humidity) in hard outdoors conditions over a period of one year. In this paper they evaluate the communication over IEEE802.15.4, by providing and analyzing the throughput and the link quality statistics. They also present the power consumption measurements and discuss the observations performed.

The paper "Infrared wireless network sensors for imminent forest fire detection", authored by I. Bosch and L. Vergara, is the fourth paper. It presents an automatic forest fire surveillance ground system applied to early fire detection based on several sensors strategically located to render a required coverage. The images obtained from an infrared sensor network are processed by advanced thermal image processing techniques with the purpose of determining the presence of fire. The sensor wireless network supervises remotely the wide-forest area in order to detect immediately any fire threat. It provides total control of a tolerable level of false alarms and has maximum sensitivity to the presence of an uncontrolled fire for the defined false-alarm rate. The authors present some results obtained from a real environment in order to corroborate the control of the probability of false alarm and to evaluate the probability of detection dependence on signal to noise ratio. The delays of the system for alarm detection of controlled fire are also evaluated in order to show the performance of the system in the real environment.

B. Part 2: Deployments for healthcare and human assistance

P. Bustamante et al. present the fifth paper "A new Wireless Sensor for Intravenous Dripping Detection". This paper presents a WSN deployment for Intravenous Dripping System, which can detect when an intravenous liquid, provided to patients in hospitals, run out, as well as detecting obstructions in the catheter. The system allows more efficient and immediate attention in sanitary centers because the observation of the state of the container will not need human supervision. They paid attention to the reduction in consumption in comparison to other wireless devices with the same characteristics, the low cost of the wireless devices, the size of the devices, the flexibility forming the network, and the scalability of the system.

The sixth paper is authored by K. F. Navarro et al. with title "A Distributed Network Management Approach to WSN in Personal Healthcare Applications". It describes the development of a WSN for personal health monitoring system called Medical MoteCare. It uses a combination of medical and environmental sensors in an inherent network management distributed environment for the handling of medical data for patients. The use of SNMP and CodeBlue agents and a tailored MIB enhanced the scalability, modularity and flexibility of the system by potentially bringing the freedom of selecting from a vast range of existing SNMP based network management tools (such as the network management software Jaguar SX, iReasoning MIB Browser, and SysUpTime Network Monitor) to fit their specific WSN application requirements. Network management tools provide data storage correlation and dissemination as well as timely alerts when parameters are breached. They implemented JaguarSX in order to add intelligence to the system by utilizing network management correlation techniques that interpret the collected events automatically and react, or sometimes even anticipate from the collected statistical data, to harmful health conditions.

The paper "Design and Implementation of Multi-User Wireless Body Sensor Networks", by J. A. Afonso et al. is the seventh paper. It describes the design and development of two multi-user low power wireless body sensor network (BSN) that allow the real-time monitoring of wearable sensors data of several users with a single central monitoring unit (base station). They use inertial sensors that allow the monitoring of users' posture, goniometric development, movement as well as heart rate and respiratory rate sensors. Both BSNs present differences in the architecture, wireless network hardware and implemented protocols. They took into account the requirements of quality of service provisioning and low energy consumption. Their results show that their system provides good bandwidth efficiency and decreases the delivery error rate without significant increase in the energy consumption. The system can be used for the monitoring of teams of athletes in a gymnasium for sports with the goal of providing detailed information in order to enhance the performance of both the athletes individually and the team as a whole, and it can also be used for the medical field, namely in physiotherapy sessions, where such a system can benefit both the patient, by increasing his levels of confidence, and the therapist, by providing detailed information about the patient evolution.

In the eighth paper, "Human-Assisted Calibration of an Angulation based Location Indoor System with Preselection of Measurements", J. Kemper et al. present a software aided approach for the calibration of a triangulation-based indoor location system. The aim of their approach is to significantly reduce the calibration effort of the system by automating the process of node localization. The system estimates the localization of the sensor nodes using an algorithm based on the Newton-Raphson method for solving non-linear equation systems. It uses a passive sensor technology. The sensor nodes are equipped with infrared sensors only. As infrared radiation does not penetrate walls, the location system is limited to one room and the number of required sensors is small. Node localization is realized without any prior knowledge of sensor positions, orientations or the location of the moving person. The system has been implemented using a location system that exploits the thermal radiation of humans for localization. The algorithm works fine under the influence of noise. Moreover, an increased number of source locations improve the node localization accuracy.

C. Part 3: Deployments in oceanography and marine zones

O. Bondarenko et al. authored the ninth paper: "Deployment of Wireless Sensor Network to Study Oceanography of Coral Reefs". They deployed a WSN for in situ monitoring the Coral Sea and upwelled on the reef. The array of underwater sensors was deployed at various depths on the coral reef in Nelly Bay, Magnetic Island, Great Barrier Reef Australia (GBR). They used the temperature and 3D dense spatial data to correctly describe upwelling and their impact on plankton abundance (they collect the plankton data in real time synchronized to the temperature changes). The temperature data are sent in real time via the ad hoc network using RF signal to the on-shore base station. They also deployed dataloggers to collect temperature data from the same location. They used the WSN to demonstrate that short term stratification can occur in shallow tropical waters and influence the distribution of plankton.

The tenth paper is "Target Tracking in Marine Wireless Sensor Networks", by A. M. Mahdy and J. M. Groenke. They describe the main underwater tracking algorithms and thoroughly present a perspective on target tracking in marine WSNs. They also show the major challenges and applications about the deployment of underwater WSNs. Then, they present a two-layer broadband wireless infrastructure for marine/terrestrial sensor networks with military, inhabit monitoring, and homeland security applications. This paper has been included in this special issue because of the practical point of view provided in this work.

D. Part 4: Several types of deployments

The paper "An Integrating Platform for Environmental Monitoring in Museums Based on Wireless Sensor Networks", by L. M. Rodríguez and L. M. Pestana, is the eleventh paper. They present a WSN deployment for automatically and continuously monitoring the environment (temperature, humidity and light, throughout the day and night) of a Contemporary Art Museum located in Madeira Island, Portugal. They developed a new wireless sensor node that allow to automatically control real-time the dehumidifying, maintaining the humidity at more constant levels. They found some problems related with the signal propagation (because of the building walls where the WSN is being deployed), and with the hardware characteristics and resource limitations of the sensor nodes.

D. Bri et al. authored the twelfth paper: "A Wireless IP Multisensor Deployment". They present the deployment of a Wireless IP multisensor that uses IEEE 802.11b/g standard. It is able to gather several types of data from the environment and transmit the result of their combination. It is flexible and it could be adapted to any type of environment and to any type of physical sensor with a serial output. They took into consideration its development costs, its expansion capacity, the possibilities provided by the operating system, and its flexibility to add more features to the sensor node. Finally, they compared it with other existing sensors in the market.

The thirteenth paper is authored by F. Kerasiotis et al. In their work "Evaluation of Outdoor RSS-Based Tracking for WSNs Aiming at Topology Parameter Ranges Selection" they use the received signal strength (RSS), of the exchanged messages, for outdoor localization and tracking application under well defined topology constraints. They combine target tracking considerations by means of tracking techniques, topology parameters and factors influencing the tracking accuracy. Their study was focused on identifying the most crucial RSS-based tracking problems and to determine and evaluate the topology parameters that can guarantee successful tracking. The real outdoor tracking test demonstrates that the RSS can be used for outdoor localization and tracking applications under well-defined topology constraints and only after the proper calibration.

The paper "Deployment Considerations for Reliable Communication in Wireless Sensor Networks", authored by T. Stoyanova et al., is the fourteenth paper selected for this special issue. In this work, the authors study the deployment factors and requirements, which can ensure reliable communication links among the sensor network nodes. They used a RF signal propagation-based connectivity algorithm (RFCA), which utilizes an outdoor RF signal propagation model for predicting the RSS in the positions (based on the RF frequency, transmission power, transmitter–receiver distance, height from the ground and antenna's characteristics such as gain, polarization and orientation, etc.), where sensor nodes are supposed to be deployed. The RFCA is able to find the most appropriate, from the communication point of view, deployment parameters (height from the ground, T-R distance, and transmission power) for positioning the sensor nodes in outdoor environment in order to guarantee reliable connectivity level and minimize the interference from non-neighbor nodes. Finally, real outdoor measurements are compared with the simulation results for the verification of their RF propagation models.

Finally, the fifteenth paper selected has been "Practical Deployments of Wireless Sensor Networks: a Survey", by M. Garcia et al. The paper is focused on classifying applications of WSNs and describing real implementations. The goal of the authors is to complement the existing surveys, by presenting details of real implementations and practical deployments in order to understand how these networks run, and how they are designed, maintained and operated. Finally, the authors compare them in terms of network size, technology used, the communication type, single/multiple use and the domain where they are applied.

IV. CONCLUSION

In this special issue, we have seen a great variety of WSN deployments. We observed that the problems found in the real world could make the wireless sensor network deployment a difficult task. The WSN must be adapted to the environment in order to take the most advantage of them.

We hope the reader enjoys the papers published in this special issue and learns from the experiences provided by the authors in these papers.

ACKNOWLEDGMENT

The author wants to give special thanks to Petre Dini because he has believed in this special issue. We have tried to make closer the practical deployments of WSNs, and the main problems given by their real implementations, to the scientific community.

REFERENCES

[1] W. Göpel , J. Hesse, J. N. Zemel. Sensors: A Comprehensive Survey, pp. 1–7. New York: VCH. 1995.

[2] R. Min, T. Furrer, A. Chandrakasan, Dynamic voltage scaling techniques for distributed microsensor networks, Proceedings of ACM MobiCom'95, August 1995.

[3] P. Letteri, M.B. Srivastava, Adaptive frame length control for improving wireless link throughput, range and energy efficiency, Proceedings of IEEE INFOCOM'98, San Francisco, USA, March 1998, pp. 564–571.

[4] Wendi Heinzelman, Joanna Kulik, and Hari Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks", 5th ACM/IEEE Mobicom Conference, Seattle, WA, August 1999.

[5] Stephan Mank, Reinhardt Karnapke, Jorg Nolte, MAC Protocols for Wireless Sensor Networks: Tackling the Problem of Unidirectional Links. International Journal on Advances in Networks and Services. Vol 2, No 4, year 2009. Pp. 218 – 229.

[6] Jian Zhong, Peter Bertok. Self-organization supported algorithms for wireless sensor networks. International Journal on Advances in Security. Volume 2, Number 4, 2009. Pp. 298 – 311.

[7] J. N. AL-Karaki, A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey", IEEE Wireless Communications, Volume 11, Issue 6, Dec. 2004. Page(s):6 – 28.

[8] Kemal Akkaya, and Mohamed Younis. A survey on routing protocols for wireless sensor networks. Ad Hoc Networks. Volume 3, Issue 3, May 2005, Pages 325-349

[9] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. Computer Networks, 38(4):393–422, April 2002.

[10] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. Computer Networks. Volume 52, Issue 12, 22 August 2008, Pages 2292-2330.

[11] Al-Sakib Khan Pathan, Choong Seon Hong, Hyung-Woo Lee. Smartening the Environment using Wireless Sensor Networks in a Developing Country. Proceedings of 8th IEEE ICACT 2006, Volume I, February 20-22, Phoenix Park, Korea, 2006, pp. 705-709

[12] Ian F. Akyildiz, Tommaso Melodia, and Kaushik R. Chowdhurya, A survey on wireless multimedia sensor networks, Computer Networks. Volume 51, Issue 4, 14 March 2007, Pages 921-960.

[13] I. Khemapech, I. Duncan and A. Miller, A survey of wireless sensor networks technology, in Proceedings of the 6th Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking & Broadcasting, PGNET 2005. Liverpool, UK, 2005.

[14] Ning Xu, "A Survey of Sensor Network Applications,". University of. Southern. California. 2003. Available at. http://enl.usc.edu/~ningxu/papers/survey.pdf

[15] L. Filipe, M. Augusto, L. Ruiz, A. Alfredo, D. Ceclio, and A. Fernandes, "Efficient incremental sensor network deployment algorithm," in Proc. of Brazilian Symposium on Computer Networks 2004, Gramado/RS, Brazil, 2004, pp. 3 – 14.

[16] Wu, Chun-Hsien; Chung, Yeh-Ching. 2009. "A Polygon Model for Wireless Sensor Network Deployment with Directional Sensing Areas." Sensors 9, no. 12. Pp. 9998-10022.

[17] Jingbin Zhang, Ting Yan, Sang H. Son, "Deployment Strategies for Differentiated Detection in Wireless Sensor Networks," 3rd Annual IEEE International Conference on Sensor Mesh and Ad Hoc Communications and Networks (IEEE SECON'06), Reston, VA, 2006.

[18] A. S. Ibrahim, K. G. Seddik, and K. J. R. Liu, "Improving connectivity via relays deployment in wireless sensor networks," in Proc. IEEE Global Telecommun. Conf. (Globecom'07), pp. 1159-1163, Nov. 2007.

[19] You-Chiun Wang and Yu-Chee Tseng. Distributed Deployment Schemes for Mobile Wireless Sensor Networks to Ensure Multi-level Coverage. IEEE Transactions on Parallel and Distributed Systems. Vol. 19, No. 9, Sept. 2008, pp. 1280-94

[20] S. Toumpis and L. Tassiulas, "Optimal deployment of large wireless sensor networks," IEEE Transactions on Information Theory, vol. 52, no. 7, pp. 2935-2953, 2006.

[21] K. Xu, Q. Wang, H. Hassanein and G. Takahara, Optimal wireless sensor networks (WSNs) deployment: minimum cost with lifetime constraint, Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'05). Montreal, Canada. 24-22 Aug. 2005.

[22] Chun-Hsien Wu and Yeh-Ching Chung. Heterogeneous Wireless Sensor Network Deployment and Topology Control Based on Irregular Sensor Model, Lecture Notes in Computer Science, Vol. 4459/2007. Pp. 78-88. June 2007.

[23] Leila Ben Saad, Bernard Tourancheau, Towards an Optimal Positioning of Multiple Mobile Sinks in WSNs for Buildings. International Journal on Advances in Intelligent Systems, vol 2 no 4, 2009. Pp. 411-421.

[24] L. Lazos, Radha Poovendran, and James A. Ritcey, "On the Deployment of Heterogeneous Sensor Networks for Detection of Mobile Targets," 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '07), April 16-20, Limassol, Cyprus . 2007.

[25] M. Sheldon, Deji Chen, M. Nixon, A. K. Mok, A practical approach to deploy large scale wireless sensor networks. IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005. Washington, DC. 7 November 2005.

[26] Dario Pompili , Tommaso Melodia , Ian F. Akyildiz, Deployment analysis in underwater acoustic wireless sensor networks, Proceedings of the 1st ACM international workshop on Underwater networks, September 25-25, 2006, Los Angeles, CA, USA.

[27] E. Onur, C. Ersoy and H. Deliç, "Quality of Deployment in Surveillance Wireless Sensor Networks," International Journal of Wireless Information Networks, vol. 12, no. 1, pp. 61-67, Jan. 2005.

[28] Mohamed Youssef and Naser El-Sheimy. Wireless Sensor Network: Research vs. Reality Design and Deployment Issues. Fifth Annual Conference on Communication Networks and Services Research (CNSR'07). Fredericton, New Brunswick, Canada, May 14-17, 2007.

[29] Tzu-Che Huang, Hung-Ren Lai and Cheng-Hsien Ku, A deployment procedure for wireless sensor networks, The 2nd Workshop on Wireless, Ad Hoc, and Sensor Networks (WASN 2006). National Central University, Taiwan. August 10, 2006.

[30] Zoran Bojkovic , Bojan Bakmaz, A survey on wireless sensor networks deployment, WSEAS Transactions on Communications, v.7 n.12, p.1172-1181, December 2008.

[31] Bin Tong, Zi Li, Guiling Wang, and Wensheng Zhang, "How Wireless Power Charging Technology Affects Sensor Network Deployment and Routing," The 30th International Conference on Distributed Computing Systems (ICDCS), Genoa, Italy, June 21-25, 2010.

[32] Ashok-Kumar Chandra-Sekaran, Peter Schenkel, Christophe Kunze, Klaus D. Müller-Glaser. RealWorld Evaluation of a New Environment Adaptive Localization System. International Journal on Advances in Life Sciences. Vol. 1, Number 4, 2009. Pp. 143 - 157.

[33] H. Liu, Leo Selavo, John A. Stankovic: SeeDTV: deployment-time validation for wireless sensor networks. Fourth. Workshop on Embedded Networked Sensors (EmNets 2007). Cork, Ireland. June 25-26, 2007.

[34] Niclas Finne, Joakim Eriksson, Adam Dunkels, Thiemo Voigt, Experiences from Two Sensor Network Deployments - Self-Monitoring and Self-Configuration Keys to Success, Lecture Notes In Computer Science archive. Proceedings of the 6th international conference on Wired/wireless internet communications, Tampere, Finland. Pp 189-200. 2008

[35] Tsuyoshi Suzuki, Kuniaki Kawabata, Yasushi Hada and Yoshito Tobe. Deployment of Wireless Sensor Network using Mobile Robots to Construct an Intelligent Environment in a Multi-Robot Sensor Network. Boock Chapter "Advances in Service Robotics". Editor Ho Seok Ahn. InTech Education and Publishing, July 2008

[36] Antony Guinard, Alan McGibney, Dirk Pesch. A wireless sensor network design tool to support building energy management. Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings 2009, Berkeley, California, November 03, 2009. Pp. 25-30.

[37] José Pinto, Alexandre Sousa, Paulo Lebres, Gil Manuel Gonçalves, João Sousa, MonSense – Application for Deployment, Monitoring and Control of Wireless Sensor Networks. ACM Workshop on Real-World Wireless Sensor Networks REALWSN'06. June 19, 2006.

[38] Ioannis Chatzigiannakis, Georgios Mylonas, and Sotiris Nikoletseas. The Design of an Environment for Monitoring and Controlling Remote Sensor Networks. International Journal of Distributed Sensor Networks, 5: 262–282, 2009

[39] Prabhakar T V , N V Chalapathi Rao, Sujay M S, Jacques Panchard, H S Jamadagni, Andre Pittet. Sensor Network Deployment For Agronomical Data Gathering in Semi-Arid Regions. In Proceedings of the Second International Conference on COMmunication System softWAre and MiddlewaRE (COMSWARE 2007), January 7-12, 2007, Bangalore, India.

[40] Johnstone, I.; Nicholson, J.; Shehzad, B.; Slipp, J. Experiences from a wireless sensor network deployment in a petroleum environment. In Proceedings of the International Wireless Communications and Mobile Computing Conference 2007 (IWCMC'07), Honolulu, HI, USA, August 12–16, 2007; pp. 382–387.

[41] Lakshman Krishnamurthy, Robert Adler, Phil Buonadonna, Jasmeet Chhabra, Mick Flanigan, Nandakishore Kushalnagar, Lama Nachman, Mark Yarvis, Design and Deployment of Industrial Sensor Networks: Experiences from a Semiconductor Plant and the North Sea. Proceedings of the 3rd international conference on Embedded networked sensor systems SenSys'05, November 2–4, 2005, San Diego, California, USA.

[42] Umberto M. Cella, Nicholas Shuley and Ron Johnstone. Wireless sensor networks in coastal marine environments: a study case outcome. Proceedings of the Fourth ACM International Workshop on UnderWater Networks (WUWNet'09). Berkeley, California. 2009

[43] Michael V. Gangone, Matthew J. Whelan, Kerop D. Janoyan, Ratneshwar Jha, Field deployment of a dense wireless sensor network for condition assessment of a bridge superstructure, SPIE Smart Structures Symposium, San Diego, California. 2008

[44] Matteo Ceriotti, Luca Mottolal, Gian Pietro Picco, Amy L. Murphy, ¸ Stefan Guna, Michele Corrà, Matteo Pozzi, Daniele Zonta, Paolo Zanon. Monitoring Heritage Buildings with Wireless Sensor Networks: The Torre Aquila Deployment. 8th Int. Conf. on Information Processing in Sensor Networks (IPSN), 13-16 April 2009, San Francisco, California, USA. Pp. 277-288.

[45] J. Rousselot, Ph. Dallemagne, J.-D. Decotignie, Deployments of Wireless Sensor Networks performed by CSEM, Proceedings of the 1st conference on COGnitive systems with Interactive Sensors (COGIS'09), Paris, 16-18 November 2009.

[46] Workshop on Real-World Wireless Sensor Networks (REALWSN'05). Available at http://www.sics.se/realwsn05/

[47] International Conference on Sensor Technologies and Applications (SENSORCOMM 2007). At http://www.iaria.org/conferences2007/SENSORCOMM07.html

[48] Shamim N. Pakzad, Gregory L. Fenves, Sukun Kim, and David E. Culler, Design and Implementation of Scalable Wireless Sensor Network for Structural Monitoring. Journal of Infrastructure Systems. Volume 14, Issue 1, MARCH 2008.Pp 89-101.

[49] Lea-Cox, J. D., A. G. Ristvey, F. R. Arguedas, D. S. Ross and G. F. Kantor. 2008. Wireless Sensor Networks for Real-Time Management of Irrigation and Nutrient Applications in the Nursery and Greenhouse Industry. HortScience, Vol.43, July 2008.

[50] Lloret, Jaime; Garcia, Miguel; Bri, Diana; Sendra, Sandra. 2009. "A Wireless Sensor Network Deployment for Rural and Forest Fire Detection and Verification." Sensors 9, no. 11: 8722-8747.

[51] Special Issue on Mobility in Wireless Sensor Networks (Guest editors: Damianos Gavalas, Grammati Pantziou, Charalampos Konstantopoulos), The Computer Journal, At: http://www2.aegean.gr/dgavalas/ComputerJ_SI.pdf [Last access: July 2010]

[52] Special Issue on Wireless Sensor Networks: Designing for Real-World Deployment and Deployment Experiences (Guest editors: Elena Gaura, Utz Roedig and James Brusey), Measurement Science and Technology Journal (Institute of Physics). At: http://www.coventry.ac.uk/researchnet/external/content/1/c4/24/92/v1258035327/user/MST_WSN_special_issue.pdf [Last access: July 2010].

Wireless multi-sensor embedded system for Agro-industrial monitoring and control

Chandani Anand, Shashikant Sadistap, Satish Bindal, B. A. Botre and KSN Rao
Central Electronics Engineering Research Institute (CEERI) Pilani-333031, India
Council of Scientific and Industrial Research (CSIR)
e-mail: Chandnianand_ddu@yahoo.com, ssadistap@yahoo.co.in, bindalonline@yahoo.com, bbotre@gmail.com, ksn@ceeri.ernet.in

Abstract— **The paper presents a development of the multi sensor embedded system for measuring up to eight sensor parameters optimized by appropriate algorithms. Since most of the industrial applications use the analog sensors with transmitters for sensing the process parameters particularly in harsh environment because of their strong mechanical packaging and ruggedness. Then the benefits of digital technology in the vast world of analog sensors can be implemented by development of appropriate multi sensor embedded system. The developed system has most of the features of smart sensing and communicating to other embedded/Host PC through a wireless interface. The developed prototype system has been tested with RTD temperature sensors for temperature measurement over a wireless connectivity for various distances. From an application point of view, a mobile robotic platform is developed to test the multi-sensor embedded system for Agro-Industrial Applications. Further, the average loss in signal is measured and received power is calculated and compared. Finally, the effect of obstacles at indoor and outdoor range for wireless transmission has been presented. It is observed that for better transmission of signals via wireless communication, the low frequency along with low baud rate and line of sight range is required to minimize the signal loss.**

Keywords- Embedded System; Sensors and Actuators; Wireless Mobile Robots; Zigbee Wireless Connectivity

I. INTRODUCTION

A "smart sensor" is a transducer (or actuator) that provides functions beyond what is necessary to generate a correct representation of a sensed or controlled quantity. The "smart sensor" functionality will typically simplify the integration of the transducer into applications in a networked environment [1]. Embedded with a microcontroller unit, a smart sensor has much more built in intelligence over a traditional sensor. So, it can perform more powerful functions such as self-identification, self–calibration, converting the raw sensor signal into a digital form . Multi sensor systems are now used in many application areas, including environment and habitat monitoring, healthcare, home automation, and traffic control especially in agro and food applications like Reverse Osmosis (RO) plant automation, fish pond management systems, environmental monitoring, precision agriculture, machine and process control, building and facility automation. Wireless based Smart sensor module has drawn attention of the industry on account of low cost, low power consumption, flexibility, and use in remote areas.

Precision farming is the conjunction of a new management perspective with the new and emerging information and communication technologies leading to higher yields and lower costs in the running of large scale commercial agricultural fields. Further, the embedded sensing technologies allow the identification of pests in the crops, increased moisture or drought at a real-time interval with automated actuation devices to control irrigation, fertigation and pest control in order to offset the adverse conditions. The Precision farming system incorporates: a) Sensing agricultural parameters b) Identification of sensing location and data gathering c) Transferring data from crop field to control station for decision making and d) Actuation and Control decision based on sensed data.

Over the last few years, the advancement in sensing, embedded technology and wireless communication technologies has significantly brought down the cost of deployment and running of a feasible precision agriculture framework. Emerging wireless technologies with low power needs and low data rate capabilities, which perfectly suites precision agriculture, have been developed [2, 3]. Agricultural Sensors, positioning systems for detecting location of sensors, actuators like sprinklers, foggers, valve-controlled irrigation system, etc. are also reported in the literature. However, very limited work has been done so far on the wireless multi-sensing system to be used to transfer sensor data wirelessly from crop field to the remote central PC.

In this paper development of a low cost multi sensor system for sensing eight input analog sensors along with reconfigurable automation, and communicating with host (wired and wireless) is presented. The proposed smart multi sensor system is an attempt to develop a generic platform with 'plug-and-play' capability to support hardware interface, communication, needs of multiple sensors, and actuators. The system has been implemented using our own developed eight potentiometer based kit and tested with resistance temperature detector (RTD) sensors for sensing, decision-making and their control in order to reduce the false alarms and unwanted process shut downs. The data sent using wireless communication makes system efficient, effective and intelligent decisions based on processed data that is accurate and analyzed. Wireless communication has its own advantages like Ease of installation; no bulky cables are needed and simplify design of systems. In this paper, we introduce the new smart sensor device with processing and communication as well as various sensing abilities for industrial and agro – based monitoring applications.

Further, considering the potential application of this field, the present extended work details on the development of wireless mobile robotic platform for farm sensing is also presented [4]. The organization of this paper is as follows. Section 2 covers related work on wireless communication based smart sensors related to agro, food and other applications. Section 3 gives the system description. Finally, Section 4 covers experimental results of the current implementation and Section 5 includes conclusion and future work.

II. Related Work

There are many projects undertaken and many researches proposed the development of smart sensors and their networking. The design and implementation of home network based on smart sensor devices focuses on the development of system for home purposes where it can be implemented for door lock system, gas detecting system, controls TV, refrigerator, and outlets, and applying RFID into home networks [5]. The development of smart sensors for pollutant gas detection like CO and LPG where a CO gas detector is developed and controlled by smart sensor to give alarm if gas generation is above the set limit and used in home, institutions and industries [6, 7]. The Motes and Smart Dust project at UC, Berkeley focused on creating low-cost micro-sensors, with emphasis on the development of sensors and an embedded operating system, Tiny OS . Recently, the wireless sensor networking is used for life science automation where they propose a wireless sensor network (WSN) especially dedicated to the field of life science automation (LSA), which can be used for multiple purposes [8]. This area provides a bulk of process and environmental parameters, which have to be controlled. Passel and Danzer developed a portable, mobile instrument to measure temperature, relative humidity, noise, brightness and ammonia content in the air within the house and transferred the data wirelessly to a PC through an infrared data link [9, 10, 11].

In precision agriculture, the most important step is the generation of maps of the soil with its characteristics. These included grid soil sampling, yield monitoring and crop scouting. RS (remote sensing) coupled with GPS, coordinates and produce accurate maps and models of the agricultural fields. The sampling was typically through electronic sensors such as soil probes and remote optical scanners from satellites. The collection of such data in the form of electronic computer databases gave birth to the GIS. Statistical analyses were then conducted on the data and the variability of agricultural land with respect to its properties was charted. The technologies used are expensive like satellite sensing and was labor intensive where the maps charting the agricultural fields were mostly manually done. Blackmore et al., in 1994 [12] reported a comprehensive system designed to optimize agricultural production by carefully tailoring soil and crop management to correspond to the unique condition found in each field while maintaining environmental quality. Further, only large farms could afford the high initial investment in the form of electronic equipment for sensing and communicating. The technologies proposed at this point comprised of three aspects: (a) Remote Sensing (RS), (b) Geosynchronous Positioning System (GPS) and (c) Geographical Information System (GIS).

The system we have developed is useful because various physical parameters can be sensed and manually controlled on automatic false detection, as well as eight parameters can be manage precisely via single micro controller. Finding the better location and minimum loss in signal for wireless communication to device, makes the system more challenging and applicable [13].

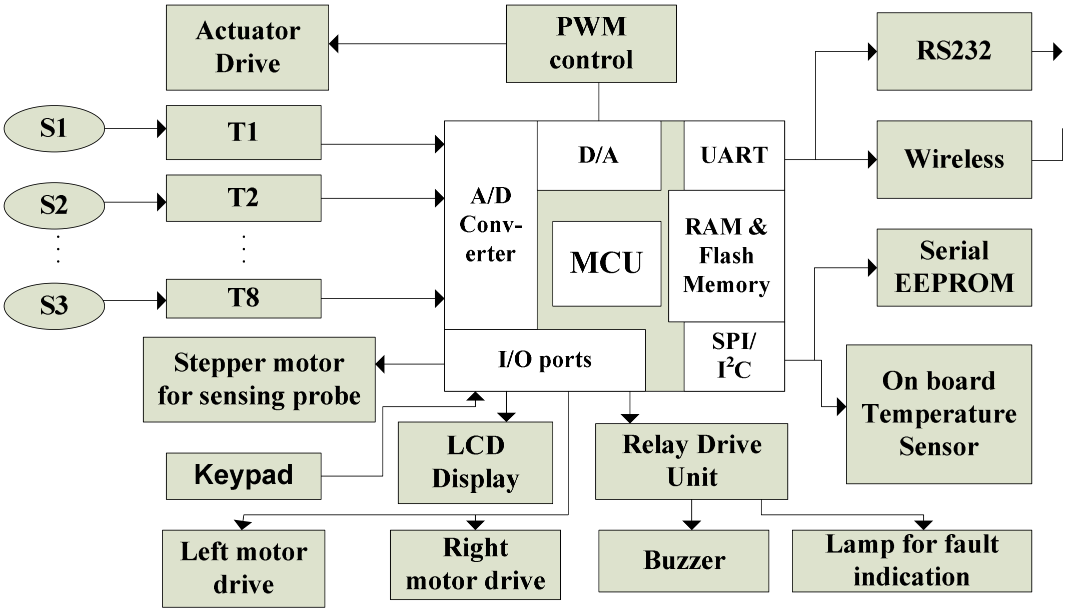


Figure 1. General Architecture of developed system.

III. System description

The generic architecture of system is shown in figure 1. It consists of eight sensors with transmitters. To make the system more compact MICROCHIP's 8 bit PIC18f452 has been used which has most of the peripherals on chip. It has low power consumption, fast executing speed and on chip 1536 bytes of flash and 256 bytes of data EEPROM memory. It can operate up to 10MIPS (DC to 40MHz). There are 18 interrupt sources, two-timer modules, two capture/ compare / PWM modules, 8x8 hardware multiplier and master synchronous serial port module having SPI and I2C interface. On chip serial communication supports both RS232 and RS485. The system can work up to 40 MHz clock frequency. But response time of many sensors is of the order of 100µs or more [12]. So, a processor speed of 10MHz to 15MHz should be adequate. Considering the optimum performance and cost of the overall system and due to easy availability of crystal oscillators up to 16MHz, the system clock of 10 MHz is selected for current application. There is 10 bits, 8 analog input channels A/D converter with acquisition time 12.86µs. The eight input sensor nodes operate under stored program control [14]. The micro controller A/D converter performs periodic scans of these sensors. The scan rate is programmable and can be adaptive based on the rate of change of sensor reading. The sensors data are compared with set-point values stored in memory.

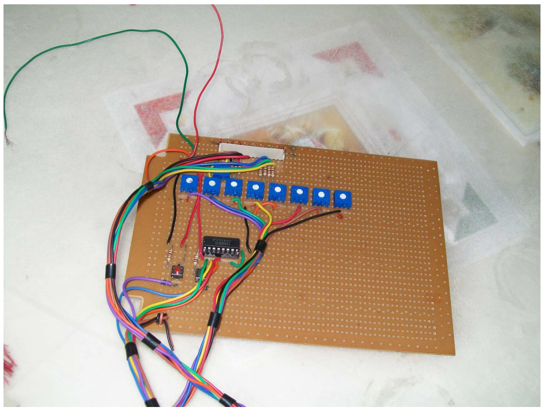
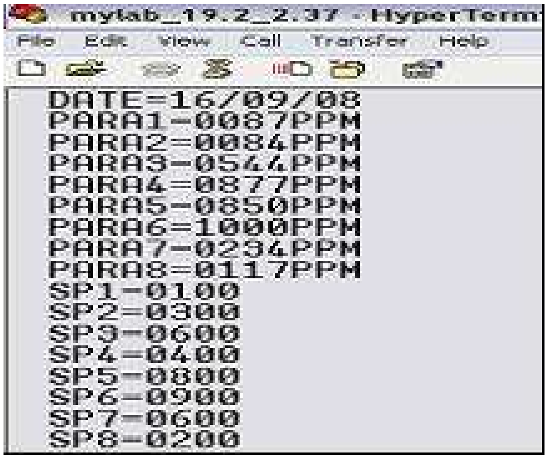


Figure 2. Eight potentiometer based kit.

For testing of eight channels, an eight potentiometer based kit has been developed in laboratory shown in figure 2. The parameters are firstly checked with eight potentiometer kit and then module has been checked by RTD sensors. The eight potentiometers are connected at PortA, analog input channels of microcontroller. Voltage (0-5V) is converted into corresponding engineering units for different parameters. The control of system can be done by two switches (RB0 and RB1) connected on port B of microcontroller. The value of parameter one (0 - 1000PPM) can be adjusted by first potentiometer and when switch RB0 is pressed it comes to the set point value for first parameter which is already stored in flash memory. The set point value can be increment or decrement (0-1000PPM at span of 100PPM) by pushing switch RB0 according to user's need. If switch RB1 is pressed, it escapes from set point routine and automatically compares set point and parameter value to give alarm indication if sensor parameter value is greater than set point values and parameter value can be adjusted, if needed. Similarly, using second potentiometer, parameter two is tested and set point value can be set for second parameter. If another parameter has to be set among eight parameters, the processor can switch to them periodically in similar manner by pressing key RB1. Alarm acquisition is done by open collector source ULN2003 (transistor array).

In between all process, the processor continuously checks if character '*' has been received from keyboard, then it transmits all parameters and set point values to the hyper terminal as shown in figure 3 and come back to the main program. All measured data is stored in on chip Flash and also serial EEPROM has been interfaced via I2C bus to utilize the non – volatile memory. All values are transmitted to LCD unit using the LCD interface. The parameter values and their set - point values can be transmit via RS232 or wireless module (DIGI's X-Bee – PRO 802.15.4 transceiver) which has capability to transmit data up to 1.6 km line of sight at frequency 2.4 GHz. In general, a lower frequency allows a longer transmission range and stronger capability to penetrate through walls and glasses [3]. However, the GHz bands of 2.400-2.4835 are worldwide acceptable. The microcontroller apart from the measurement of analog parameters [5] also performs all the required housekeeping tasks and interacts with the other peripherals. A battery supply monitoring circuit generates an interrupt on detection of a battery fail condition and initiates an emergency measurement backup. There are two main time intervals that need to be maintained. The sensors with their transmitters are carefully timed using software timers. Assembly and C language programming is done using MPLAB IDE tool along with PICDEM-2PLUS debugger and programmer for implementation of developed algorithms, monitoring and display of parameters and set point values.



```
DATE=16/09/08
PARA1-0087PPM
PARA2=0084PPM
PARA3-0544PPM
PARA4=0877PPM
PARA5-0850PPM
PARA6=1000PPM
PARA7-0294PPM
PARA8=0117PPM
SP1-0100
SP2=0300
SP3-0600
SP4=0400
SP5-0800
SP6-0900
SP7=0600
SP8-0200
```

Figure 3. Data transmission at hyper terminal.

The developed multi-sensing system performs both tasks viz. sensing of parameters as well as controlling the movements of the mobile robot using left and right DC motors [14]. The summary of hardware configuration for mobile multi-sensor embedded system is as follows:

- Central PC: Data acquisition using wireless communication and analysis
- PIC18F452 – 4 MHz, RAM 1.5kbytes, flash program memory – 32kbytes, 8 channel - 10-bit A/D converter for sensor and motor control
- Soil temperature and humidity measurement using sensor s LM335 and RDT respectively
- ZigBee/802.15.4 RS232 RF Modem – wireless connectivity between mobile robot and central PC
- Two DC motors: Supply voltage 12V, 6mm shaft diameter, 30 RPM
- One stepper motor: for mobile robot arm for sensor assembly
- Stepper motor driver: 1 L293. Battery: 12volt, 3.6Ah

The temperature sensors LM335 and RTD are mounted on the robotic arm assembly. The sensors calibration circuitry is also included in the design. The sensor interface circuit to PIC18f452 is composed of unity gain amplifiers followed by a filter circuit to reduce the loading effect and noise respectively. The filtered signal is given to the A/D converter for sensor signal digitization.

The motor driver circuit is designed to drive DC motor and stepper motor. The stepper motor for arm is driven in half stepping mode, so that the effective step angle is 1.8 degrees. IC L293 is used to drive the DC and stepper motors, which limits the current up to 600mA.

A. Temperature measurement

The temperature measurement is implemented using the PT100 Resistance Temperature Detector (RTD) sensors [15]. The transmitter circuit takes RTD as input and provides 4-20mA output corresponding to the measuring range of 0-50°C. The XTR-103 has built-in provisions for RTD current excitation, signal amplification and liberalization on a single integrated circuit. The zero and span adjustment are carried out to get 4-20mA output signals for a working temperature range of 0-50°C. These 4-20mA current signals are interfaced to the ADC input port of the micro controller for further processing. First, temperature transmitter circuit is calibrated by connecting a standard resistance box. The resistance is set to 100Ω, corresponding to 0^{o}C, and then ZERO adjustment is done to get 4mA at the output. Similarly it is set to 119.4Ω, corresponding to 50^{o}C, and SPAN adjustment is done to get 20mA at the output. Similarly the other temperature transmitter circuits are calibrated.

B. Relative humidity measurement

Measurement of relative humidity in the field plays an important role during the harvesting of the seeds [15]. So this facility is also included in the designed system.

1) Measurement Technique

Psychrometry is the best-suited technique for measuring relative humidity particularly in process industries, which call for rugged and continuous operation [16]. The accuracy of the dry and wet bulb sensors, the maintenance of a minimum ventilation speed, and a clean wick are the major factors that affect the measurement of the relative humidity. When the wick of the psychrometer becomes hard, the bulb will not be thoroughly wet and inaccurate readings will result. Care should be taken to keep the wick clean. It should not be handled unless the fingers are perfectly clean. Distilled water should be used for keeping the wick wet. Accuracy with the fixed position psychrometers can be obtained only by creating free circulation of ambient air around the bulbs. Since relative humidity depends on more than one parameter, multiple regression method is used to derive the relationship. Again by dividing the operating range into three zones, viz. 0^{o}C - l0^{o}C, 11^{o}C - 20°C and 21°C – 50°C and then by using the MATLAB and Simulink software package, second order equations are developed for the entire range of temperature 0^{o}C - 50°C to get the desired accuracy of about ± % in the RH measurement. The equation for the temperature range of 0^{o}C to 10°C is :

$$RH = 97.4056 - 15.4898\, T_d + 16.2592\, T_w + 0.2317\, T_d^2 - 0.2972\, T_w^2 \qquad (1)$$

The equation for range the 11°C to 20°C is:

$$RH = 93.0129 - 12.6491\, T_d + 13.4314\, T_w + 0.1229\, T_d^2 - 0.1485\, T_w^2 \qquad (2)$$

The equation for the temperature range of 21°C -50°C is:

$$RH = 91.0058 - 9.0872\, T_d + 9.5597\, T_w + 0.0471\, T_d^2 - 0.0540\, T_w^2 \qquad (3)$$

where, T_d – dry bulb temperature and T_w – wet bulb temperature.

2) Dry and wet bulb temperature transmitter

The dry and wet bulb temperature transmitter is implemented using the PT100 Resistance Temperature Detector (RTD) sensors [17]. A mechanism of a water container for keeping the wet bulb wet, by covering the corresponding RTD with one end of a wick and keeping the other end of the wick in the water tank is incorporated. At least half of the length of the wick remains dipped in the water. Similarly a fan mechanism is provided for getting the required airflow around the wet RTD. Circuits using XTR 103 are implemented for measuring the two temperatures. Each circuit takes RTD as input and provides 4-20mA output corresponding to the measuring range of 0-50°C. The XTR-103 has built-in provisions for RTD current excitation, signal amplification and liberalization on a single integrated circuit. Two similar circuits, based on XTR-103, are implemented for measuring the dry and wet temperatures. The zero and span adjustment are carried out to get 4-20mA output signals for a working temperature range of 0-50°C. These two 4-20mA current signals are interfaced to the ADC input port of the micro controller card for further processing.

C. Wireless communication

For wireless communication RS 232 based DIGI’s X-Bee–PRO 802.15.4 transceiver module having OQPSK based modulation is used. The operating temperature is –40 °C to 85°C and the power down current is less than 6mA. It can transmit data up to 1 mile for outdoors range (LOS) and 100m in indoor range at frequency 2.4GHz .The receiver sensitivity is - 100dBm. For every 6dB gain in TX power or RX sensitivity the range of wireless link doubles. For wireless communication, location is also an important issue. Therefore, the module has been tested at different distances and different baud rates inside and outside office to analyze the effect of obstacles in line of sight (LOS) up to 300m. The baud rate error was 0.16%, which is obtained from the calculated and desired baud values at 16MHz [18]. A serial RS-232 and wireless interface is provided to communicate with a host system for remote/ automated monitoring. The signal loss and power received have been calculated at different distances by using (4) and (5). We are procuring the portable RF power meter from Keithley (Model 3500) for measurement of actual power loss.

The signal loss is given by-

$$L\ (dB) = -20 \log_{10} (\lambda/\ 4\Pi d). \qquad (4)$$

where – L = signal loss in free space (dB) and d = distance (meters)

The power received at receiver is given by-

$$P_r= (P_t\ G_r\ G_t\ \lambda_2) / ((4\Pi\ )2\ d_2\ L) \qquad (5)$$

where – Pt = transmitted power = 100mW and G_r & G_t = antenna gain at receiver and transmitter = 2.1dB.

D. *Distance measurement*

For testing of information loss at different places two wireless modules are used. One module is connected to the system through RS232 port placed in a test room and another wireless module is connected with a laptop to check the loss at different places. To calculate line of sight distances between two places, help has been taken from Google Earth. Google Earth maps the Earth by superimposition of images obtained from satellite imagery, aerial photography and GIS 3D globe which can be downloaded free. It provides facility to calculate distance between two selected objects under the Tool option by selecting the ruler. Thus we have located our CEERI office area at Pilani, Rajasthan through Google Earth and then calculated approximate distance in meters between two locations in outdoor ranges. For indoor ranges the distance between two rooms are calculated approximately using Pythagoras theorem on first floor and ground floor.

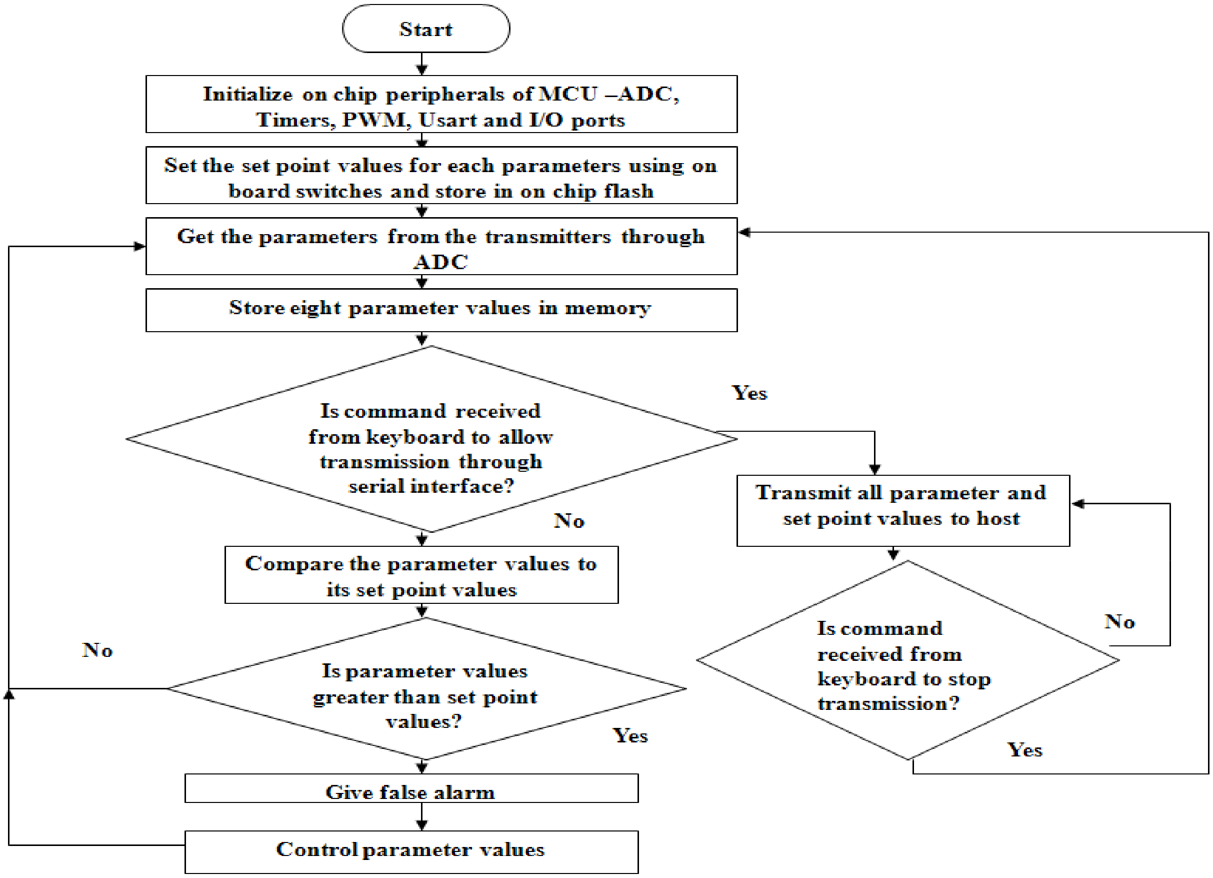

Start
Initialize on chip peripherals of MCU –ADC, Timers, PWM, Usart and I/O ports
Set the set point values for each parameters using on board switches and store in on chip flash
Get the parameters from the transmitters through ADC
Store eight parameter values in memory
Is command received from keyboard to allow transmission through serial interface?
Yes
No
Transmit all parameter and set point values to host
Compare the parameter values to its set point values
Is command received from keyboard to stop transmission?
No
Yes
No
Is parameter values greater than set point values?
Yes
Give false alarm
Control parameter values

Figure 4. Software Flow of Multi-Sensor Embedded System.

IV. System Software

The embedded software to operate the wireless multi-sensor system is implemented. Subroutine modules for sensor signal digitization, motor control, LCD, keypad, wireless data acquisition and transmission & field tracking algorithm are developed in C language and compiled with MP-LAB C18 cross compiler. The hex file is then ported to the flash memory of the PIC18F452 using PIC programmer. The operation of the software for multi-sensor embedded system is shown in the flowchart in figure 4.

V. Experimental results

Figure 5 shows the experimental setup for testing of the developed system. The developed module is interfaced with temperature sensors through transmitters giving 4 to 20mA current outputs. The wireless module is connected through RS232 interface. The device is programmed using MPLAB IDE programmer. Figure 6 shows the results of the temperature measurement conducted by keeping the temperature sensors in the environmental chamber where programmable temperature setting is done for specified temperature. The temperature measurement results are verified and found to be +/- 1% as tabulated in table I over a span of 100 to 120^{o}C. The variation of calculated signal loss and received power against distance for testing wireless modules are shown in figure 7 and figure 8 measured at various aerial distances with different baud rates.

The RH is calculated from measured dry and wet bulb temperatures and the (1) - (3), and is compared with the expected value from the psychometric chart. The results are given in the following tables II, III and IV respectively.

Since, wireless communication works by creating electromagnetic waves at a source and being able to pick up those electromagnetic waves at a particular destination [14]. During the propagation, some attenuation of the signal takes place due to properties of the transmission medium (air) and obstacles and power density decreases at receiving end. Figure 7 shows this attenuation as a function of the distance between the transmitter and receiver and figure 8 shows the power received at receiver end decreases with increasing the distance between transmitter and receiver locations. The information loss in wireless data communication using the ZIGBEE wireless modules is shown in table V and table VI at outdoor and indoor distance range with and without obstacles.

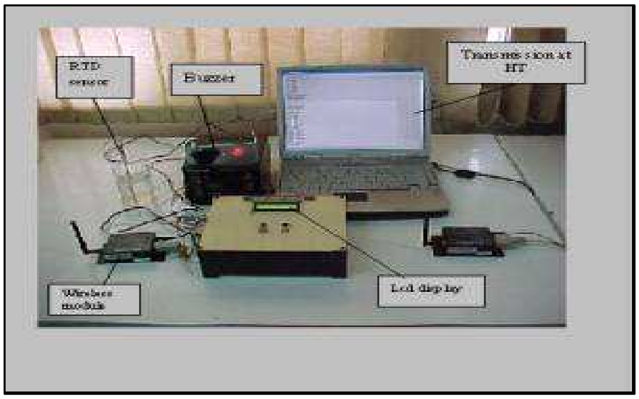


Figure 6. Experimental setup.

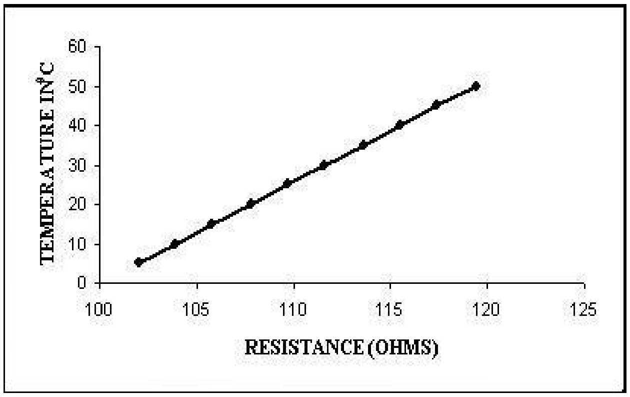


Figure 5. Temperature measurement.

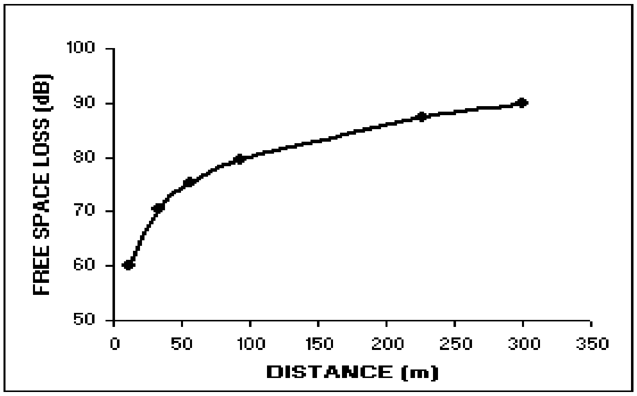


Figure 7. Signal loss.

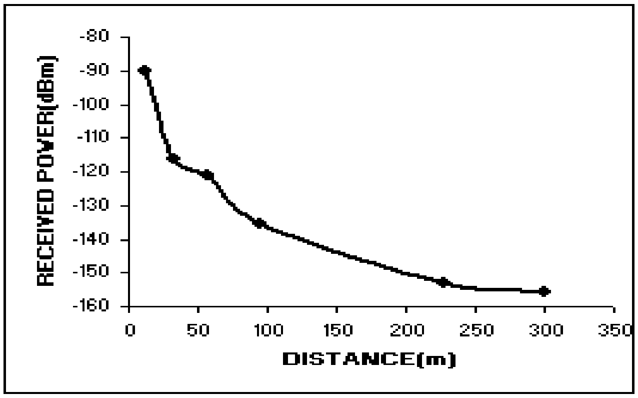


Figure 8. Received power.

TABLE I. TEMPERATURE MEASUREMENT

Resistance (Ohms)	Temperature from Chart(Deg Cen.)	Measured Temperature (Deg Cen)
102.0	5	5.2
103.9	10	10.1
105.8	15	15.0
107.8	20	20.0
109.7	25	25.1
111.6	30	30.1
113.6	35	35.0
115.5	40	40.2
117.4	45	45.1
119.4	50	50.1

TABLE II. COMPARISON BETWEEN MEASURED AND HCART RH VALUES FOR 0-10^{O}C RANGE

Dry Blub Temperature	Wet Bulb Temperature	Measure RH value	RH Value from Psychrometry Chart	Difference between Measured and Chart Values	Percentage Error
2	1.5	91.0728	92	-0.9272	-1.0078
2	1	83.3147	83	0.3147	0.3792
4	3.5	92.4198	93	0.5802	-0.6239
4	3	85.2560	85	0.0256	0.3012
6	5.5	93.2426	94	-0.7574	-0.8057
6	3	58.9099	60	-1.0901	-1.8168
8	7.5	93.5412	94	-0.4588	-0.4881
8	5	62.1805	62	0.1805	0.2911
10	9.5	93.3157	94	-0.0684	-0.7280
10	6.5	58.8034	60	-1.1966	-1.9943

TABLE III. COMPARISON BETWEEN MEASURED AND CHART RH VALUES FOR IO-20°C RANGE

Dry Blub Temperature	Wet Bulb Temperature	Measure RH value	RH Value from Psychrometry Chart	Difference between Measured and Chart Values	Percentage Error
10	9.5	93.0144	94	-0.9856	-1.0485
10	6.5	59.8460	60	-0.1540	-0.2567
12	11.5	93.7525	94	-0.2475	-0.2633
12	11.0	88.7069	89	-0.2931	-0.3293
14	13.5	94.2862	95	-0.7138	-0.7514
14	13.0	89.5375	90	-0.4625	-0.5139
16	12.5	66.7915	67	-0.2085	-0.3112
16	12.0	61.8943	62	-0.1057	-0.1705
18	17.5	94.7409	95	-0.2591	-0.2727
18	17.0	90.5860	90	0.5860	0.6511
20	16.0	66.0971	66	0.0971	0.1471
20	15.5	61.7195	62	-0.2805	-0.4524

TABLE IV. COMPARISON BETWEEN MEASURED AND CHART RH VALUES FOR 20-50°C RANGE

Dry Blub Temperature	Wet Bulb Temperature	Measure RH value	RH Value from Psychrometry Chart	Difference between Measured and Chart Values	Percentage Error
20	19.50	93.9807	96	-2.0193	-2.1034
20	15.50	63.3039	62	1.3039	2.1031
22	21.50	94.4538	96	-1.5462	-1.6106
24	19.00	62.1831	63	-0.8169	-1.2967
26	25.50	95.2343	96	-0.7657	-0.7976
28	26.00	85.5367	85	0.5367	0.6314
30	29.50	95.7938	96	-0.2062	-0.2148

32	30.00	86.6340	86	0.6340	0.7372
34	32.00	87.0997	87	0.0997	0.1146
36	30.00	63.0991	64	-0.9009	-1.4077
38	31.00	58.1618	60	-1.8382	-3.0637
40	39.50	96.2262	97	-0.7738	-0.7977
40	33.00	59.5422	61	-1.4578	-2.3898
42	41.50	96.1469	97	-0.8531	-0.8795
44	37.00	62.1372	63	-0.8628	-1.3695
46	45.50	95.8228	97	-1.1772	-1.2136
46	45.00	93.4871	94	-0.5129	-0.5456
48	47.50	95.5779	97	-1.4221	-1.4661
50	42.00	60.6471	62	-1.3529	-2.1821

TABLE V. INFORMATION LOSS – INDOOR RANGE

S.N.	Baud rate (bps)	Baud rate error (%)	Distance (m) with obstacles	Distance (m) without obstacles	Information loss (Yes/No)
1	1200	0.16	-	<50	No
			57	-	Yes
			64	-	
			-	227	No
2	9600	0.16	-	<50	No
			57	-	Yes
			64	-	
			-	227	No
3	19200	0.16	57		Yes
			64		
				227	No

TABLE VI. INFORMATION LOSS – OUTDOOR RANGE

S.N.	Baud rate (bps)	Baud rate error (%)	Distance (m)	Information loss (Yes/No)
1	1200	+0.16	<20	No
			33	Yes
			57	
			61	
2	9600	+0.16	<20	No
			33	Yes
			57	
			61	
3	19200	+0.16	<20	No
			33	Yes
			57	
			61	

The developed multi-sensor embedded system is also tested on the mobile robotic platform in the field of an area 5m x 5m. The system is tested with LM 335 and RTD temperature sensors for temperature and humidity measurement of soil in the farm over a wireless connectivity. The observed trajectory of the wireless robot in the tested field of 5mx5m area is as shown in figure 9.

VI. CONCLUSION AND FUTURE WORK

A development of the multi sensor embedded system for measuring up to eight sensor parameters optimized by appropriate algorithms with its various features are presented. The test results of the developed prototype system with RTD temperature sensors for temperature measurement over a wireless connectivity for various distances are presented. From the results it can be concluded that for better transmission of signals via wireless communication, the low frequency range along with low baud rate and line of sight range is required to minimize the signal loss. The developed system is cost effective, versatile, and based on generic platform. Currently, efforts are going on to increase measurements up to 16 sensors channels, which are suitable for most of our current applications like RO water purification Plant automation and Smart pond fish management system, where in up to 15 parameters are to be monitored. Also implementation of appropriate wireless reconfigurable sensor network for future perspective so as to make the developed embedded system more applicable in Agro based Industrial applications such as grain and fruit, storage, vegetable storage smart pond automation for fresh aquaculture and so on.

The developed multi-sensor embedded system on mobile robot is also successfully tested to monitor the soil parameters in the farm and transmit the data to the central system using wireless protocol for precision agriculture. Considering the high cost and complexity of number of wireless sensing network systems in agriculture, the proposed system is low power and cost effective with less number of sensors and wireless nodes. Although the experimental conditions were limited, but the preliminary results are promising and shows that the wireless multi-sensing system can be utilized for farm application in precision agriculture.

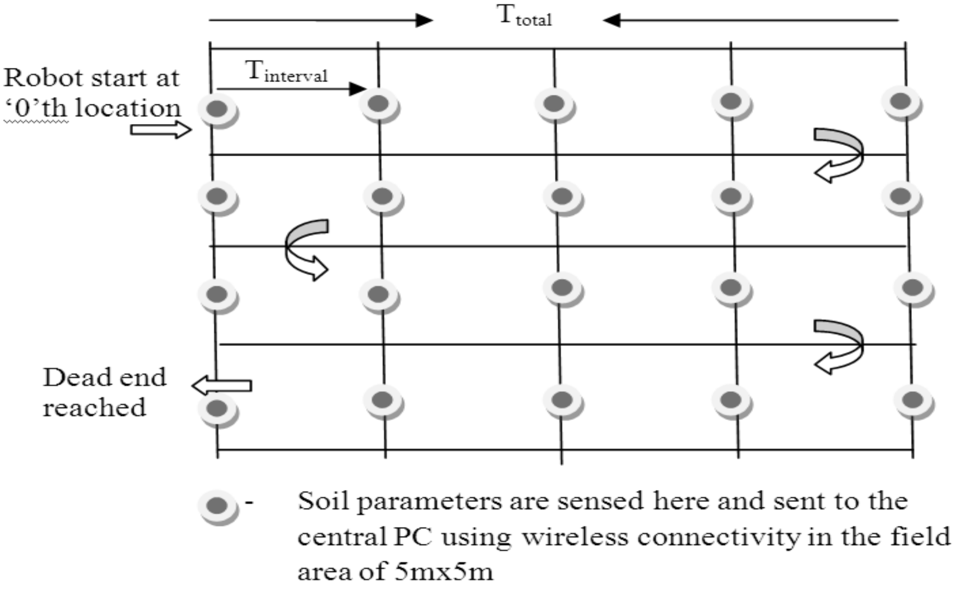

T_{total}
$T_{interval}$
Robot start at
'0'th location
Dead end
reached
- Soil parameters are sensed here and sent to the central PC using wireless connectivity in the field area of 5mx5m

Figure 9. Trajectory of the Wireless Mobile Robot sensing soil parameter.

Acknowledgment

The authors are thankful to Dr. Chandra Shekhar, Director CEERI and guidance. Thanks are due to Mr. A. P. Sharma, Mr. Pankaj Kumar and Mr. Ravi Chaudhary for their help and support for the work.

References

[1] Chandani Anand, Shashikant Sadistap, Satish Bindal and K. S. N. Rao, "Multi -Sensor Embedded System for Agro-Industrial Applications", Proc. 3rd IEEE conference ICN 2008, pp. 94-99, 2009.

[2] H. Ramamurthy, B. S. Prabhu, R. Gadh and A. M. Madni, "Wireless Industrial Monitoring and Control using a Smart Sensor Platform" IEEE sensors journal, vol. 7, no. 5, pp. 611-618 2007.

[3] N. Wang, N. Zhang and M. Wang, "Wireless sensors in agriculture and food industry—Recent development and future perspective", Journal of Computers and Electronics in Agriculture, vol. 50, pp. 1–14, 2006.

[4] B. A. Botre, Ganji Manu, P. Shashank Reddy and S. S. Sadistap, "Development of Mobile Embedded System for Farm Sensing", Journal of Instrument Society of Inda, vol. 40, no. 1, pp 4-7, India, 2010.

[5] IEEE Standard for a Smart Transducer Interface for Sensors and Actuators- IEEE 1451.3 edition, 2004.

[6] L. Liang, L. Huang, X. Jiang and Y. Yao, "Design and implementation of wireless Smart-home sensor network based on ZigBee protocol", Proc. of IEEE International conference on communication, circuits and systems, (ICCCAS-2008), pp. 434-439, 2008.

[7] S. S. Sadistap, Y.K Jain, L. Narayan and S. Devi, "Smart sensor development for CO sensor", Proc. National conference on Instrumentation, (NSI 32), Tamil Nadu, 2002.

[8] R. Vrba, O. Sajdl and M. Sveda, "Smart sensor in wireless network system", Proc. of IEEE 3rd International conference on Information and Communications Technology, pp. 23-33, 2005.

[9] R. Behnke, F. Golatowski, K. Thurow, and D. Timmermann, "Wireless sensor networks for life science automation", In International Forum Life Science Automation, 2007.

[10] J. M. Kahn, R. H. Katz, and K. S. J. Pister. "Mobile Networking for Smart Dust", Proc. of ACM/IEEE International Conference on Mobile Computing and Networking, (MOBICOM-99), March 1999.

[11] www.smartsensors.com

[12] Blackmore, S., "Precision Farming: An Introduction", Outlook on Agriculture, vol. 23, no. 4, pp. 275-280, 1994.

[13] A. Mason, N. Yazdi and A. V. Chavan "A generic multi-element micro-system for portable wireless applications" Proceedings of the IEEE, vol. 86, no. 8, pp. 1733 – 1746, 1998.

[14] B. A. Botre, S. B. Waphare, D. Shirke, D. C. Gharpure, and A. D. Shaligram, "Mobile Odor Tracking Robot Based On Embedded Technology", Proc. IEEE International conference Emerging Trends in Electronic and Photonic Devices & Systems (ELECTRO-2009), Varanasi, pp. 111-114, 2009.

[15] V. Sai Krishana, S. S. Sadistap, Bhanu Prasad and Pawan Kapur, “Microcontroller based relative humidity meter”, Journal of Instrument Society of India, vol. 35, No. 4, pp. 350-356, 2003

[16] L. R. Adams, and M. K. Pritchard, “A low-cost microcomputer system for controlling relative humidity in horticultural storages”, Hort Technology, vol. 4, no.1, pp. 51-54, 1994.

[17] P. Kapur, P. Bhanu Prasad, M. Singh, B. Das, M. P. Kaushik, and K. K. Pandey, “PC based Systems for Composting and Cropping Stages of Mushroom Cultivation”, IETE Technical Review, vol. 20, no.3, pp. 241-249, 2003.

[18] Technical data manual PIC18F452, available online, pp. 185-188.

Distributed Monitoring Systems for Agriculture based on Wireless Sensor Network Technology

Davide Di Palma, Luca Bencini, Giovanni Collodi,
Gianfranco Manes, Francesco Chiti, Romano Fantacci
Department of Electronics and Telecommunications
University of Florence
Via di Santa Marta 3, 50019 Florence Italy
name.surname@unifi.it

Antonio Manes
Netsens S.r.l.
Via Tevere 70, 50019 Florence Italy
antonio.manes@netsens.it

***Abstract*—The adoption of Wireless Sensor Networks (WSN) for wide area environmental monitoring is currently considered one of the most challenging application scenarios for this emerging technology. The promise of an unmanaged, self-configuring and self-powered wireless infrastructure, with a continuously decreasing cost per unit, attracts the attention of both final users and system integrators, replacing previously deployed wired solutions and opening new business opportunities. Agricultural scenarios seem to be one of the most promising application areas for WSN due to the necessity of improving the agro-food production chain in terms of precision and quality. This involves a careful system design, since a rural scenario consists of an extensive area devoid of an electrical power supply and availabile wired connections. This paper shows and describes a practical case study, starting from a real problem and reaching the best architectural solutions with particular focus on hardware implementation and communication protocol design. Moreover, encouraging and unprecedented results are shown, achieved with this approach and supported by several pilot sites in different vineyards throughout Italy and France. Finally, the commercial system ”VineSense”, born from previous experimental solutions, and its agronomic results are also presented.**

***Keywords*-Wireless Sensor Network, Distributed Agricultural Monitoring, Hardware and Protocol Design, Physiology and Pathogens Control, Pilot Sites.**

I. Introduction

Agriculture is one of the most ancient activities of man in which innovation and technology are usually accepted with difficultly, unless real and immediate solutions are found for specific problems or for improving production and quality. Nevertheless, a new approach, of gathering information from the environment, could represent an important step towards high quality and ecosustainable agriculture.

Nowadays, irrigation, fertilization and pesticides management are often left to the farmer and agronomist's discretion: common criteria used to guarantee safe culture and plant growth is often giving a greater amount of chemicals and water than necessary. There is no direct feedback between the decision of treating or irrigating plants and the real effects in the field. Plant conditions are usually committed to sporadic and faraway weather stations which cannot provide accurate and local measurements of the fundamental parameters in each zone of the field. Also, agronomic models, based on these monitored data, cannot provide reliable information. On the contrary, agriculture needs detailed monitoring in order to obtain real time feedback between plants, local climate conditions and man's decisions.

The most suitable technology to fit an invasive method of monitoring the environment is a *Wireless Sensor Network* (WSN) system [2].

The requirements that adopting a WSN are expected to satisfy in effective agricultural monitoring concern both *system level* issues (i.e., unattended operation, maximum network life time, adaptability or even self-reconfigurability of functionalities and protocols) and *final user* needs (i.e., communication reliability and robustness, user friendly, versatile and powerful graphical user interfaces). The most relevant mainly concerns the supply of *stand-alone* operations. To this end, the system must be able to run unattended for a long period, as nodes are expected to be deployed in zones that are difficult to maintain. This calls for optimal energy management. An additional requirement is *robust* operative conditions, which needs fault management since a node may fail for several reasons. Other important properties are scalability and adaptability of the network's topology, in terms of the number of nodes and their density in unexpected events with a higher degree of responsiveness and reconfigurability. Finally, several user-oriented attributes, including fairness, latency, throughput and enhanced data querying schemes [3] need to be taken into account even if they could be considered secondary with respect to our application purposes because the WSN's cost/performance trade-off [4].

The before mentioned requirements call for a carefully designed and optimized overall system for the case study under consideration.

In this paper an end-to-end monitoring solution is presented [1], joining hardware optimization with communications protocols design and a suitable interface. In particular, Section II provides an overview of the related works. Section III presents the overall system in terms of hardware, protocol and software design. Section IV describes the real experiences, focusing on several case studies analyses for highlighting the effectiveness and accurateness of the developed system. Section V and Section VI describe respectively the commercial system ”VineSense”, born from the experimental solution, and some agronomic results. Finally, in Section VII some conclusions are drawn in order to explain the future direction of the current research study.

II. Related Works

The concept of precision agriculture has been around for some time now.

Starting from 1994, Blackmore et al. [5] defined it as a comprehensive system designed to optimize agricultural production by carefully tailoring soil and crop management to correspond to the unique condition found in each field while maintaining environmental quality. The early adopters during that time found precision agriculture to be unprofitable. Moreover they found the instances of implementation of precision agriculture were few and far between. Further, the high initial investment in the form of electronic equipment for sensing and communication meant that only large farms could afford it. The technologies proposed at that point comprised of three aspects: *Remote Sensing* (RS), *Global Positioning System* (GPS) and *Geographical Information System* (GIS). RS coupled with GPS coordinates produced accurate maps and models of the agricultural fields. The sampling was typically through electronic sensors such as soil probes and remote optical scanners from satellites. The collection of such data in the form of electronic computer databases gave birth to the GIS. Statistical analyses were then conducted on the data and the variability of agricultural land with respect to its properties was charted. The technology apart from being non real-time, involved the use of expensive technologies like satellite sensing and was labor intensive where the maps charting the agricultural fields were mostly manually done.

Over the last seven years, the advancement in sensing and communication technologies has significantly brought down the cost of deployment and running of a feasible precision agriculture framework. Emerging wireless technologies with low power needs and low data rate capabilities, which perfectly suites precision agriculture, have been developed [6].

The sensing and communication can now be done on a real-time basis leading to better response times. The wireless sensors are cheap enough for wide spread deployment in the form of a mesh network and offers robust communication through redundant propagation paths [7]. Wireless sensor networks allow faster deployment and installation of various types of sensors because many of these networks provide self-organizing, self-configuring, self-diagnosing and self-healing capabilities to the sensor nodes.

The applications using wireless sensor technology for precision agriculture are briefly explored below.

Cugati et al. [8] developed an automated fertilizer applicator for tree crops. The system consisted of an input module for GPS and real-time sensor data acquisition, a decision module for calculating the optimal quantity and spread pattern for a fertilizer, and an output module to regulate the fertilizer application rate. Data communications among the modules were established using a Bluetooth network.

Evans and Bergman [9] are leading a USDA research group to study precision irrigation control of self-propelled, linear-move and center-pivot irrigation systems. Wireless sensors were used in the system to assist irrigation scheduling using combined on-site weather data, remotely sensed data and grower preferences.

The US Department of Agriculture (USDA) [10] conducted a research in Mississippi to develop a high-speed wireless networking system to help farmers download aerial images via WLAN to their PCs, laptops or PDAs. The images were mainly used for precision farming applications.

Mahan and Wanjura [11] cooperated with a private company to develop a wireless, infrared thermometer system for in-field data collection. The system consisted of infrared sensors, programmable logic controllers and low power radio transceivers to collect data in the field and transmit it to a remote receiver outside the field.

The Institut Für Chemie und Dynamik der Geosphäre [12] developed a soil moisture sensor network for monitoring soil water content changes at high spatial and temporal scale.

III. Overall System Characterization

The overall system is shown in Figure 1. It is comprised of a self-organizing WSN endowed with sensing capabilities, a GPRS Gateway which gathers data and provides a TCP-IP based connection toward a Remote Server and a Web Application which manages information and makes the final user capable of monitoring and interacting with the instrumented environment.

In the following subsections, an end-to-end system description of the hardware, protocol and software design is presented.

A. Hardware Design

Focusing on an end-to-end system architecture, every constitutive element has to be selected according to the application requirements and scenario issues, especially the hardware platform. Many details have to be considered, involving the energetic consumption of the sensor readings, the power-on and power-save states management and a good trade-off between the maximum radio coverage and the transmitted power. After an accurate investigation of the out-of-the-shelf solutions, accordingly to these constraints and to the reference scenarios, 868 MHz *Mica2* motes [13] were adopted. The *Tiny Operative System* (TinyOS), running

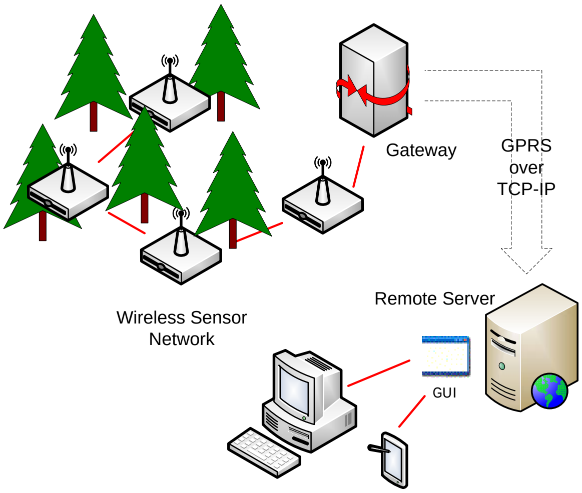


Figure 1. Wireless Sensor Network System.

on this platform, ensures a full control of mote communication capabilities to attain optimized power management and provides necessary system portability towards future hardware advancements or changes. Nevertheless, *Mica2* motes are far from perfection especially in the RF section, since the power provided by the transceiver (*Chipcon CC1000*) is not completely available for transmission, but it is lost to imperfect coupling with the antenna, thus reducing the radio coverage area. An improvement of this section was performed, using more suitable antennas and coupling circuits and increasing the transmitting power with a power amplifier, thus increasing the output power up to 15 dBm, respecting international restrictions and standards. These optimizations allow for a larger radio coverage (about 200 m) and better power management. In order to manage different kinds of sensors, a compliant sensor board was adopted, allowing up to 16 sensor plugs on the same node, this makes a single mote capable of sensing many environmental parameters at a time [14]. Sensor boards recognize the sensors and send *Transducer Electronic Datasheets* (TEDS) through the network up to the server, making an automatic sensor recognition possible by the system. The overall node stack architecture is shown in Figure 2.

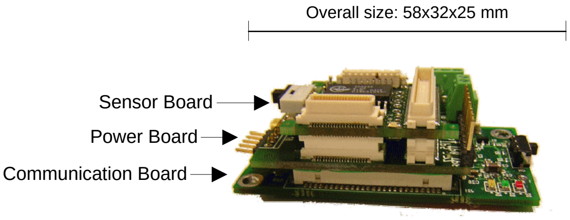


Figure 2. Node Stack Architecture.

The *GPRS embedded Gateway* [15], shown in Figure 3, is a stand-alone communication platform designed to provide transparent bi-directional wireless TCP-IP connectivity for remote monitoring in conjunction with *Remote Data Acquisition* (RDA) equipment, such as WSN where it acts when connected with a Master node or when directly connected to sensors and transducers (i.e., Stand-Alone weather station, Stand-Alone monitoring camera). The Gateway can be configured to operate in several ways like ”*always on*”, ”*on demand*”, or ”*periodically* connected”. The Gateway subsystem has been designed to operate unattended, in outdoor environments, since there is usually no access to a power supply infrastructure. Therefore, the hardware design has been oriented to implement low power operating modalities, using a 12 V rechargeable battery and a 20 W solar panel. Data between the Gateway and Protocol Handler are carried out over TCP-IP communication and encapsulated in a custom protocol; from both local and remote interfaces it is also possible to access part of the Gateway's configuration settings.

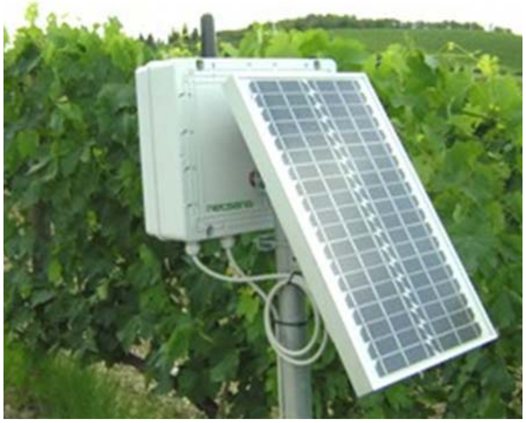


Figure 3. GPRS Gateway.

B. Protocol Design

The most relevant system requirements, which lead the design of an efficient Medium Access Control (MAC) and Routing protocol for an environmental monitoring WSN, mainly concern power consumption issues and the possiblity of a quick set-up and end-to-end communication infrastructure that supports both synchronous and asynchronous queries. The most relevant challenge is to make a system capable of running unattended for a long period, as nodes are expected to be deployed in zones that are difficult to maintain. This calls for optimal energy management since a limited resource and node failure may compromise WSN connectivity. Therefore, the MAC and the network layer must be perfected ensuring that the energy used is directly related to the amount of handled traffic and not to the overall working time.

Other important properties are scalability and adaptability of network topology, in terms of number of nodes and their density. As a matter of fact, some nodes may either be turned off may join the network afterward.

Taking these requirements into account, a MAC protocol and a routing protocol were implemented.

1) MAC Layer Protocol: Taking the IEEE 802.11 *Distributed Coordination Function* (DCF) [16] as a starting point, several more energy efficient techniques have been proposed in literature to avoid excessive power waste due to so called idle listening. They are based on periodical preamble sampling performed at the receiver side in order to leave a *low power* state and receive the incoming messages, as in the WiseMAC protocol [17]. Deriving from the classical contention-based scheme, several protocols (S-MAC [18], T-MAC [19] and DMAC [20]) have been proposed to address the overhead idle listening by synchronizing the nodes and implementing a duty cycle within each slot.

Resorting to the above considerations, a class of MAC protocols was derived, named *Synchronous Transmission Asynchronous Reception* (STAR) which is particularly suited for a *flat* network topology and benefits from both WiseMAC and S-MAC schemes. More specifically, due to the introduction of a duty-cycle, it joins the power saving capability together with the advantages provided by the offset scheduling, without excessive overhead signaling. According to the STAR MAC protocol, each node might be either in an idle mode, in which it remains for a time interval T_l (*listening time*), or in an energy saving sleeping state for a T_s (*sleeping time*). The transitions between states are synchronous with a period *frame* equal to $T_f = T_l + T_s$ partitioned in two sub-intervals; as a consequence, a *duty-cycle* function can also be introduced:

$$d = \frac{T_l}{T_l + T_s} \tag{1}$$

To provide the network with full communication capabilities, all the nodes need to be *weakly synchronized*, meaning that they are aware at least of the awaking time of all their neighbors. To this end, as Figure 4 shows, a node sends a *synchronization message* (SYNC) frame by frame to each of its neighbor nodes known to be in the listening mode (Synchronous Transmission), whereas, during the set-up phase in which each node discovers the network topology, the control messages are asynchronously broadcasted. On the other hand, its neighbors periodically awake and enter the listening state independently (Asynchronous Reception). The header of the synchronization message contains the following fields: a unique node identifier, the message sequence number and the *phase*, or the time interval after which the sender claims to be in the listening status waiting for both synchronization and data messages from its neighbors. The phase ϕ is evaluated according to the following rule:

$$\phi_1 = \tau - T_l \tag{2}$$

if the node is in the sleeping mode, where τ is the time remaining to the next frame beginning. Conversely, if the mote is in the listening status, ϕ is computed as:

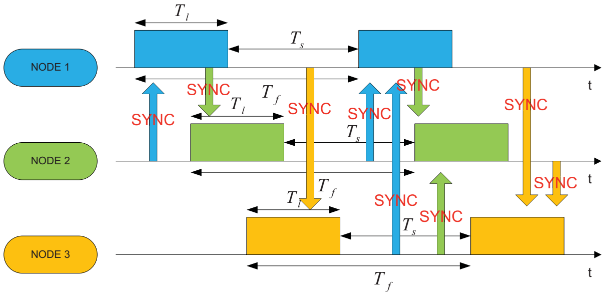


Figure 4. STAR MAC Protocol Synchronization Messages Exchange.

Table I
POWER CONSUMPTION PARAMETERS FOR THE CONSIDERED PLATFORM.

c_{rx}	12 mA
c_{sleep}	0.01 mA
C_{tx}	30 mAh
c_{tx}	0.001 mA

$$\phi_2 = \tau + T_s \tag{3}$$

In order to fully characterize the STAR MAC approach, the related energy cost normalized can be evaluated as it follows:

$$C = c_{rx} d T_f + c_{sleep}[T_f(1-d) - NT_{pkt}] + NC_{tx} \quad [mAh]$$

where c_{sleep} and c_{rx} represent the sleeping and the receiving costs $[mA]$ and C_{tx} is the single packet transmission costs $[mAh]$, while T_{rx} and T_s are the receiving and sleeping times $[s]$, T_{pkt} is the synchronization packet time length $[s]$ and finally N is the number of neighbors. When the following inequality is hold:

$$NT_{pkt} \ll T_f$$

then:

$$C \simeq c_{rx} d T_f + c_{sleep} T_f (1-d) + NC_{tx} \quad [mAh]$$

The protocol cost normalized to the synchronization time is finally:

$$\frac{C}{T_f} = c_{rx} d + c_{sleep}(1-d) + \frac{NC_{tx}}{Tf} \quad [mA] \tag{4}$$

As highlighted in Table I, it usually happens that $c_{tx} \ll c_{sleep} \ll c_{rx}$, where $c_{tx} = C_{tx}/T_{pkt}$ and T_{pkt} is the packet transmission time $[s]$ assumed equal to $100\ ms$ as worst case. This means that the major contribution to the overall cost is represented by the listening period that the STAR MAC protocol tries to suitably minimize.

In Figure 5(a) the normalized cost versus the number of neighbor nodes is shown for the S-MAC and STAR MAC schemes. It is worth noticing that the performance of the proposed protocol is better with respect to the existing

Table II
ROUTING TABLE GENERAL STRUCTURE.

Target	NH	HC	PH	LQ	BL	CL
Sink 1	A	N_A	ϕ_A	η_A	B_A	C_A
	B	N_B	ϕ_B	η_B	B_B	C_B
Sink 2	C	N_C	ϕ_C	η_C	B_C	C_C
	D	N_D	ϕ_D	η_D	B_D	C_D

approach for a number of neighbor nodes greater than 7. Finally, in Figure 5(b) the normalized costs of S-MAC and STAR MAC approaches are compared with respect to the duty cycle duration for a number of neighbor nodes equal to 8. It is possible to notice that for $d < 3.5\%$ the proposed protocol provide a significant gain.

2) Network Layer Protocol: In order to evaluate the capability of the proposed MAC scheme in establishing effective end-to-end communications within a WSN, a routing protocol was introduced and integrated according to the *cross layer* design principle [21]. In particular, we refer to a proactive algorithm belonging to the class link-state protocol that enhance the capabilities of the *Link Estimation Parent Selection* (LEPS) protocol. It is based on periodically information needed for building and maintaining the local routing table, depicted in Table II. However, our approach resorts both to the signaling introduced by the MAC layer (i.e., synchronization message) and by the Network layer (i.e., ping message), with the aim of minimizing the overhead and make the system more adaptive in a cross layer fashion. In particular, the parameters transmitted along a MAC synchronization message, with period T_f, are the following:

- *next hop* (NH) to reach the gateway, that is, the MAC address of the one hop neighbor;
- *distance* (HC) to the gateway in terms of number of needed hops;
- *phase* (PH) that is the schedule time at which the neighbor enter in listening mode according to (2) and (3);
- *link quality* (LQ) estimation as the ratio of correctly received and the expected synchronization messages from a certain neighbor.

On the other hand, the parameters related to long-term phenomena are carried out by the ping messages, with period $T_p \gg T_f$, in order to avoid unnecessary control traffics and, thus, reducing congestion. Particularly, they are:

- *battery level* (BL) (i.e., an estimation of the energy available at that node);
- *congestion level* (CL) in terms of the ratio between the number of packets present in the local buffer and the maximum number of packets to be stored in.

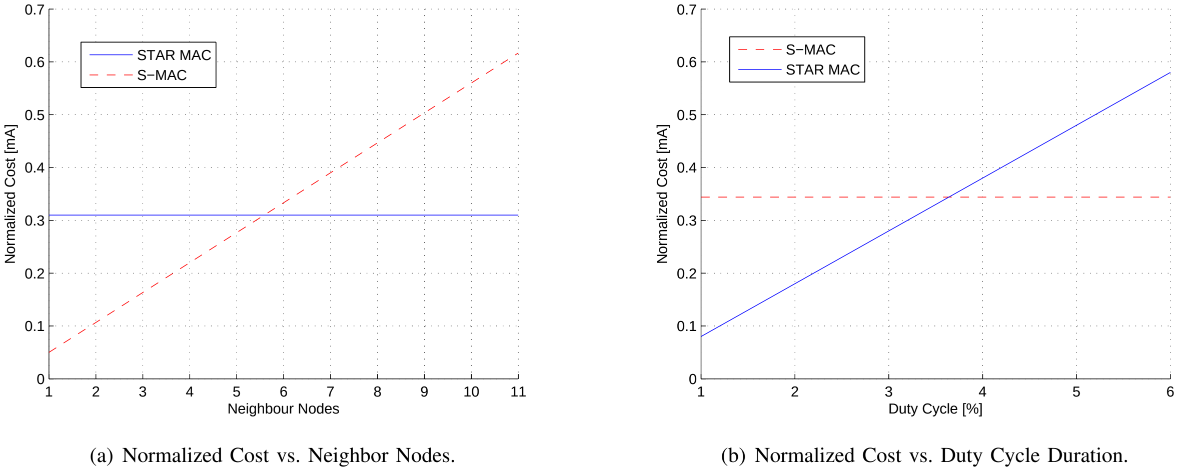
Once, the routing table has been filled with these parameters, it is possible to derive the proper metric by means of a weighted summation of them. It is worth mentioning that the routing table might indicate more than one destination (*sink*) thanks to the ping messages that keep trace of the intermediate nodes within the message header.

C. Software and End User Interface Design

The software implementation was developed, considering a node as both a single element in charge of accomplishing prearranged tasks and as a part of a complex network in which each component plays a crucial role in the network's maintenance. As far as the former aspect is concerned, several TinyOS modules were implemented for managing high and low power states and for realizing a finite state machine, querying sensors at fixed intervals and achieving anti-blocking procedures, in order to avoid software failure or deadlocks and provide a robust stand alone system. On the other hand, the node has to interact with neighbors and provide adequate connectivity to carry the messages through the network, regardless of the destination. Consequently, additional modules were developed according to a cross layer approach that are in charge of managing STAR MAC and multihop protocols. Furthermore, other modules are responsible for handling and forwarding messages, coming from other nodes or from the gateway itself. Messages are not only sensing (i.e., measures, battery level) but also control and management messages (i.e., synchronization, node reset). As a result, a full interaction between the final user and the WSN is guaranteed.

The final user may check the system status through graphical user interface (GUI) accessible via web. After the log-in phase, the user can select the proper pilot site. For each site the deployed WSN together with the gateway is schematically represented through an interactive map. In addition to this, the related sensors display individual or aggregate time diagrams for each node with an adjustable time interval (Start/Stop) for the observation. System monitoring could be performed both at a high level with a user friendly GUI and at a low level by means of message logging.

Figure 6 shows some friendly Flash Player applications that, based on mathematical models, analyze the entire amount of data in a selectable period and provide ready-to-use information. Figure 6(a) specifically shows the aggregate data models for three macro-parameters, such as vineyard water management, plant physiological activity and pest management. The application, using cross light colors for each parameter, points out normal (green), mild (yellow) or heavy (red) stress conditions and provides suggestions to the farmer on how to apply pesticides or water in a certain part of the vineyard. Figure 6(b) shows a graphical representation of the soil moisture measurement. Soil moisture sensors positioned at different depths in the vineyard make it possible to verify whether a summer rain runs off on the soil surface or seeps into the earth and provokes beneficial effects on the plants: this can be appreciated with a rapid look at the soil moisture aggregate report which, shows the



(a) Normalized Cost vs. Neighbor Nodes.

(b) Normalized Cost vs. Duty Cycle Duration.

Figure 5. STAR MAC Performance

moisture sensors at two depths with the moisture differences colore in green tones. Finally, Figure 6(c) highlights stress conditions on plants, due to dry soil and/or to hot weather thanks to the accurate trunk diametric growth sensor that can follow each minimal variation in the trunk giving important information on plant living activity.

IV. Real World Experiences

The WSN system described above was developed and deployed in three pilot sites and in a greenhouse. Since 2005, an amount of 198 sensors and 50 nodes have continuously sent data to a remote server. The collected data represents a unique database of information on grape growth useful for investigating the differences between cultivation procedures, environments and treatments.

A. Pilot Sites Description

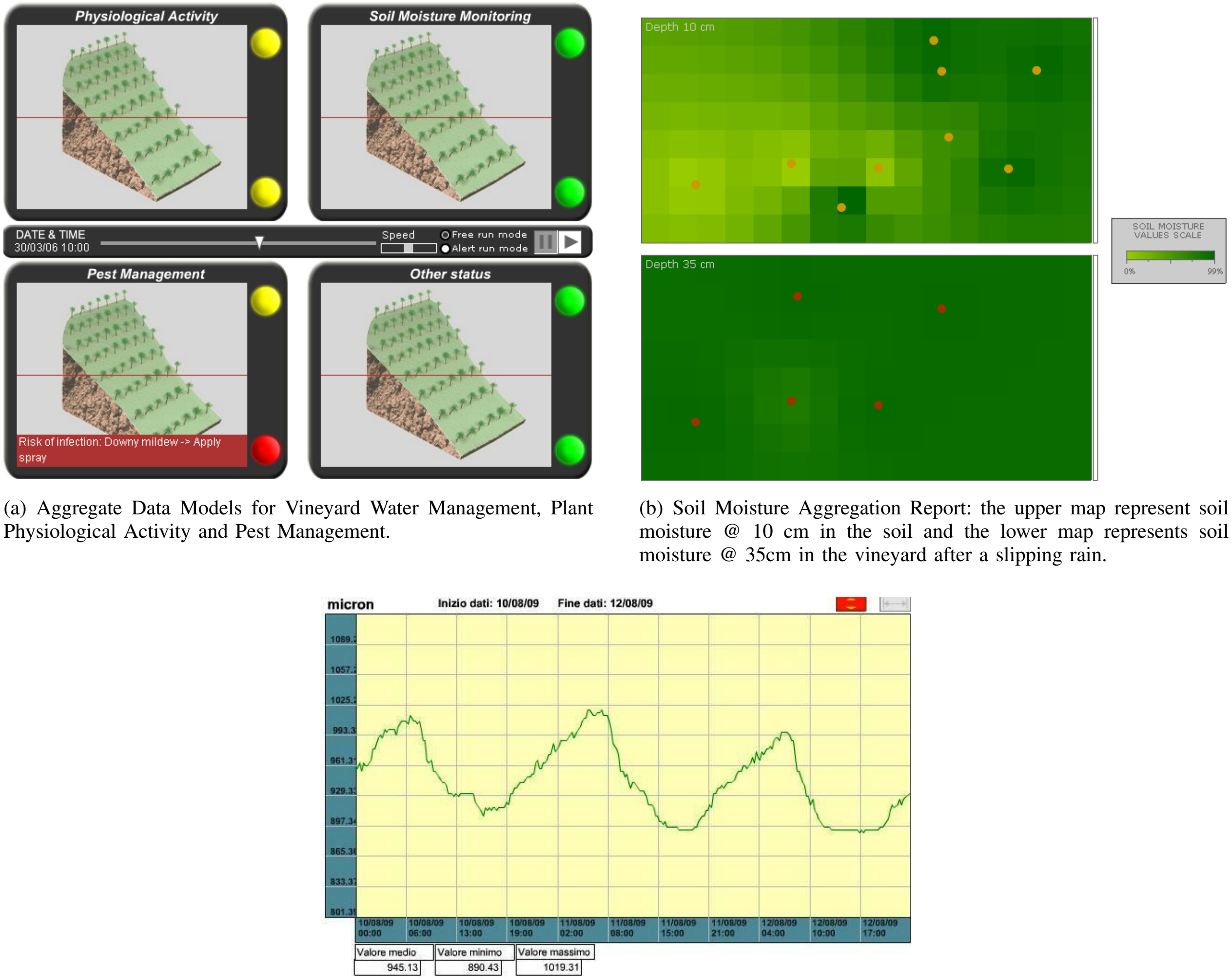
The first pilot site was deployed in November 2005 on a sloped vineyard of the Montepaldi farm in Chianti Area (Tuscany - Italy). The vineyard is a wide area where 13 nodes (including the master node) with 24 sensors, running STAR MAC and dynamic routing protocols were successfully deployed. The deployment took place in two different steps: during the first one, 6 nodes (nodes 9,10,14,15,16,17) were placed to perform an exhaustive one week test. The most important result regards the multi-hop routing efficiency, estimated as:

$$\eta_{MHop} = \frac{M_{EU}}{M_{ex}} \tag{5}$$

where η_{MHop} is the efficiency, M_{EU} are the messages correctly received by the remote user and M_{ex} are the expected transmitted messages. For the gateway neighbors, η_{MHop} is very high, over 90%. However, even nodes far from the gateway (i.e., concerning an end-to-end multihop path) show a message delivery rate (MDR) of over 80%. This means that the implemented routing protocol does not affect communication reliability. After the second deployment, in which nodes 11,12,13,18,19,20 were arranged, the increased number of collisions changed the global efficiency, thus decreasing the messages that arrived to the end user, except for nodes 18,19,20, in which an upgraded firmware release was implemented. The related results are detailed in Table III. This confirms the robustness of the network installed and the reliability of the adopted communications solution, also considering the power consumption issues: batteries were replaced on March 11^{th} 2006 in order to face the entire farming season. After that, eleven months passed before the first battery replacement occurred on February 11^{th} 2007, confirming our expectations and fully matching the user requirements. The overall Montepaldi system has been running unattended for one year and a half and is going to be a permanent pilot site. So far, nearly 2 million samples from the Montepaldi vineyard have been collected and stored in the server at the University of Florence Information Services Centre (CSIAF), helping agronomist experts improve wine quality through deeper insight on physical phenomena (such

Table III
Message Delivery Rate for the Montepaldi Farm Pilot Site

Location	MDR
Node 9	72.2
Node 10	73.7
Node 11	88.5
Node 12	71.4
Node 13	60.4
Node 14	57.2
Node 15	45.6
Node 16	45.4
Node 17	92.1
Node 18	87.5
Node 19	84.1



Valore medio	Valore minimo	Valore massimo
945.13	890.43	1019.31

(a) Aggregate Data Models for Vineyard Water Management, Plant Physiological Activity and Pest Management.

(b) Soil Moisture Aggregation Report: the upper map represent soil moisture @ 10 cm in the soil and the lower map represents soil moisture @ 35cm in the vineyard after a slipping rain.

(c) Trunk Diametric Growth Diagram: daily and nightly metabolic phases.

Figure 6. Flash Player User Interface

as weather and soil) and the relationship with grape growth.

The second pilot site was deployed on a farm in the Chianti Classico with 10 nodes and 50 sensors at about 500 m above sea level on a stony hill area of 2.5 hectares. The environmental variations of the the ”terroir” have been monitored since July 2007, producing one of the most appreciated wines in the world.

Finally, the third WSN was installed in Southern France in the vineyard of Peach Rouge at Gruissan. High sensor density was established to guarantee measurement redundancy and to provide a deeper knowledge of the phenomena variation in an experimental vineyard where micro-zonation has been applied and where water management experiments have been performed for studying plant reactions and grape quality.

B. Greenhouse

An additional deployment at the University of Florence Greenhouse was performed to let the agronomist experts conduct experiments even in seasons like Fall and Winter, where plants are quiescent, thus breaking free from the natural growth trend. This habitat also creates the opportunity to run several experiments on the test plants, in order to evaluate their responses under different stimuli using in situ sensors.

The greenhouse environmental features are completely different from those of the vineyard: as a matter of fact, the multipath propagation effects become relevant, due to the indoor scenario and the presence of a metal infrastructure. A highly dense node deployment, in terms of both nodes and sensors, might imply an increased network traffic load. Nevertheless, the same node firmware and hardware used in the vineyard are herein adopted; this leads to a resulting star topology as far as end-to-end communications are concerned.

Furthermore, 6 nodes have been in the greenhouse since June 2005, and 30 sensors have constantly monitored air temperature and humidity, plants soil moisture and temperature, differential leaf temperature and trunk diametric growth. The sensing period is equal to 10 minutes, less than the climate/plant parameter variations, providing redundant data storage. The WSN message delivery rate is extremely high: the efficiency is over 95%, showing that a low number of messages are lost.

V. VineSense

The fruitful experience of the three pilot sites was gathered by a new Italian company, Netsens, founded as a spin off of the University of Florence. Netsens has designed a new monitoring system called VineSense based on WSN technology and oriented towards market and user applications.

VineSense exalts the positive characteristics of the experimental system and overcomes the problems encountered in past experiences, thus achieving an important position in the wireless monitoring market.

The first important outcome of the experimental system, enhanced by VineSense is the idea of an end-to-end system. Sensors deployed in the field constantly monitor and send measurements to a remote server throught the WSN. Data can be queried and analyzed by final users thanks to the professional and user-friendly VineSense web interface. Qualified mathematic models are applied to monitoring parameters and provide predictions on diseases and plant growth, increasing agronomists' knowledge, reducing costs, while paving the road for new vineyard management.

VineSense improves many aspects of the experimental system, both in electronics and telecommunications.

The new wireless nodes are smaller, more economical, more robust and suited for vineyard operations with machines and tractors. The electronics is more fault tolerant, easier to install and more energy efficient: only a 2200 mAh lithium battery for 2-3 years of continuous running without human intervation. Radio coverage has been improved up to 350 m and nodes deployment can be easily performed by end users who can rely on a smart installation system with instantaneous radio coverage recognition. Sensors used in the VineSense system are low-cost, state-of-the-art devices designed by Netsens in order to guarantee the best accuracy-reliability-price ratio.

The VineSense wireless-sensor unit is shown in Figure 7.

Figure 7. VineSense Wireless Unit.

Recovery strategies and communication capabilities of the stand-alone GPRS gateway have been improved: in fact, data received by wireless nodes are both forwarded in real time to a remote server and temporarily stored on board in case of abrupt disconnections; moreover, automatic reset and restart procedures avoid possible software deadlocks or GPRS network failures; finally, a high gain antenna guarantees a good GPRS coverage almost everywhere. The GPRS gateway firmware has been implemented to allow the remote management of the acquisition settings, relieving users from the necessity of field maintenance.

In Figure 8 the GPRS gateway with weather sensors is shown.

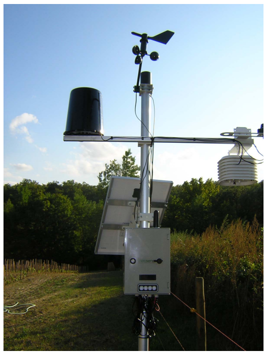


Figure 8. VineSense GPRS Gateway with Weather Sensors.

The web interface is the last part of VineSense's end-to-end: the great amount of data gathered by the sensors and stored in the database needs a smart analysis tool to become useful and usable. For this reason different instruments are at the disposal of various kinds of users: on one hand, some innovative tools such as control panels for real time monitoring or 2D chromatic maps create a quick and easy approach to the interface; on the other hand, professional plots and data filtering options allow experts or agronomists to study them more closely.

VI. Agronomic Results

The use of VineSense in different scenarios with different agronomic aims has brought a large amount of important results.

When VineSense is adopted to monitor soil moisture positive effects can be obtained for plants and saving water, thus optimizing irrigation schedules. Some examples of this application can be found in systems installed in the Egyptian desert where agriculture is successful only through wise irrigation management. In such a terroir, plants suffer continuous hydric stress during daylight due to high air temperature, low air humidity and hot sandy soils with a low water retention capacity. Water is essential for plant survival and growth, an irrigation delay can be fatal for the seasonal harvest therefore, a reliable monitoring system is necessary. The adoption of VineSense in this scenario immediately resulted in continuous monitoring of the irrigating system, providing an early warning whenever pump failure occurred. On the other hand, the possibility to measure soil moisture at different depths allows agronomists to decide on the right amount of water to provide plants; depending on different day temperatures and soil moisture, pipe schedules can be changed in order to reduce water waste and increase water available for plants.

An example of different pipe schedules is shown in Figure 9 . Originally, the irrigation system was opened once a day for 5 hours giving 20 liters per day (schedule 1); since sandy soils reach saturation very rapidly most of this water was wasted in deeper soil layers; afterwards irrigation schedules were changed (schedule 2), giving the same amount of water in two or more times per day; the water remained in upper soil layers at plant root level, reducing wastes and increasing the amount of available water for plants, as highlighted by soil moisture at 60 *cm* (blue plot).

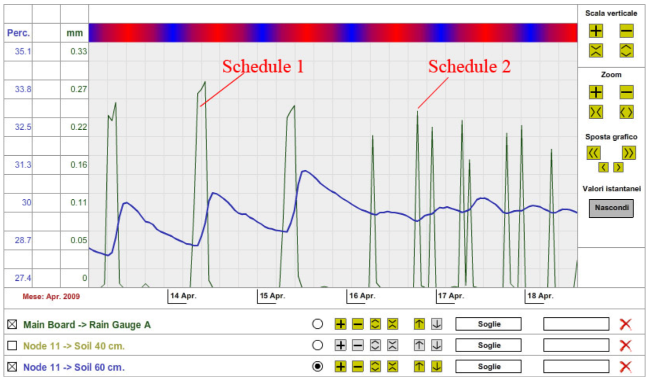


Figure 9. Different Pipe Schedules in Accordance with Soil Moisture Levels.

Another important application of the VineSense system uses the dendrometer to monitor plant physiology. The trunk diametric sensor is a mechanical sensor with $+/-$ 5 microns of accuracy; such an accurate sensor can appreciate stem micro variations occurring during day and night, due to the xilematic flux inside the plant. Wireless nodes measure plant diameter every 15 minutes, an appropriate time interval for following these changes and for creating a plot showing this trend. In normal weather conditions, common physiologic activity can be recognized by agronomists the same as a doctor can do reading an electrocardiogram; when air temperature increases and air humidity falls in combining low soil moisture levels, plants change their activity in order to face water stress, preserve their grapes and especially themselves. This changed behavior can be registered by the dendrometer and plotted in the VineSense interface, warning agronomists about incoming risks; as a consequence, new irrigation schedules can be carried into effect.

Figure 10 shows an example of a plant diametric trend versus air temperature. The blue plot represents the air temperature in 10 days, from 13^{th} August until the 22^{nd} August 2009 in Italy; the blue line becomes red when the temperature goes over a 35 degree threshold. During the period in which the temperature is so high, plant stem variations are reduced due to the lower amount of xilematic flux flowing in its vessels, a symptom of water leakage.

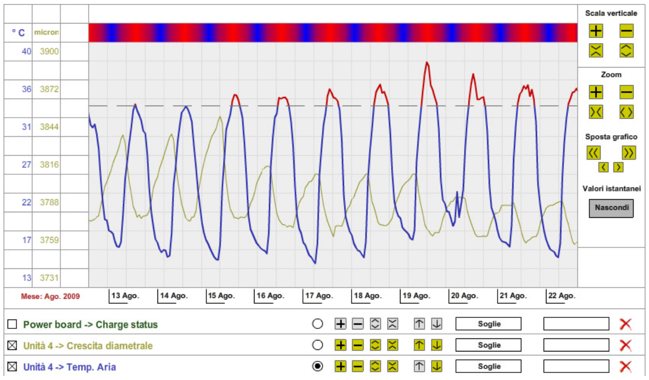


Figure 10. Plant Diametric Trend vs. Air Temperature.

WSN in agriculture are also useful for creating new databases with historical data: storing information highlighting peculiarities and differences of vineyards provides agronomists an important archive for better understanding variations in plant production capabilities and grape ripening. Deploying wireless nodes on plants in interesting areas increases the knowledge about a specific vineyard or a specific terroir, thus recording and proving the specificity of a certain wine. I.E., the quality of important wines such CRU, coming from only one specific vineyard, can be easily related to ”grape history”: data on air temperature and humidity, plant stress, irrigation and rain occurring during the farming season can assess a quality growing process, that can be declared to buyers.

Finally, VineSense can be used to reduce environmental impact thanks to a more optimized management of pesticides in order to reach a sustainable viticulture. Since many of the most virulent vine diseases can grow in wet leaf conditions, it is very important to monitor leaf wetness in a continuous and distributed way. Sensors deployed in different parts of vineyards are a key element for agronomists in monitoring risky conditions: since wetness can change very rapidly during the night in a vineyard and it is not homogeneous in a field, a real time distributed system is the right solution for identifying risky conditions and deciding when and where to apply chemical treatments. As a result, chemicals can be used only when they are strictly necessary and only in small parts of the vineyard where they are really needed, thus reducing the number of treatments per year and decreasing the amount of active substances sprayed in the field and in the environment. In some tests performed in 2009 in Chianti, the amount of pesticides was reduced by 65% compared to the 2008 season.

Figure 11 shows a vineyard map: the green spots are wireless units, distributed in a vineyard of one hectare. Leaf wetness sensors on nodes 2 and 3 measure different wetness conditions as shown in Figure 12. The upper part of the vineyard is usually wetter (brown plot) than the lower part (blue plot) and sometimes leaf wetness persists for many hours, increasing the risk of attacks on plants.

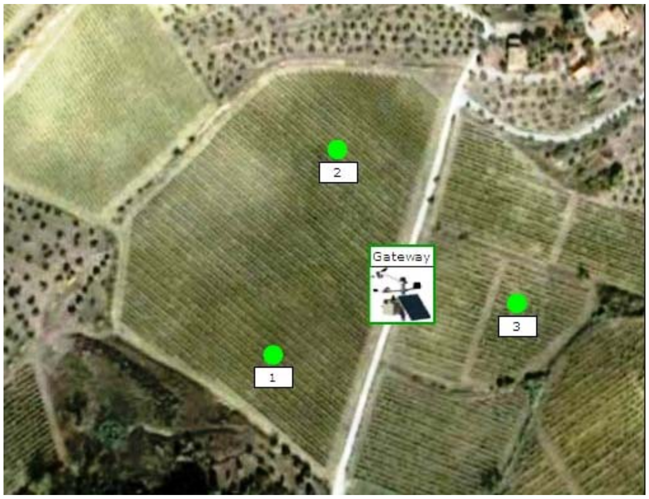


Figure 11. Distributed Wireless Nodes in a Vineyard.

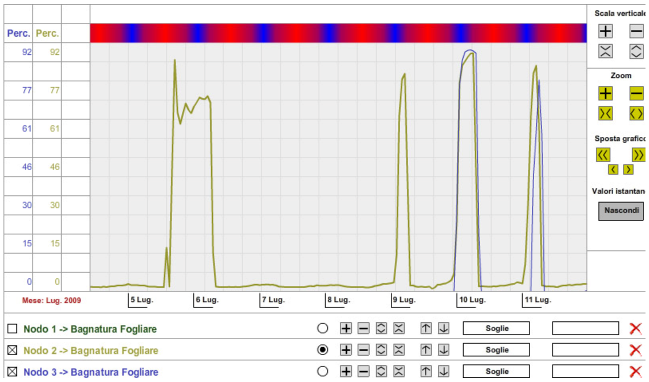


Figure 12. Different Leaf Wetness Conditions in a Small Vineyard.

VII. Conclusion

This paper deals with the design, optimization and development of a practical solution for application to the agro-food chain monitoring and control. The overall system was addressed in terms of the experienced platform, network issues related both to communication protocols between nodes and gateway operations up to the suitable remote user interface. Every constitutive element of the system chain was described in detail in order to point out the features and the remarkable advantages in terms of complexity reduction and usability.

To highlight the effectiveness and accurateness of the developed system, several case studies were presented. Moreover, the encouraging and unprecedented results achieved by this approach and supported by several pilot sites into different vineyard in Italy and France were shown.

The fruitful experience of some pilot sites was gathered by a new Italian company, Netsens, founded as a spin off of the University of Florence. Netsens has designed a new monitoring system called VineSense based on WSN technology and oriented towards market and user applications. In order to point out the improvements of the new solution respect to the experimental one, the main features of VineSense were described. Moreover, some important agronomic results achieved by the use of VineSense in different scenarios were sketched out, thus emphasizing the positive effects of the WSN technology in the agricultural environment.

Nowadays, the application of the solution described in this paper is under investigation to the more general field of environmental monitoring, due to its flexibility, scalability, adaptability and self-reconfigurability.

Acknowledgment

The authors would like to thank the ”GoodFood”' project partners and Netsens S.r.l. for their supporting in this research.

References

[1] L. Bencini, F. Chiti, G. Collodi, D. Di Palma, R. Fantacci, A. Manes, and G. Manes, *Agricultural Monitoring based on Wireless Sensor Network Technology: Real Long Life Deployments for Physiology and Pathogens Control*, in Proc. of SENSORCOMM 2009, pp. 372-377, June 2009.

[2] I. Akyldiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci, *A Survey on Sensor Networks*, IEEE Comm. Mag., vol. 40, pp. 102-114, August 2002.

[3] J. Al-Karaki and A. Kamal, *Routing Techniques in Wireless Sensor Networks: a Survey*, IEEE Comm. Mag, vol. 11, pp. 6-28, May 2004.

[4] K. Langendoen and G. Halkes, *Energy-Efficient Medium Access Control*, Delft University of Technology.

[5] S. Blackmore, *Precision Farming: An Introduction*, Outlook on Agriculture Journal, vol. 23, pp. 275-280, February 1994.

[6] N. Wang, N. Zhang, and M. Wang, *Wireless sensors in agriculture and food industry - Recent development and future perspective*, Computers and Electronics in Agriculture Journal, vol.50, pp. 114, January 2006.

[7] I.F. Akyildiz and W. Xudong, *A Survey on Wireless Mesh Networks*, IEEE Comm. Mag., vol. 43, pp. S23-S30, Settembre 2005.

[8] S. Cugati, W. Miller, and J. Schueller, *Automation concepts for the variable rate fertilizer applicator for tree farming*, in Proc. of 4^{th} European Conference in Precision Agriculture, June 2003.

[9] R. Evans and J. Bergman, *Relationships Between Cropping Sequences and Irrigation Frequency under Self-Propelled Irrigation Systems in the Northern Great Plains (Ngp)*, USDA Annual Report, December 2007.

[10] A. Flores, *Speeding up data delivery for precision agriculture*, Agriculture Research Magazine, June 2003.

[11] J. Mahan and D. Wanjura, 2004. *Upchurch, Design and Construction of a Wireless Infrared Thermometry System*, USDA Annual Report, May 2004.

[12] Institute Of Chemistry And Dynamics Of The Geosphere (ICG), *SoilNet: A Zigbee based soil moisture sensor network*, [avaiable: http://www.fz-juelich.de/icg/icg-4/index.php?index=739].

[13] *Mica2 Series*, Avaiable on: http://www.xbow.com

[14] V. Mattoli, A. Mondini, K.M. Razeeb, B. Oflynn, F. Murphy, S. Bellis, G. Collodi, A. Manes, P. Pennacchia, B. Mazzolai, and P. Dario, *Development of a Programmable Sensor Interface for Wireless Network Nodes for Intelligent Agricultural Applications*, in Proc. of IE 2005, pp. 1-6, June 2005.

[15] F. Chiti, M. Ciabatti, G. Collodi, D. Di Palma, and A. Manes, *An Embedded GPRS Gateway for Environmental Monitoring Wireless Sensor Networks*, in Proc. of EWSN 2006, February 2006.

[16] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11, 1999.

[17] A. El-Hoiydi, J. Decotignie, C. Enz, and E. Le Roux, *WiseMAC, an Ultra Low Power MAC Protocol for the WiseNET Wireless Sensor Network*, in Proc. of SENSYS 2003, vol. 1, pp. 244-251, November 2003.

[18] W. Ye, J. Heidemann, and D. Estrin, *An Energy-Efficient MAC Protocol for Wireless Sensor Networks*, in Proc. of INFOCOM 2002, vol. 3, pp. 1567-1576 June 2002.

[19] T. Dam and K. Langendoen, *An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks*, in Proc. of SENSYS 2003, pp. 171-180, November 2003.

[20] G. Lu, B. Krishnamachari, and C. Raghavendra, *Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Sensor Networks*, in Proc. of WMAN 2004, pp. 2440-2443, April 2004.

[21] S. Shakkottai, T. Rappaport, and P. Karlsson, *Cross-Layer Design for Wireless Networks*, IEEE Comm. Mag., vol. 41, pp. 74-80, October 2003.

Evaluation of Environmental Wireless Sensor Network - Case Foxhouse

Ismo Hakala, Jukka Ihalainen, Ilkka Kivelä
University Of Jyväskylä
Kokkola University Consortium Chydenius
P.O.Box 567, FI-67701, Kokkola, Finland
{ismo.hakala, jukka.ihalainen, ilkka.kivela}@chydenius.fi

Merja Tikkakoski
Veteli, Finland
merja.tikkakoski@pp.inet.fi

***Abstract*—Environmental monitoring in agriculture is an interesting and promising area of application for wireless sensor networks. Wireless sensor networks can deliver valuable information about environment, animals and their habitat. This paper describes a case where such sensing application was implemented by the authors together with biologists. The wireless sensor network collected data in hard outdoors conditions over a period of one year, during which luminosity, temperature and humidity were measured in a foxhouse. Evaluation of IEEE802.15.4 based communication used was one of the main subjects of the study. The throughput and the link quality statistics are presented and some factors related to link quality are analysed. In addition to reliability analysis, this paper describes the Foxhouse Case implementation, reports on its results, presents the power-consumption measurements and discusses the observations made during project.**

***Index Terms*—wsn implementation; environmental monitoring; reliability; link quality; power-consumption**

I. Introduction

Environmental monitoring, both indoors and outdoors, is one of the most promising application areas for wireless sensor networks. Compared to traditional sensing methods wireless sensor network (WSN) technology offers some important benefits: wide areas can be covered with inexpensive nodes, battery-powered devices with a self-configuration ability enable quick and easy system installation, the energy-efficiency of battery-powered devices makes long-term monitoring possible, and typically there is also a real time access to data. In addition, when monitoring animals, for example, human presence is unwanted and can distort the results or even cause damage to the subjects of monitoring. By using WSNs, measurements can be done without any other disturbance except that caused by the deployment of the network. Thus the results would also be free from any external impact.

In spite of the improvements in WSN technology there are not too many cases reported where WSN has been used in environmental monitoring. One such experiment was the Foxhouse case implemented by The Kokkola University Consortium Chydenius (later Chydenius) [1]. The other environmental wireless sensor networks have been built also, and valuable information about hardware and software has been obtained, see, e.g., [2], [3], [4], [5], [6], [7], and [8].

The Foxhouse project was undertaken jointly with the MTT Agrifood Research Finland. The wireless sensor network for environmental monitoring was implemented in the Fur Farming Research Station at Kannus. The reason for the WSN implementation was the need to get real time information about the habitat of foxes in a foxhouse. The amount of light received is presumed to be the key factor in stimulating reproduction of foxes, so measuring light intensity in different parts of the foxhouse was the focal point of interest. Measurement data for luminosity as well as temperature and humidity were gathered outdoors over a period of one year. We also observed the functionality and usability of the network, and some tools for network maintenance were developed during the project.

In addition to habitat monitoring we also wanted long-term information about WSNs in environmental monitoring as well as about the performance of wireless communication due to the IEEE802.15.4 standard. The amount of collected data was large in the Foxhouse project. A total of 1 707 758 received packets were stored in the database over a period of one year. This large database enabled us to evaluate the wireless sensor network's IEEE802.15.4 based communication. The performance of wireless communication is studied by analysing the throughput and the quality of links. The link quality is evaluated by using received signal strength indicator (RSSI) which indicates the strength of the radio signal at the receiver's position. Measurement data for temperature and humidity enabled analysis of the effects of weather conditions on link quality. Measurements to analyse the effect of angular orientation on link quality as well as to evaluate node's power consumption were performed also.

This paper is organized as follows. First we overview some related research where WSN has been used in agriculture and environmental monitoring. The Foxhouse case as a sensing application is described and the CiNet sensor network and node architectures are introduced. Resulting statistics about network reliability, like the throughput and the quality of links, as well as the application data are given. Some factors related to link quality are analysed and the power consumption measurements of a battery-powered sensing node are presented. Finally, the project experiences are discussed briefly.

II. Related Work

WSN implementations have been used and reported in monitoring tasks during the past few years. In agriculture there have been applications where WSNs have delivered valuable information about environment, e.g., about soil moisture and microclimate [2], [3], and [6]. Also animals and their habitat have been monitored. A famous implementation of this kind was created by the project in Great Duck Island, where some researchers from the University of California, Berkeley, together with biologists from the College of the Atlantic, built a sensor network and collected data from a seabird habitat [7]. In that project the researchers discussed also the need and possibilities for network status monitoring and retasking.

In the implemented sensor network projects, performance of wireless sensor network has been one of the main concerns. The impact of environmental conditions on the link performance has been studied. For example, distance, height and angular orientation of devices have been reported to affect the received signal strength sensitivity [9]. Also the effect of foliage and weather conditions on the propagation of radio waves has been studied in [5], [10] and [11]. Reliability of sensor networks includes more than just error-free wireless communication. For example, validity of data is important, and, when problems occur, fault detection is needed. A fault detection system was tested in the project where groundwater quality was measured [12] and [13].

Typically, monitoring testbeds have included some tens of nodes. In some cases a network may include a few hundreds of nodes. Trio Testbed was a large network with 557 solar-powered nodes [14]. That network was functioning for quite a long time. The researchers of Trio Testbed discussed maintenance issues as well as middleware and system software challenges in outdoor sensing systems.

When monitoring environment, animals or their habitat, pure technical knowledge about sensor network technology is usually not sufficient; the implementations require also knowledge about the ecosystem.

III. Foxhouse Case

Furbearing animals have been raised on farms from the 19th century. In the present housing system for foxes, rows of cages are placed in sheds. They provide a normal outside temperature while protecting against direct sunlight, wind and rain. In addition to the traditional sheds, also special halls have been tested as shelters. The goal of the scientific research has been to improve the health and welfare of animals as well as their productivity. From the economical point of view, successful breeding of animals is important. The amount of light received is presumed to be the key factor in stimulating reproduction. This has been studied in the Fur Farming Research Station at Kannus where, among other things, light intensity has been measured in different conditions in sheds and in a hall (foxhouse), and the results of fox breeding have then been compared [15].

Luminosity varies during the day and year. So the measurement should be more or less constant at least in the breeding season. Luminosity can also vary a lot in different parts of the foxhouse and many sensors are needed to cover the whole area. The measuring problem thus matched perfectly the potential for solution behind the idea of WSNs. By using the wireless sensor network the researchers could get real time data from the habitat all year round. Another reason for this experiment was to get practical experience about the reliability of communications in a CiNet network. The project was carried out cooperatively between Chydenius and MTT Agrifood Research Finland.

The Foxhouse project, funded by the Finnish Funding Agency for Technology and Innovations, started on April 11th 2006, and the wireless sensor network gathered data for over one year. A foxhouse is a large wooden building (approximately 15x75 meters in area) where fox cages are placed in four rows about 80 centimeters above the ground. The building has an asphalt floor and a part of the roof and the walls are made of transparent fiberglass. Luminosity in all areas of the foxhouse was the focal point of interest, but some of the nodes were equipped also with temperature and humidity sensors. There was no heating in the building.

The system architecture in the Foxhouse case is shown in Figure 1. The sensor nodes in the foxhouse send measurements to a sink, which is connected to a PC by a serial cable. All received information is stored into a SQL database in the PC. Data in the database can be browsed by a web application.

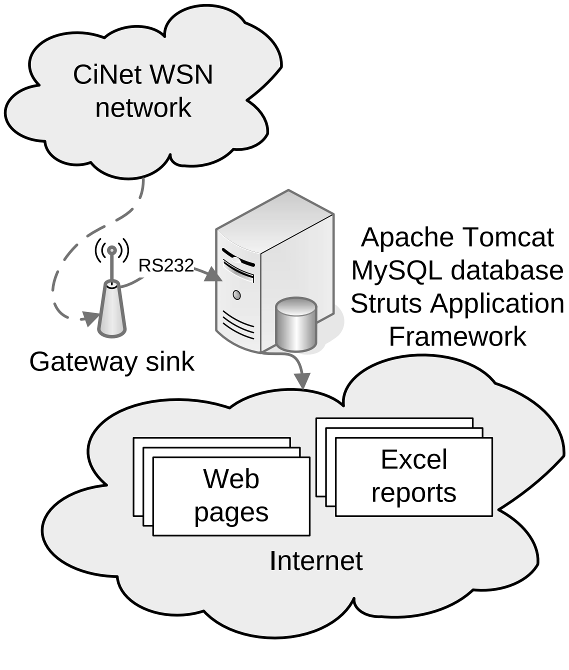


Figure 1. System architecture

Normally environmental monitoring networks require nodes with a battery supply. In this special case we could however use the main supply, because electricity was already available. This way we could also eliminate any effects of batteries' voltage variation on link quality. A battery powered node was included in the network for testing the real power consumption of the measuring nodes.

The nodes were attached about 1 meter above the metal cages as shown in Figure 2. Each wireless node was lightly encapsulated in a small plastic box while the sensors were left outside the box. The photodiode was shielded additionally with an aluminium tube.

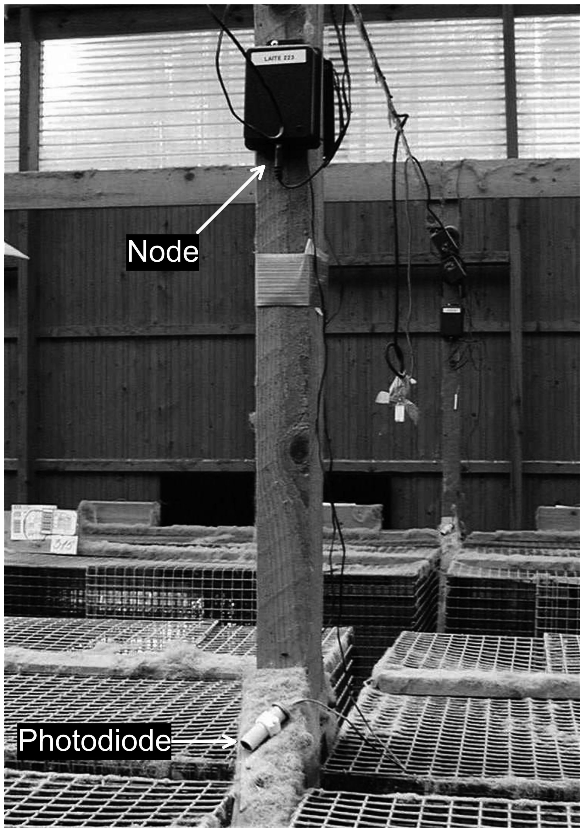
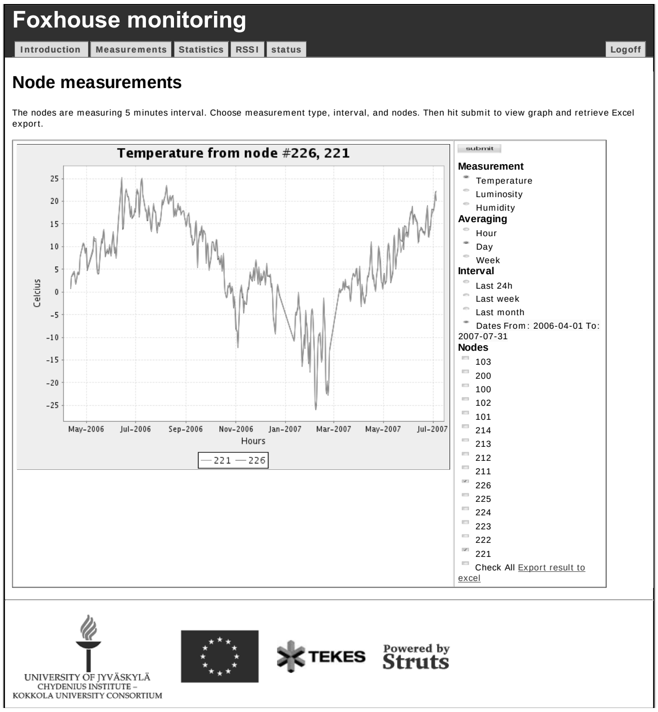


Figure 2. Node and photodiode placement

A. System Architecture

The system architecture is depicted in Figure 1. The wireless sensor network produces measurements data, which is delivered to the gateway (sink node). The sink node is connected to the server via a RS232 cable. The PC acts as a server in this system, running a Java program for reading packets from serial port. The Java program has the following functions: it reads data from the serial port, parses measurement data and adds timestamps, and finally stores the prepared data to a MySQL database. The Java program has no control functions, it only stores incoming packets. The system's database contains all information about the actual measurements (temperature, humidity, and luminosity) and also management data (RSSI, packet counts) for diagnostic purposes. The database includes basic information about nodes, nodes' location etc. A raw packet payload is also stored in the database.

The application is built as a 3-tier web application and it relies on freeware and open-source software. It consists of a Tomcat Java application server, Struts framework, and MySQL database server. The Tomcat Application server and Struts framework together act as a web server and the MySQL database server provides data storage.



Node measurements

The nodes are measuring 5 minutes interval. Choose measurement type, interval, and nodes. Then hit submit to view graph and retrieve Excel export.

Measurement
● Temperature
○ Luminosity
○ Humidity
Averaging
○ Hour
● Day
○ Week
Interval
○ Last 24h
○ Last week
○ Last month
● Dates From: 2006-04-01 To: 2007-07-31
Nodes
☐ 103
☐ 200
☐ 100
☐ 102
☐ 101
☐ 214
☐ 213
☐ 212
☐ 211
☑ 226
☐ 225
☐ 224
☐ 223
☐ 222
☑ 221
☐ Check All Export result to excel

Figure 3. User interface in the Foxhouse monitoring system.

The user interface of the web application enables browsing and reporting of stored measurements. Data can be viewed as a graph on a web page or Excel report. An example of the user interface can be seen in Figure 3, in which the temperature from the 1st May 2006 until the 30th July 2007 is displayed. The measurement period can be defined by start and end dates. The period is also possible to choose from pre-entered values for day, week, or month. Also averaging is done by day, week, or month. In addition to physical quantities also statistical information about measurements as well as communication statistics can be monitored on the web site.

B. Network

The network was build to collect as reliable information as possible from the habitat of animals. It was likely that luminosity values would vary inside the foxhouse and nodes were situated in places that were interesting from the researchers' point of view. Cluster topology appeared to be the most suitable topology for this purpose. The network included two kind of devices: sensing nodes (RFD) and routing nodes (FFD). Some of the routing nodes worked as cluster heads also. The wireless sensor network in the Foxhouse case consisted of 14 nodes in two clusters: the front cluster and the rear cluster. Node 102 was the cluster head of the rear cluster (nodes 211-213) and node 101 was the cluster head of the front cluster (nodes 221-225). The placement can be seen in Figure 4. Nodes were programmed beforehand with fixed routing table as well as ID numbers, since the platform does not support hardware MAC address.

In figure 4 the nodes with RFD labels are sensing nodes that are extremely energy-efficient. They used the same mea-

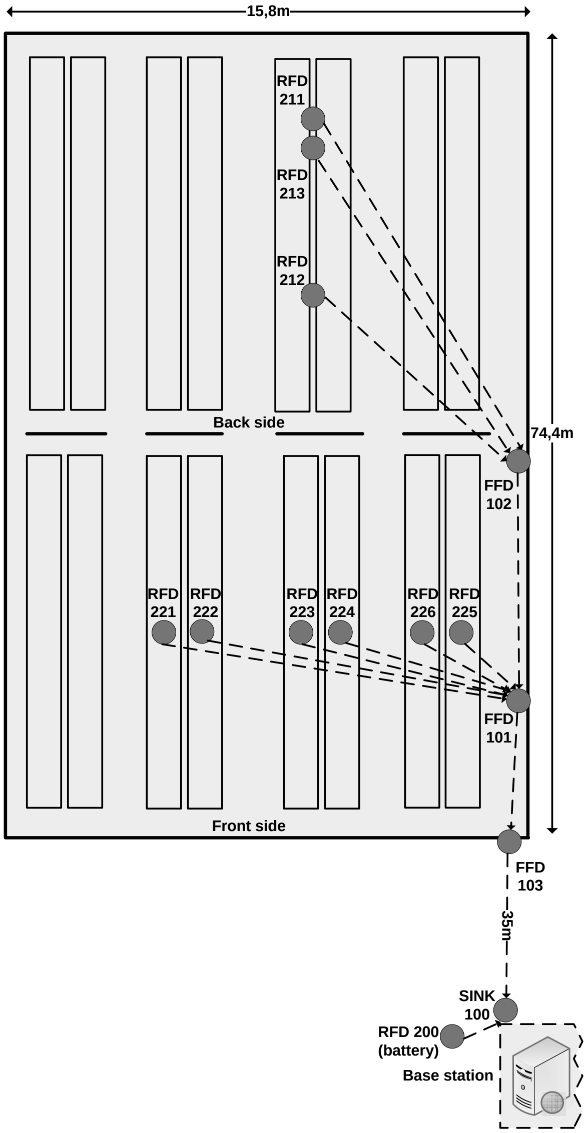


Figure 4. Node locations in the Foxhouse

surement interval and woke up in every 5 minutes to send measured values to cluster heads which forwarded packets through routing nodes to the sink. Rest of the time they were in deep sleep. Measuring in every 5 minutes produced more data than needed, but, because packet losses are likely in WSNs, a quite high measuring frequency was wanted. The current consumption of a sensing node during the measuring and transmitting period is explained in Subsection 4F.

The nodes with FFD labels are routing nodes, which take care of multi-hop communication and collect also statistical information about packet losses and link qualities. The network was not synchronized, and that is why routing nodes had to be active and listen to incoming packets all the time. Their energy consumption was higher than that of sensing nodes. Multi-hop communication was based on fixed routing tables, which means that each sensing node is sending packets to a sink via the same route, which minimizes control traffic. The solution was justified in this special case, because the network was both small and static and a predefinition of topology was possible.

The communication protocol used, IEEE 802.15.4, allows frames of 128 bytes, including the protocol overhead which is needed in every packet. Because all the messages have their starting cost, we aggregated management data cumulatively in each node, i.e., when a sensing node sends a message to a sink, each node in the communication path appends its own management data to the message. This method reduces the number of delivered packets. Thus, the light data aggregation used is cost-effective.

For studying the reliability of wireless communication two different communication methods were used. In the first phase (implementation and testing) from April 2006 to October 2006 all the communication links were unidirectional, so no acknowledgements were used. In the second phase (evaluation) from October 2006 to June 2007, acknowledgements and management data were required to ensure communication over link. In case of missing packets, there could be a maximum of three retransmissions.

C. Sensor Node

The nodes used in the Foxhouse network were CiNet nodes. CiNet is a research and development platform for wireless sensor network implemented by Chydenius. The hardware in the CiNet node is specially designed for WSNs and consists of inexpensive standard off-the-shelf components. The CiNet node includes all the basic components for wireless sensor networks. It has a microcontroller and a transceiver on board as well as one temperature sensor for testing purposes. In real monitoring more sensors are needed. These sensors can be placed on a special sensor board, which can be connected to the main board. The device is shown in Figure 5 and its architecture in Figure 6.

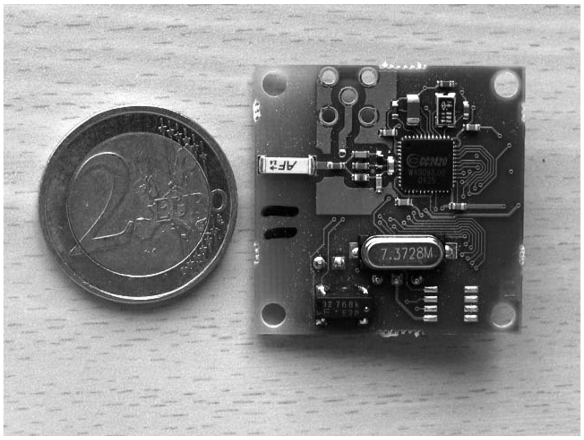
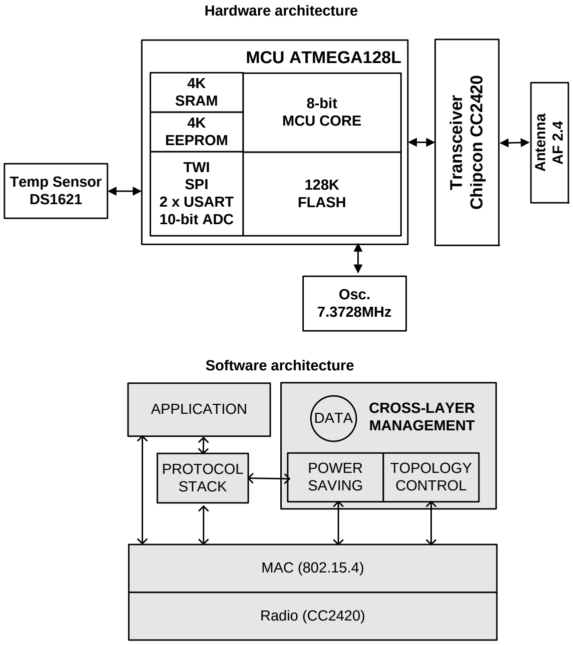


Figure 5. The CiNet main board

The processor in the CiNet main board is an inexpensive 8-bit controller ATmega128L. It is highly integrated, and as a low-power CMOS controller it is suitable for battery-powered


Hardware architecture

Software architecture

Figure 6. The CiNet software and hardware architectures

devices. It has good power saving features and enough internal memory. The AVR RISC architecture supports efficient C programming and is able to execute most instructions in a single clock cycle. The RF transceiver CC2420 is connected directly to the controller with an SPI connection.

In addition to the main board, a sensing node in the Foxhouse network included a sensor board. The sensor boards were equipped with photodiodes (Osram BPW21), and some of the nodes have also temperature/humidity sensors (Sensirion SHT7x). A measurement period could take a relatively long time, in the worst case 380ms. To save energy, the sensing nodes use a short duty cycle, i.e., they are in a sleep mode approximately 99.8% of time depending on the sensors connected.

The software in nodes was implemented according to the cross-layer architecture [16]. The cross-layer approach seemed to offer good performance in devices with reduced resources and it appeared to be a working solution. The software architecture used in a node is shown in Figure 6. The main idea in this model has been to implement the wireless sensor network's basic tasks, such as topology management and power saving functionalities, as separate protocols in a cross-layer management entity. These modules can control directly both the application and protocol stack. Data structures, which are in shared use, are also implemented in the cross-layer management entity. The biggest advantage of the cross-layer implementation is its reduced computational and memory requirements - not all the information need to be transmitted between the application interfaces and protocol layers.

Wireless communication between two neighbour nodes takes place according to the 802.15.4 protocol, and these layers are implemented in the RF transceiver. All other modules are implemented in the microcontroller. The application takes care of communication with sensors. If a node has several sensors connected, data from all used sensors is added to the same payload. The protocol stack routes packets according to the IP protocol by using routing table information. The modules in cross-layer management are in common use or control both the protocol stack and application. The power saving module controls the functionality of other modules. Topology control in the cross-layer management gathers some statistical information from the received packets. The measured statistics were link quality and number of received packets in each node. Link quality was estimated in practice by measuring the signal strength from each received packet. These statistics are presented in more detail in Chapter IV.

IV. Results

The amount of collected data in the Foxhouse project was large. A total of 1 707 758 received packets were stored in the database. Based on this material, the researchers were able to analyse the physical circumstances of the foxhouse, the reliability of the network, and the quality of links. The effects of weather conditions and angular orientation of devices on the link quality are analysed also as well as the power consumption of the battery-powered sensing node.

A. *Environmental Monitoring*

The WSN in the Foxhouse collected luminosity, temperature and humidity values for one year. Typically, luminosity measurements were done manually a few times each day. The WSN reduced the need for manual measurement and recording in the database and made constant real-time data available for the biologists of MTT Agrifood Research Finland. Moreover, the network increased the number of measurements manyfold and, therefore, the network gave more information about the physical circumstances in the foxhouse. The measurements were stored in a SQL database, and are thus easily available for further studies. In this respect the sensor network can be regarded as a working solution for environmental monitoring.

For the biologists the most important information was the luminosity inside the foxhouse. They were especially eager to know how light conditions varied in different parts of the foxhouse and also how luminosity values change during winter and spring. In Figure 7 there is an example of luminosity values in the foxhouse depicted graphically. From the figure it is easy to get an overall understanding of how luminosity begins to increase after winter and of the related differences between nodes. In the figure, luminosity is averaged by week, and a rising trend is clear. Differences between trends show that the amount of light varies in different parts of the foxhouse. By using other filters, more variations in daytime luminosity values can be observed. In Figure 8 luminosity values are averaged by day. Measurements are available also in the SQL database and in an Excel format. Similar statistics are also available about temperature and humidity conditions.

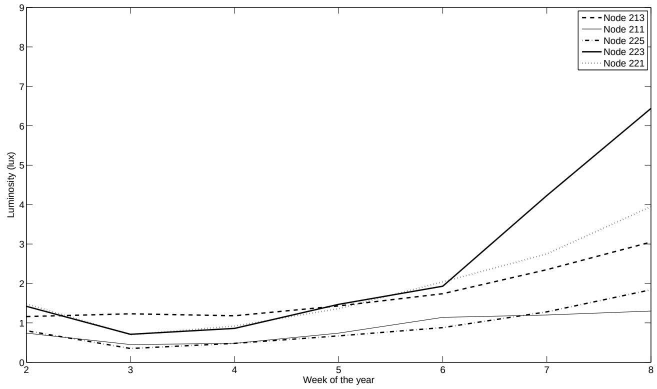


Figure 7. Averaged luminosity values during two months from the 11th of January in 2007.

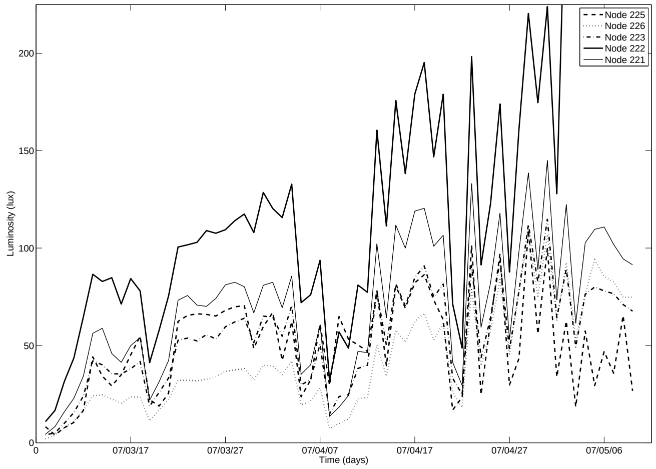


Figure 8. Development of averaged luminosity from March to June 2007.

B. Reliability of Communication

Reliability of IEEE 802.15.4 based wireless communication was one of the research subjects in this project. The plan was to relate it to the number of packet losses. Due to collisions and external disturbances, packet losses are possible.

Table I shows the received packets from the sensing nodes from the 12th of January until the 8th of February 2007. The throughput of the rear cluster was very good and that of the front cluster was almost 100%. The period was free from device failures and the throughput of the routing nodes 101, 102 and 103 was 100%. Thus the results in Table 1 show also the ratio of received packets to delivered packets at the cluster heads. Table II shows the longest continuous break and the number of consecutive missed packets for each sensing node during the break. The results of the front cluster were very good, at most only 8 consecutive packets were missed. Nodes 212 and 213, the sensing nodes of the rear cluster, had minor problems: node 212 missed 501 consecutive packets and node 213 missed 171 consecutive packets. This can be seen in the throughput statistics also. The short breaks of a single node had no effects on the environmental monitoring application. It can be concluded that wireless links were very reliable. The statistical calculations were based on 69 014 received packets.

Table I
THROUGHPUT STATISTIC BETWEEN 12TH JAN AND 8TH FEB

Node	Throughput %
Rear cluster	
211	99.85
212	93.42
213	97.81
Front cluster	
221	99.59
222	99.29
223	99.95
224	99.77
225	99.95
226	99.28

Table II
THE LONGEST BREAK AND THE NUMBER OF CONSECUTIVE MISSED PACKETS BETWEEN 12TH JAN AND 8TH FEB

Node	Packets	Break start		Break end	
Rear cluster	#	Date	Time	Date	Time
211	3	25.01	20:52	25.01	21:07
212	501	16.01	05:06	17.01	22:53
213	171	16.01	07:56	16.01	22:53
Front cluster					
221	7	24.01	04:27	24.01	05:02
222	8	02.02	08:04	02.02	08:44
223	3	13.01	09:05	13.01	09:20
224	6	24.01	05:32	24.01	06:02
225	3	02.02	08:04	02.02	08:44
226	8	02.02	06:59	02.02	07:39

Table III presents the corresponding figures as Table I from the 12th of January until the 1st of May 2007. The throughput of the nodes of the front cluster is good (excepting node 224), but not as good as in Table I. The reason is that the cluster head, node 101 in the front cluster, had a severe two weeks failure between the 24th of February and the 8th March. Similarly, the throughput of the rear cluster was not good and was basically due to two longer periods when node 102, the cluster head of the rear cluster, was dead and the communication from the rear cluster was disabled. The throughput of the rear cluster was also affected by the failure of node 101. The statistical calculation in Table III was based on 203 494 received packets.

The results show clearly that the network used could not guarantee reliable communication whenever there were problems in some critical nodes, like cluster heads. The poor results were basically due to device failures and the topology used. On the other hand, a failure of a single node is not necessarily a big problem in environmental monitoring. For example, a sensing node, node 224, broke during the spring 2007, and it shows in the statistics as a poor throughput. In any case, only the measurements of that node were lacking; the other nodes in the front cluster gave enough information for the needs of the environmental monitoring application. Thus, it can be concluded that the network of sensing nodes was dense enough for the application but the network of routing nodes was too sparse for reliable communication. Reliability of communication can easily be improved by increasing alternative communication paths to the sink.

Table III
THROUGHPUT STATISTIC BETWEEN 12TH JAN AND 1ST MAY

Node	Throughput %
Rear cluster	
211	58.14
212	57.81
213	59.76
Front cluster	
221	88.93
222	88.66
223	89.01
224	28.62
225	89.02
226	88.78

C. Link Quality

The significance of sufficient link quality came apparent at the beginning of the project when the network was installed. Distributing the nodes was difficult because of the unpredictable radio range. As it is well known, soil and other surfaces considerably affect the radio range by emitting signals and causing reflections. Also devices' angular orientation as well as variable weather conditions affect wireless communication. For that reason, radio range is difficult to predict beforehand. The average radio range inside the foxhouse was only 30-40m with the maximum transmission power of 0 dBm. Some nodes did not necessarily have a line of sight. Outside with the same transmission power the nodes have about 100m radio range.

Link quality can be evaluated by using the Received Signal Strength Indicator (RSSI), which indicates the strength of the radio signal at the receiver's position. The observed good throughput statistics indicate that there are no problems with the quality of links. However, during the project the received RSSI values varied significantly in time. An example of the received RSSI variation is shown in Figure 9.

The noticed RSSI variation forced us to ensure an adequate signal level. We added some functionalities which helped us to monitor the quality of links. The cluster heads measured the strength of the received signal from each sensing node and included this information to the end of data frame. The received RSSI values appeared to be especially useful when distributing nodes. Finding the positions for each node in a static network was easy and quick when the quality of link was known.

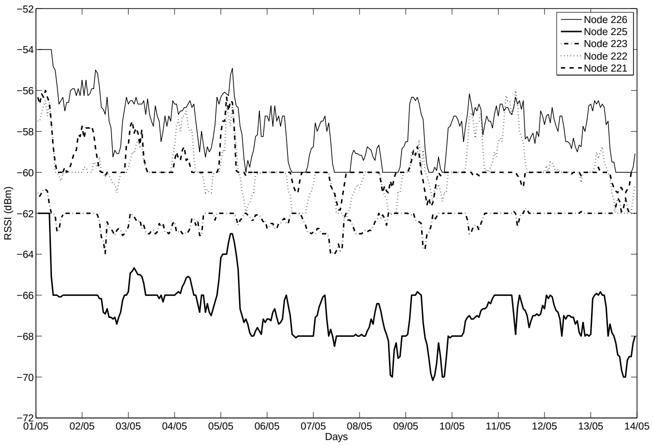


Figure 9. RSSI of the front cluster between 1st May and 15th May.

Tables IV and V present the statistics of RSSI values from the 12th of January until the 8th of February 2007. For each node the average of RSSI is above -80dBm and the standard deviation of RSSI varies between 0.76dBm and 2.90dBm. From Table IV it can be seen that the RSSI value of a single node can vary as much as 18dBm. However, Tables IV and V show that on average the link quality of nodes is good apart from the sensing node 211, which had the RSSI value below -75dBm for most of time. The minimum RSSI value was -83dBm, which in our experience can indicate unreliable communication. On the other hand, the throughput between sensing node 211 and sink was very good, which implies that the link quality was good enough for reliable communication. In spite of significant variation of the received RSSI values it can be concluded that the link qualities of the sensing nodes were very good on average.

Table IV
LINK QUALITY STATISTIC BETWEEN 12TH JAN AND 8TH FEB

Node	Mean RSSI	Std RSSI	Min RSSI	Max RSSI
Rear cluster	dBm	dBm	dBm	dBm
211	-77.2	1.7	-83	-73
212	-69.5	2.9	-75	-62
213	-71.1	1.0	-73	-68
Front cluster	dBm	dBm	dBm	dBm
221	-59.9	1.6	-64	-55
222	-65.4	2.5	-78	-60
223	-54.9	1.9	-63	-54
224	-55.1	1.0	-57	-53
225	-64.4	1.0	-67	-61
226	-54.5	0.8	-58	-53

D. Effects of Angular Orientation on Link Quality

During the project there seemed to be also variations between apparently similar devices; the link quality seemed to be better in some devices than others. Variations of the distance

Table V
RELATIVE FREQUENCY DISTRIBUTION FOR RSSI BETWEEN 12TH JAN AND 8TH FEB

Node	Relative frequency %		
Rear cluster	[−83, −75] dBm	(−75, −65] dBm	(−65, −57] dBm
211	85.7	14.3	0
212	0	89.8	10.2
213	0	100	0
Front cluster	[−83, −75] dBm	(−75, −65] dBm	(−65, −57] dBm
221	0	0	100
222	0.1	54.1	45.8
223	0	0	100
224	0	0	100
225	0	13.3	86.7
226	0	0	100

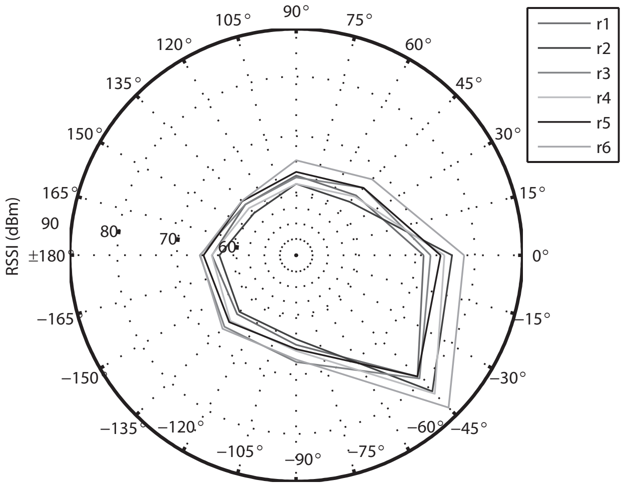


Figure 10. Variation of RSSI with angular orientation

between nodes could not alone explain the variations of the link quality. The angle of a device was also significant. Minor changes in positions were able to destroy the communication link between two devices. The effect of antenna's angular orientation has been reported in [9] also.

We designed the measurement procedure such as to obtain knowledge about how the received RSSI depends on a device and angular orientation. The measurements were done inside a large football hall in spring 2006. One transmitter device and six similar receiver devices were used. The transmitter and each receiver were attached to one-meter high wooden poles. The distance between the transmitter and a receiver was 4 meters. The transmitter was transmitting packets at -10dBm while in 8 different orientations (0, 45, 90, 135, 180, 225, 270, and 315 degrees) The transmitter was configured to send 20 packets per transmission period. When calculating the mean of RSSI, the maximum and minimum values were dropped in order to eliminate possible outliers. Figure 10 shows that there exist variations of RSSI values between different devices, but the impact of the angular orientation on the received RSSI is clearly more significant. In the worst case, the effect of the angular orientation on the received RSSI can be over 20dBm. Thus, the angular orientation clearly affects the received RSSI.

E. Effects of Weather Conditions on Link Quality

The weather conditions in the foxhouse were comparable to typical Scandinavian outdoor weather conditions. The lowest temperature during the measurement period was −33.6 °C and the highest 32.4 °C. The relative humidity inside the Foxhouse varied between 14.3% and 93.1% . The temperature and the relative humidity outside the Foxhouse were approximately the same as inside. A very low temperature and a high relative humidity make the conditions demanding for WSN.

While the network seemed to work well one day some of the static links could some other day prove unreliable. The link quality statistics show that the standard deviation of RSSI could be as high as 2.9dBm and RSSI of a single node could vary as much as 18dBm. The observed anomalies in the link performance of a single node can be explained by multipath propagation and dynamic environmental factors, such as the presence of people, movements, and weather conditions. These factors can interfere with the radio signal propagation, varying the received RSSI. On the other hand, most of the nodes had the line of sight and there was very little changes in environmental factors except in weather conditions. Since the radio frequency of 2.4GHz is also the resonant frequency of water, variation in air's moisture content can interfere with the received RSSI. An environment with a high humidity tends to absorb more power from the radio signals than when the humidity level is lower.

Figure 11 shows the average of relative humidity and RSSI from the 12th of January until the 8th of February. The values are averaged each hour. The effect of the relative humidity on the RSSI values can be seen in Figure 11. When the relative humidity increases, RSSI values decrease. Similar negative correlated results have been observed by others as well [17] and [5]. On the other hand, opposite effects of relative humidity on the RSSI values have been observed by [10]. When the relative humidity increases, the received signal strength increases. We observed similar positively correlated results also. Figure 12 shows the average of relative humidity and RSSI from the 14th of June until the 30th of June. The values are averaged each hour. The figure shows that the RSSI values are clearly positively correlated with relative humidity. The results are contradictory to each other in this matter.

Relative humidity is dependent on the temperature. Figures 13 and 14 shows temperature and relative humidity from the 12th of January until the 8th of February and from the 14th of June until the 30th of June, respectively. The values are averaged each hour. When the temperature changes, the relative humidity will change. However, as in the case of RSSI, the Figures 13 and 14 show clearly that relative humidity

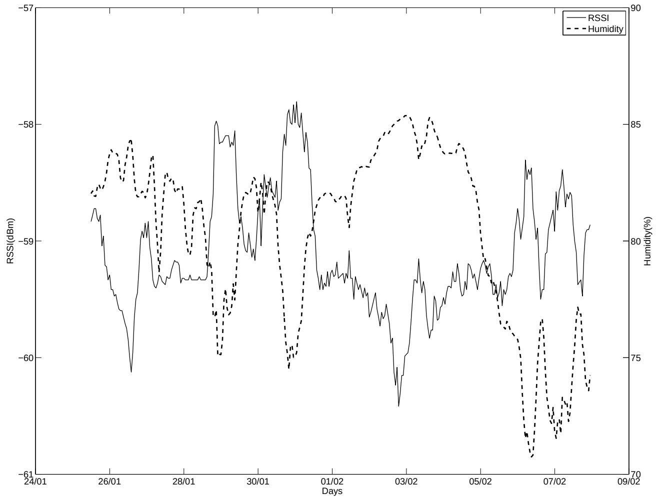


Figure 11. Relative humidity and RSSI between 12th Jan and 8th Feb

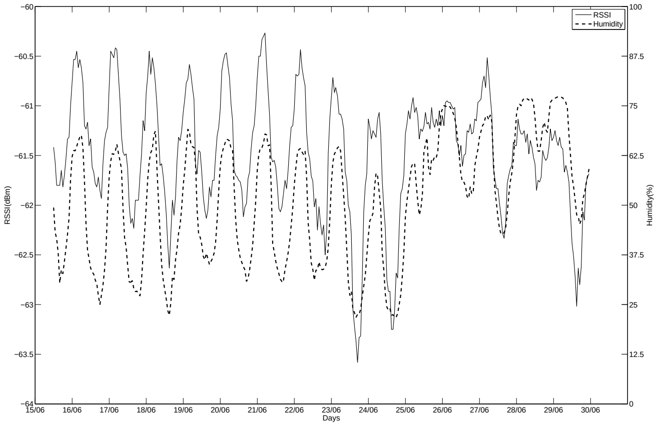


Figure 12. Relative humidity and RSSI between 14th June and 30th June

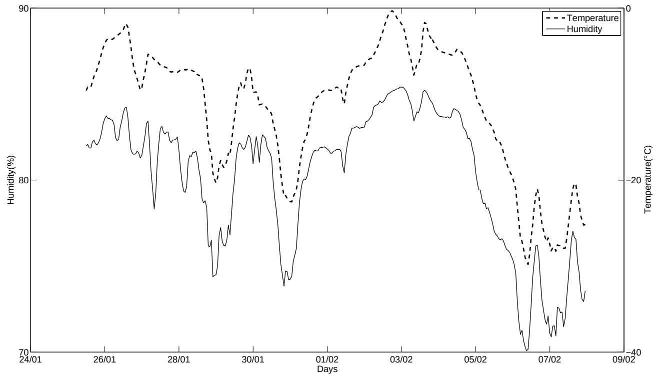


Figure 13. Relative humidity and temperature between 12th Jan and 8th Feb

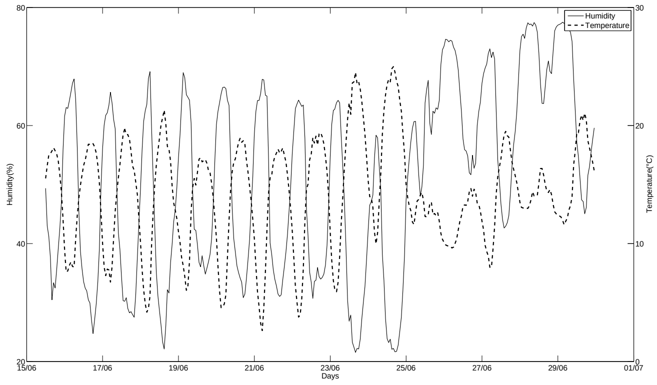


Figure 14. Relative humidity and temperature between 14th June and 30th June

correlates both positively and negatively with temperature. On the other hand, relative humidity and temperature are associated with the dew point. If the temperature T and the relative humidity R are given the dew point T_d can be approximated by

$$T_d \approx \frac{bf(T,R)}{a - f(T,R)},$$

where

$$\begin{aligned} f(T,R) &= \frac{aT}{b+T}\ln R, \\ a &= 17.27, \\ b &= 237.7\,^\circ\text{C}. \end{aligned}$$

When the dew point remains constant and temperature increases, relative humidity will decrease. At a given barometric pressure, independent of temperature, the dew point indicates the mole fraction of water vapor in the air, and therefore determines the specific humidity of the air. Thus, the dew point can be a better ”absolute” measure of the air’s moisture content than relative humidity.

Figures 15 and 16 show the dew point and relative humidity from the 12th January until the 8th of February and from the 14th of June until the 30th of June, respectively. The values are averaged each hour and the dew points are calculated by using the above equation. Both Figure 15 and Figure 16 show that RSSI values are negatively correlated with the dew point. We can conclude that the main part of the observed variation of the received RSSI of a single static node can be explained by the variation of weather conditions.

F. Power Consumption of Sensing Nodes

Normally environmental monitoring networks require battery-powered nodes. In this special case we could, however, use the main supply. In spite of main supply, the protocols of the used nodes supported energy-efficiency. The power consumption of a sensing node was measured by adding a 8.2Ω resistor next to the power supply. We measured the voltage over the resistor and calculated the current using the Ohm’s law. Similar methods have been used in [18] and [19]. In order to indicate the different phases, an output pin is triggered when there is a phase transition.

A sensing node works periodically and each cycle has two main modes: a sleep mode and a work mode. During a sleep mode the radio module is turned off and the microcontroller wakes up every second which takes at most 5ms and the power

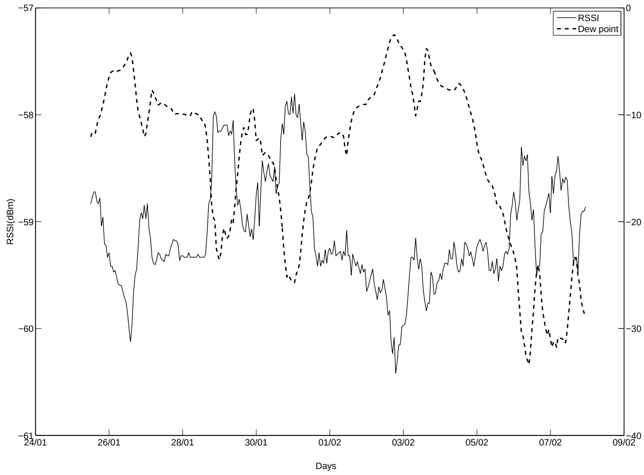


Figure 15. Dew point and RSSI between 12th Jan and 8th Feb

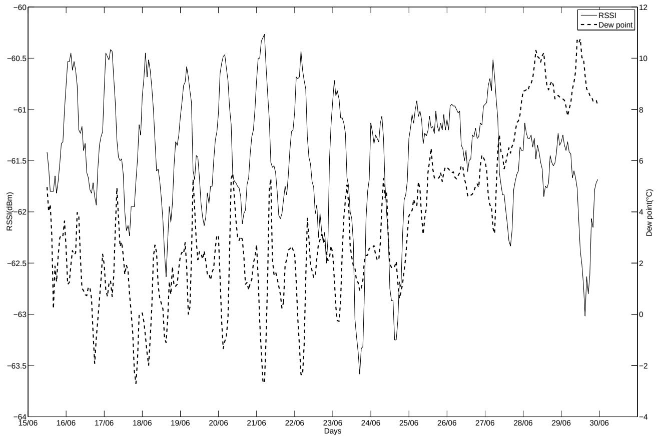


Figure 16. Dew point and RSSI between 14th June and 30th June

consumptions stays below $6mA$. The entire sleep mode takes 300s and the power consumption is below $30\mu A$ on average.

The work mode can be divided into 5 phases, which are depicted in Figure 17. The phases t_1, t_2, and t_3 are the temperature, relative humidity, and luminosity measuring phases which take approximately $216ms$, $62ms$, and $102ms$, respectively. The power consumption is approximately $12.4mA$. The phase t_4 is the message preparing phase which takes about $81ms$, and the power consumption is about $13.2mA$. During the phases $t_1 - t_4$ the microcontroller is turned on and the radio module is turned off. In the phase t_5, the transmitting phase, both radio module and microcontroller are turned on, and the duration of this phase depends on the number of retransmissions. In case of missing acknowledgements, there are at most three retransmissions. The phase takes at most $95ms$, and the power consumption stays below $30.6mA$. The entire work mode takes at most $556ms$, and the power consumption is $15.6mA$ on average.

The duty cycle of the sensing node is very short. Sensing nodes are in a sleep mode about 99.8% of the time. In average the power consumption of the entire cycle is at most $59\mu A$.

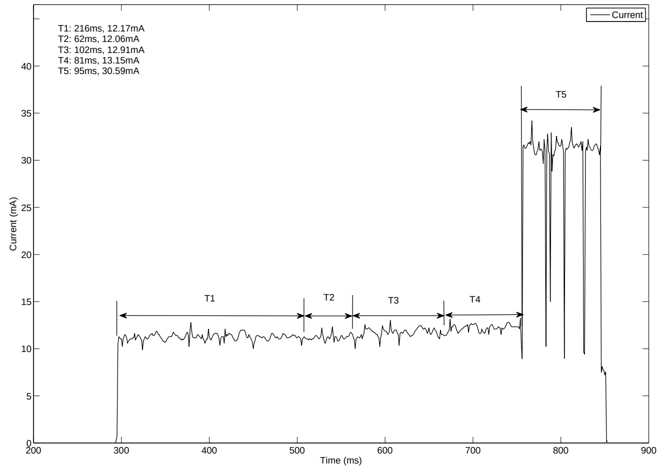


Figure 17. Power consumption of a sensing node

In the project a typical AA-type battery with voltage of $3.6V$ and capacity of $1000mAh$ was used. It can support the sensing node $1000mAh \cdot 1/0.064mA \approx 17014$ hours, which is about 708 days.

During the project battery duration in a low temperature was tested also. A battery-powered node measured temperature, humidity and luminosity during 10 months from October 2006 onwards. The lowest temperature during the measurement period was $-33.6\,°\mathrm{C}$ and the highest $32.4\,°\mathrm{C}$. The node function was 100% reliable in all weather conditions. The node was still working in July 2007 when the test period ended. Thus, the test of battery powered node supports the above calculations.

V. Discussion

Although the application in the Foxhouse case was quite small, we believe that it is actually quite a typical sensing network with its typical problems and features. The implemented network met the requirements. The biologists from MTT Agrifood Research Finland got desired information about habitat of animals, and the network reduced the need for manual measurements.

From the network developers' viewpoint the Foxhouse case was twofold. The tested IEEE802.15.4 wireless communication links proved very reliable in hard outdoors conditions. However, during the project the received RSSI of the static nodes could vary significantly in time, which could be explained by the variation of weather conditions. In spite of some variations of the received RSSI, the quality of links proved to be very good on average. Also the battery-powered node worked fine in spite of cold weather and the protocols proved to support energy-efficiency. On the other hand, the sensor nodes appeared to have some problems and the cluster topology could not guarantee reliable communication whenever there were problems in cluster heads. Alternative communication paths would have been needed. Also, it is important to recognize possible communication problems soon after malfunctioning.

During the project we added some functionalities which helped us to monitor network status. Information about quality of links together with information of successful packet delivery gave an idea about the network status and also helped to evaluate possible reasons for malfunctioning. It is however obvious that, in the future, network management in the CiNet network needs more attention, and we are going to focus our work on this question. Improvement of both diagnostics and reconfiguration is essential when thinking of usability and reliability of the CiNet network.

VI. Conclusion

The environmental monitoring system in the Foxhouse case proved that WSN using the IEEE802.15.4 communication protocol is reliable. The quality of links proved to be good on average in spite of some observed anomalies in link performance. The project showed also the technology used is relatively easy to implement in an environmental monitoring application. The use of WSN made constant real-time data available for biologists, and it also reduced manual measurements. To that extent, the WSN in the Foxhouse case was successful. There were nevertheless problems in functionalities of some routing nodes, which together with the fixed topology caused unnecessary packet loss. Reliability of communication can be improved by using a dynamic routing protocol. Design and implementation of a dynamic routing protocol will be future work. The Foxhouse case made it clear that IEEE802.15.4 based communication is suitable for environmental monitoring applications, but more attention must be paid to network management issues in the future.

References

[1] I. Hakala, M.Tikkakoski, and I. Kivelä, "Wireless sensor network in environmental monitoring - case foxhouse," in *Proceedings of the Second International Conference on Sensor Technologies and Applications (SENSORCOMM 2008)*, Cap Esterel, France, August 25-31 2008.

[2] R. Beckwith, D. Teibel, and P. Bowen, "Results from an agricultural wireless sensor network," in *IEEE 1st Workshop on Embedded Networked Sensors*, Tampa, Florida, USA, 2004.

[3] J. Burrell, T. Brooke, and R. Beckwith, "Vineyard computing: Sensor networks in agricultural production," *IEEE Pervasive Computing*, vol. 3, no. 1, 2004.

[4] R. Cardell-Oliver, M. Kranz, K. Smettem, and K. Mayer, "A reactive soil moisture sensor network: Design and field evaluation," *International Journal of Distributed Sensor Networks*, vol. 1, pp. 149–162, April 2005.

[5] P. K. Haneveld, "Evading murphy: A sensor network deployment in precision agriculture," Tech Report, 2007.

[6] K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture," in *Proc. 14th Intl. Workshop on Parallel and Distributed Real-Time Systems (WPDRTS)*, Rhodes, April 2006.

[7] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, 2002, pp. 88–97.

[8] K. Martinez, J. Hart, and R. Ong, "Environmental sensor networks," *IEEE Computer*, vol. 37, no. 8, pp. 50–56, 2004.

[9] D. Lymberopoulos, Q. Lindsey, and A. Savvides, "An empirical characterization of radio signal strength variability in 3-d ieee 802.15.4 networks using monopole antennas," in *European Workshop on Wireless Sensor Networks, EWSN'06*, Zurich, Switzerland, February 13-15 2006.

[10] J. Thelen, D. Goense, and K. Langendoen, "Radio wave propagation in potato fields," in *Proceedings of the First workshop on Wireless Network Measurements (co-located with WiOpt 2005)*, Riva del Garda, Italy, April 2005.

[11] M. Hebel, R. Tate, and D. Watson, "Results of wireless sensor network transceiver testing for agricultural applications," *ASABE Paper No. 073077. St.Joseph, Mich.:ASABE*, 2006.

[12] N. Ramanathan, E. Kohler, and D. Estrin, "Towards a debugging system for sensor networks," *International Journal for Network Management*, 2005.

[13] N. Ramanathan, L. Balzano, M. Burt, D. Estrin, E. Kohler, T. Harmon, C. Harvey, J. Jay, S. Rothenberg, and M. Srivastava, "Rapid deployment with confidence: Calibration and fault detection in environmental sensor networks," CENS, Tech Report, 2006.

[14] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler, "Trio: Enabling sustainable and scalable outdoor wireless sensor network deployments," in *the Fifth International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors*, 2006.

[15] H. T. Korhonen, T. Rekilä, and T. Kivinen, "Siniketun kasvatus halliolosuhteissa," *Turkistalous*, no. 8, 2006.

[16] I. Hakala and M. Tikkakoski, "From vertical to horizontal architecture - a cross-layer implementation in a sensor network node," in *InterSense 2006, the First International Conference on Integrated Internet Ad hoc and Sensor Networks*, Nice, France, May 30-31 2006.

[17] Y. Chen, J. Chiang, H. Chu, P. Huang, and A. Tsui, "Sensor-assisted wi-fi indoor location system for adapting to environmental dynamics," in *Proceedings of the 8th ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM'05*, Montreal, Quebec, Canada, October 10-13 2005.

[18] E. Erdogan, S. Ozev, and L. Collins, "Online snr detection for dynamic power management in wireless ad-hoc networks," *Research in Microelectronics and Electronics*, no. 22, 2008.

[19] B. Hohlt, L. Doherty, and E. Brewer, "Flexible power scheduling for sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, IPSN 2004*, Berkeley, California, Usa, April 26-27 2004.

Infrared wireless network sensors for imminent forest fire detection

Ignacio Bosch Roig
Institute of Telecommunications and Multimedia Applications (iTEAM)
Universidad Politécnica de Valencia
Valencia, Spain
igbosroi@dcom.upv.es

Luis Vergara Domínguez
Institute of Telecommunications and Multimedia Applications (iTEAM)
Universidad Politécnica de Valencia
Valencia, Spain
lvergara@dcom.upv.es

***Abstract* - This paper presents an automatic forest fire surveillance ground system applied to early fire detection. Sensor wireless network scheme allows not only supervising remotely wide-forest area, but also to detect immediately any fire threat. A real system for forest fire detection in actual operation is presented. The system is based on advanced thermal image processing techniques with the purpose of determining the presence of fire. The projected system performs the fusion of different detectors, which exploit different expected characteristics of a real fire, like persistence and increase in time. Theoretical results, practical simulations and results in a real environment are presented to corroborate the control of the probability of false alarm (PFA) and to evaluate the probability of detection (PD) dependence on signal to noise ratio (SNR). Delays of the system for alarm detection of controlled fire have been also evaluated. And finally, temporary evolution of false and true alarms is presented to evaluate the long-term performance in a real environment.**

Keywords - Infrared sensors, image processing, fire detection

I. INTRODUCTION

Video surveillance by means of sensor technologies is of great interest when monitoring wide forest areas. Video sensors and particularly thermal sensors [1] constitute a powerful tool to automate and complement human capabilities in wide-area surveillance tasks, since human capabilities are limited, among others, by the decrease of vision perception and by the range of coverage area. There is no need to comment on the importance of this type of systems for the protection and conservation of natural spaces [2]

Studies on forest fire detection and different contributions have been developed in the past. They have in general, several practice limitations. We can cite, among others, satellite based applications [3], but they are time limited when continuous surveillance over the same area is required [4]. Visible light cameras applications [5] but they are limited by the decreasing of vision perception at night [6]. And infrared based systems [7], [8] and [9] but in this kind of systems, signal processing algorithms are calibrated in a rather empirical manner, without relating to optimum filtering or detection theory.

So an automatic forest fire surveillance wireless system based on infrared sensors is presented at this paper in which we have introduced some significant innovations regarding classical infrared based systems. Basically, the algorithms are selected inside the framework of optimum statistical signal processing theory. Optimality is in the sense of the Neyman-Pearson criterion: given a selected Probability of False Alarm (*PFA*), try to maximize the Probability of Detection (*PD*). In our scheme, we have total control of the *PFA* because we have knowledge of the statistical distribution of every statistic involved in it. Also the *PD* is maximized by appropriate selection of different energy detectors, which are known to be optimum under very general models of the background noise and of the unknown signal to be detected. Therefore, the proposed system provides total control of a tolerable level of false alarms and has maximum sensitivity to the presence of an uncontrolled fire for the defined false-alarm rate. In the end, this implies a greater detection range and greater reliability of the overall system.

In this paper, we focus on how to process the images captured from each infrared sensor with simulated and real experiences. In Section 2, a description of the global system is presented. In Section 3 a scheme for infrared image processing is presented. In Section 4 the theoretical basis is corroborated with practical simulations. In Section 5 a prototype system and some real data results are presented, and finally, some conclusion about the operation of the system will be also offered.

II. GLOBAL SYSTEM

The implemented system has two types of stations; several local stations strategically located to render a required coverage and a central station, see Figure 1. The local stations have three basic stages: a first stage of thermal and visible image capture, a second stage of image processing, and a third stage of communications.

Since the system has been designed to work in wide forest areas and the required number of sensors and its location depend on the concrete scenario, it is important to choose a scalable technology system. For all these reasons, we have used an infrared sensor network based on a wireless communication link in order to collect all the sensor information in a central station.

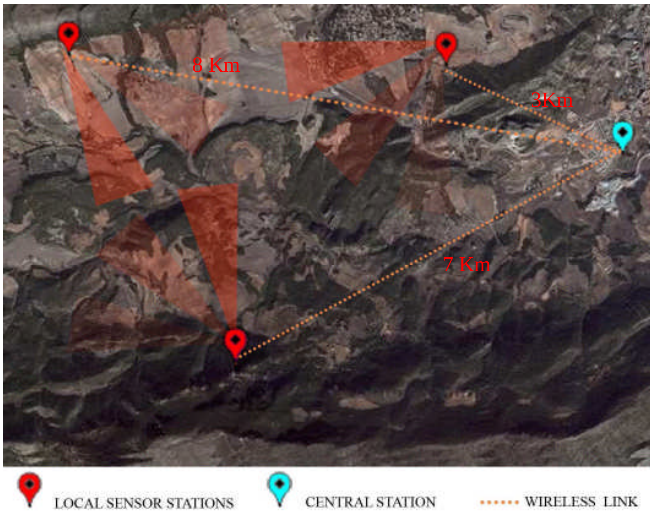


Figure 1. Global System diagram.

The local sensor station main functions are data acquisition (from each of the sensor belonging to the monitoring network) and real time image processing.

An acquisition card captures the images from the IR sensor. The captured images will be the input data to the image-processing step. Furthermore, another camera in the video range is used to provide visible access to the area under surveillance. If an alarm is detected, the system captures both thermal image and a visible one, which only contributes to information if there is sufficient diurnal light. In spite of the automatic operation of the system, both cameras can be remotely controlled in manual mode in order to change system parameters or to verify the detected fire threats. See Figure 2.

The thermal cameras which are used can be low cost and without the ability to show temperature values, they do not require the specific values of temperature, but rather the relative increments of infrared energy in every pixel with respect to some properly defined reference level.

Moreover, the proposed system indirectly uses physical properties of the environment (such as temperature) to reduce false alarms. In the calibration stage, the system adjusts the sensitivity based on numerical parameters such as level and span, which are supplied by the infrared camera. These have a direct correspondence with physical temperature values.

The thermal images captured from each sensor are processed in real time to meet temporal restrictions and to minimize the information flow as discussed below in section 3. Therefore, the communication unit performs the management for a correct transmission of this basic information.

For example, if a possible fire is detected, an alarm must be sent to the central station indicating the geographic location of the fire in minimum time and optimizing the communication resources. To avoid an excessive flow of information, only alarm threat coordinates, together with complementary time-space information, are sent from the local station to the central station.

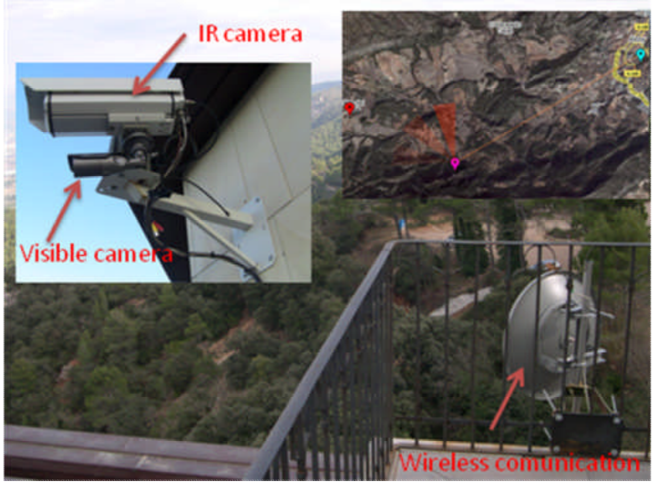


Figure 2. Real global system with infrared/visible cameras and wireless comunication unit

In addition, an optimized system of alarm information compression has been implemented to transmit only the contour of the form of the object described by the alarms, when they are numerous and close to each other.

The wireless link manages the information transmission between each sensor of the network and the central station. The central station represents fire threats in a zone map by means of a Geographic Information System (GIS), or any other equivalent form for locating alarms in a given area. The central station also performs other functions, such us management of historic alarm data-bases or remote connection to surveillance ports. In addition, all the control parameters of the system can be modified in real time by the central station through of communications unit. Finally, note that the system has control of the thresholds of each detector type, which are calculated from the desired *PFA* and from the sensitivity of the system, which depends on the level and the span obtained from the camera in each recalibration. This allows the thresholds to change based on the different seasons and the different weather conditions, such as morning, afternoon, night, summer, winter, etc.

III. PROCESSING THE IMAGES FROM INFRARED SENSORS

We focus on processing the thermal images from each local sensor, as described above. We perform a pixel to pixel processing. We turn each thermal image into a matrix of pixels, where each pixel is associated with a resolution cell corresponding to certain coordinates of rank and azimuth. In this way, we generate a vector that describes the time history of each resolution cell (see Figure 3).

A first approximation to the problem of automatic alarms detection could be to decide the presence of fire, in one resolution cell, when the energy level of the pixel under test reaches a certain threshold. If the statistic distribution of the noise is known, the threshold can be fitted to satisfy a desired probability of false alarm (PFA), getting a probability of detection (PD) that depends on the signal to noise ratio (SNR).

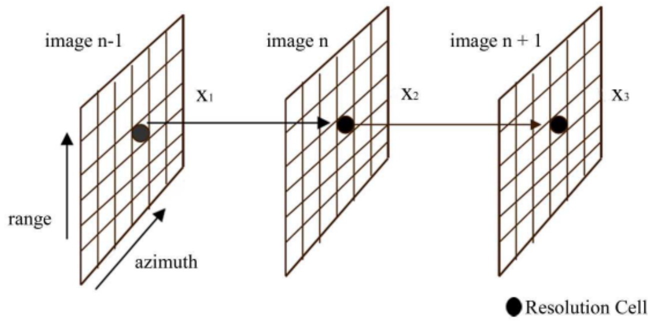


Figure 3. Temporal processing of resolution

On the other hand, usually, infrared noise registered from wide-area surveillance (registered values from usual environment conditions, without considering fires), is highly correlated, that is: the temperature among neighbor cells, and for the same cell along successive images, changes slowly. We can take advantage of this additional information about infrared background noise to improve the SNR using a noise predictor. The noise predictor will compute the noise level of the cell under test, through different scans. Then, this estimated level may be subtracted from the pixel under test, thus improving SNR. Note that if we improve the SNR we get a better PD for a given PFA.

The proposed ideas are applied in the detector scheme of Figure 4. The detector is applied to each resolution cell.

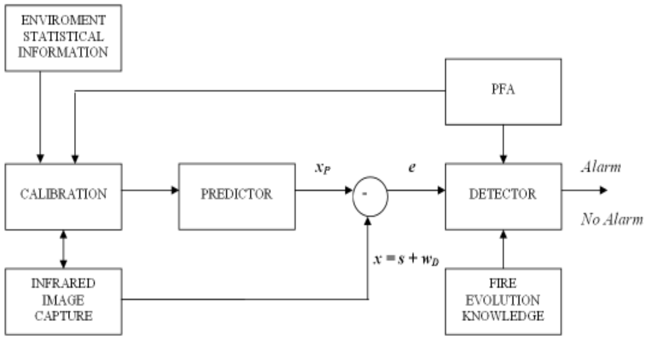


Figure 4. Uncontrolled fires detector general scheme

Partial aspects of the algorithms were previously presented. In [10] the design of an optimum linear predictor was considered. The input to the linear predictor was the values of the pixel under analysis recorded in previous scans. In [11] the linear predictor is extended to the non-gaussian case by including a nonlinear gain. Also in [10] a matched subspace detector is considered to suppress undesired alarms due to "occasional" effects like cars, people or wind. In [12] the matched subspace detector is complemented with an increase detector to take advantage of the expected increasing temperature behavior of an uncontrolled fire. Fusion of the decisions made by both detectors is also considered in [12]. Finally in [13] a complete detector scheme is proposed, which makes use of both detectors, but considering a different time-scale for everyone. Thus several matched subspace detectors corresponding to short-term intervals are fused to give the so-called persistence decision. Then a long-term increasing detector is implemented, to give the increase decision. Finally both decisions are fused.

In the system described in this paper we have basically implemented the detector scheme of [12] plus a linear predictor, which uses a reference image for prediction instead of the previous images as it was done in [10].

In the following we include a brief explanation of the algorithms. More specific details may be found in the mentioned references. A main objective is to have control of the overall PFA achieved; hence we present some new simulation to verify the capability of controlling the overall PFA in the complete detector system. And also we presented a dependence study of PFA and PD with the system parameters.

The input signal to the scheme is the vector $\mathbf{x} = \mathbf{s} + \mathbf{w_D}$ containing the level of one resolution cell through D consecutive image scans. So, $\mathbf{x}$ will always have a noise component $\mathbf{w_D}$ and when a true alarm is presented, the vector $\mathbf{x}$ will also contain the signal component $\mathbf{s}$ due to the presence of a fire. In order to improve the signal to noise ratio (SNR) of $\mathbf{x}$, a reference image is also computed. This image represents the infrared scene noise computed from N consecutive scans of the cell under test, assuming that there is no signal present at those scans. Then, we predict $\mathbf{w_D}$ from the reference image, using a predictor (details about the optimum design of the predictor could be seen at [10]). Subtracting the prediction $\mathbf{x_p}$ from the signal $\mathbf{x}$ we get vector $\mathbf{e}$ that defines the detector input with improved SNR.

An uncontrolled fire is supposed to be a fire that persists on time, increasing continuously their temperature, in contrast with other occasional effects like cars or atmospheric changes that may produce undesired alarms (see Figure 5).

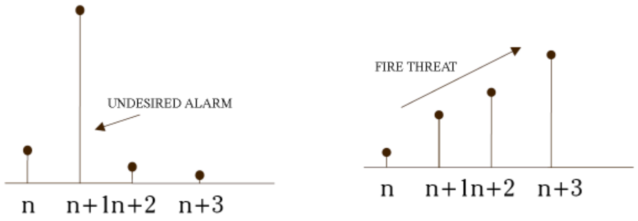


Figure 5. Uncontrolled fire vs undesired false alarm pattern

This consideration let us to propose the detector scheme of Figure 6. This scheme exploits the assumption about persistence and increase by means of decision fusion algorithms.

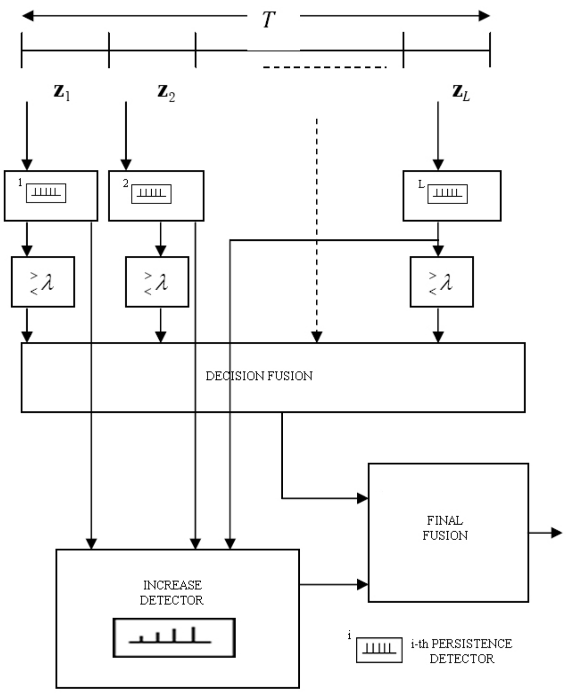


Figure 6. Detector Scheme

Persistence detector allows removing false alarms due to sporadic effects, which behaviour can be assumed to be "high pass". So the persistence detector is implemented by dividing vector **e** into L non-overlapped segments $\mathbf{z}_1$ **...** $\mathbf{z}_L$ assuming that the fire signature across each segment is inside a "low pass" subspace with projection matrix **P**. The energy E, inside the subspace **P**, of vector $\mathbf{z}_i$, is compared with a threshold λ that is calculated to meet a desired PFA_0, which is the probability of false alarm for each individual detector (which is equal for all of the chain) in the form of a classic subspace matched filter.

$$E_i = \mathbf{z}_i^T \mathbf{P} \mathbf{z}_i \gtrless \lambda \qquad (1)$$

To compute the false alarm probability after fusion decision, PFA_p, it is important to define an optimal fusion rule in order to combine the obtained decisions for each of the L segments or detectors of M elements. The optimal decision fusion, as proved in [13], is as follows:

$$R_{opt}(u) = \begin{cases} 1, & \text{, if the number of ones in } \mathbf{u} > nu; \\ 1 \text{ with probability } \gamma, & \text{if the number of ones in } \mathbf{u} = nu; \\ 0, & \text{, if the number of ones in } \mathbf{u} < nu; \end{cases} \qquad (2)$$

Where vector $\mathbf{u} = [u_1, \ldots, u_L]^T$ is the vector formed by the individual binary decisions and nu is the number of detectors, from the total number, that we require to meet the persistence criterion.

From this, we get the next expression for PFA_p and PD_p:

$$PFA_p = \sum_{k=nu+1}^{L} \binom{L}{K} PFA_0^k (1-PFA_0)^{L-k} + \gamma \binom{L}{nu} PFA_0^k (1-PFA_0)^{L-k} \qquad (3)$$

$$PD_P = \sum_{k=nu+1}^{L} \binom{L}{k} PD_0^k (1-PD_0)^{L-k} + \gamma \binom{L}{nu} PD_0^k (1-PD_0)^{L-k} \qquad (4)$$

To illustrate the performance of expressions 3 and 4 we have done a simple recreation where we select as input of the persistence detector a continuous level signal under a background noise.

As a result of this simulation we get the ROC curves depicted in Figure 7 and Figure 8.

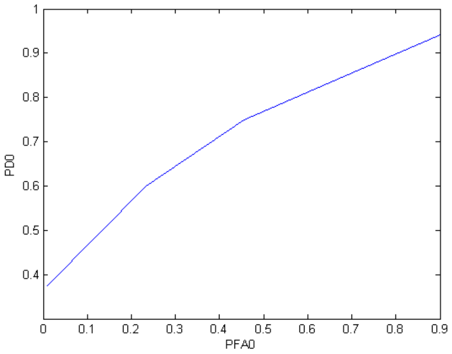


Figure 7. ROC curve PD_o vs PFA_o

The Figure 7 shows the evolution of PD_0 with PFA_0 (note that this curves are for a single detector). The PD_0 has been computed for a range of PFA's between 0.9 and 0.01 with SNR = 1dB and taking for the computation of each PD 1000 Gaussian and correlated vectors of 5 samples size, under a continuous level of signal that depends on the SNR. We can notice, as natural, that if we increase PFA_0 the PD_0 also increase.

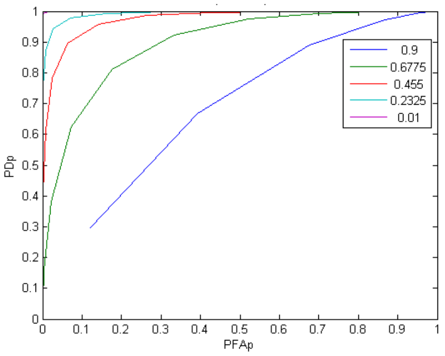


Figure 8. ROC curve: PD_p vs PFA_p with 5 PFA_o's

As regards the ROC curve for the PD_p vs PFA_p (see Figure 8. and comparing it with Figure 7) we can notice that, we get higher values of PD_p for the same PFA_0, and this is due to the fusion decisions.

On the other hand, long time increase detector will enhance the presence of increasing trends. The structure is based on a filter matched to a continuous component, preceded by a linear transformation $\mathbf{Q}^{(n)}$, which order depends on the speed of fire increasing . It is implemented by looking for increasing trends in the energy vector $\mathbf{z}_E = [E_1 ... E_L]^T$ where E_i was defined in (1). To this end, the vector $\mathbf{z}_E$ is transformed by a difference matrix of order *n*, $\mathbf{Q}^{(n)}$, and then matched to a continuous component vector of *L-n* elements $\mathbf{s}_n = [1…1]^T$. The corresponding test is given by [13],

$$\frac{\mathbf{z_E}^T \mathbf{Q}^{(n)T}\left(\mathbf{Q}^{(n)}\mathbf{Q}^{(n)T}\right)^{-1}\mathbf{s_n}}{\sqrt{2p\mathbf{s_n}^T\left(\mathbf{Q}^{(n)}\mathbf{Q}^{(n)T}\right)^{-1}\mathbf{s_n}}} \underset{<}{>} \lambda_0 \qquad (5)$$

where the difference matrix is defined by $\mathbf{Q}^{(n)} = \mathbf{Q}_{L-n+1} \cdot\cdot \mathbf{Q}_{L-n+1} \cdot \mathbf{Q}_L$ and

$$\mathbf{Q}_L = \begin{pmatrix} -1 & 1 & 0 & ... & 0 \\ 0 & -1 & 1 & ... & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & 0 & ... & -1 & 1 \end{pmatrix}_{L,L-1}$$

The threshold λ_0 is calculated to meet a desired *PFA* called PFA_i.

Once described the persistence and increase detectors, we define the final decision rule (final fusion in Figure 6). In a simple way we can formulate the final fusion rule as follows:

$$R(u) = \begin{cases} 1, \text{if } u = [1\,1] \\ 0, \text{otherwise} \end{cases} \qquad (6)$$

Where $\mathbf{u}=[u_p\ u_i]^T$ are the binary random variables that represents the persistence and increase decisions respectively. Although some correlation may exist between both decisions, in a first approximation we can formulate the total PFA and PD as:

$$PFA_t = PFA_p \cdot PFA_i \qquad (7)$$

$$PD_t = PD_p \cdot PD_i \qquad (8)$$

As presented in next section, some practical simulations have been carried out to corroborate the above equations (3), (4), (7) and (8).

IV. Practical simulations

Settled the theoretical basis and once the design of the system has been considered, some practical simulations are addressed, at this section, to show the correct operation of the proposed system.

Theoretical (continuous line) and experimental curves, obtained by simulation, are superimposed to verify the control on the required *PFA* by fitting parameters *L*, *nu* and PFA_0.

For the experimental curves, we suppose that the input vector is a correlated Gaussian vector of $\mathbf{T} = L \cdot M$ correlated samples taken in *L* segments of size *M* and fixing *M = 6* to simplify.

On the subject of the relation between PFA_p and *nu*, fitting *M*=6, *L*=20, and *nu* varying from 0 to 20 and PFA_0 from 0.5 to 0.01, we have, as expected, that for a fixed PFA_0, when *nu* increases the PFA_p decreases (see Figure 9). Moreover, if the PFA_0 decreases the curves shift to the left, that is, for the same value of *nu* we get a lower PFA_p.

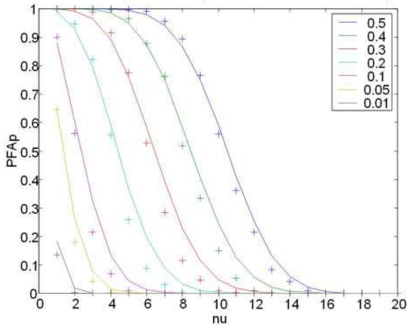


Figure 9. Therotical and practical PFA_p vs *nu* with *L=20* for different PFA_0 values

Regarding the relation between PFA_p and *L* we have fixed a value for *nu=6*, *L* varying from 6 to 20 and PFA_o from 0.5 to 0.01. As depicted in Figure 10. , if we increase the number of segments *L* we have more positive decisions, for a fixed *nu*. For this reason PFA_p increases with *L*.

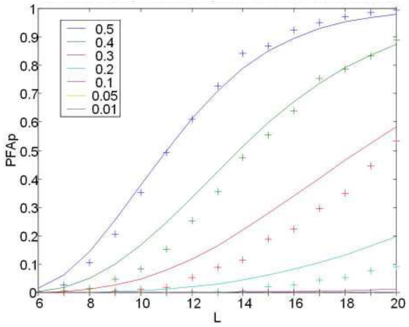


Figure 10. Theoretical and practical PFA_p vs *L* with *nu=6* for diffrerent PFA_0 values

In the case of the relation between PFA_p and PFA_o, we fixed *nu=6*, *L* varying from 6 to 18 and PFA_o from 0.5 to 0.01. Looking at Figure 11. , we can notice that PFA_p

increases with PFA_o. It is a natural result since the probability of false alarm of the fusion follows the same trend that the probability of false alarm of each independent detector.

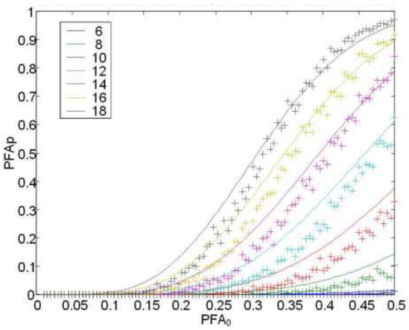


Figure 11. Theoretical and practical PFA_p vs PFA_o with $nu=6$ for different L values

An important conclusion to stress of these simulations is the similar response of theoretical and practical results, thus validating equation (3).

In the case of the dependence of PD_p on the involved parameters by simulation we only depicted practical curves since PD for each detector are known. For these curves we use the same Gaussian vector as input of the scheme, now generating a fire increasing along the T samples. To generate the simulated fire we use a linear model with increase slope, $crec$, the initial value x_{ini} will be recalculated depending on SNR and the noise variance var(x):

$$\mathbf{fire} = x_{ini} + crec \cdot \mathbf{v}' \qquad (9)$$

where $\mathbf{v}$ = [0, 1, 2, ..., T − 1], and

$$x_{ini} = \sqrt{10^{\frac{SNR}{10}}} \cdot \mathrm{var}(x) \text{ if } crec > 0 \text{ and } x_{fin} = x_{ini},$$

$$x_{ini} = x_{ini} - crec \cdot (T-1) \text{ if } crec < 0$$

The results were as follows:

Regarding the relation between PD_p and nu as we can see in Figure 12. , fitting L = 10, SNR = 1, and increase slope $crec$ = 0.01, nu varying from 0 to 20 and PFA_0 from 0.5 to 0.01, we have that when nu increases PD_p decreases. That is because when nu increase we have done a more restrictive fusion and we get lower probability of detection.

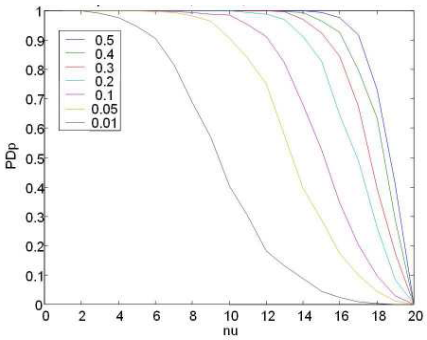


Figure 12. Practical PD_p vs nu with $L=20, SNR=1, crec=0.01$, for different PFA_0 values

Regarding the relation between PD_p and L (see Figure 13.), fitting nu = 6, SNR=1, and increase slope $crec$ = 0.01, L varying from 6 to 20 and PFA_0 from 0.5 to 0.01, we have that PD_p increases with L, because we have more local decisions to manage in the fusion decision.

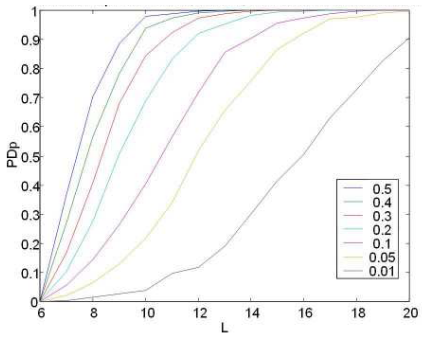


Figure 13. Practical PD_p vs L with $nu=6, SNR=1, crec=0.01$ for different $PFA0$ values

In the case of the relation between PD_p and PFA_0 we fixed nu = 6, SNR = 1, crec = 0.01, L varying from 6 to 18 and PFA_0 from 0.5 to 0.01. Looking at Figure 14. , we can notice that if PFA_0 increases PD_p also increases, note that PD_p increases quickly in comparison with the curve depicted for one detector (see Figure 7).

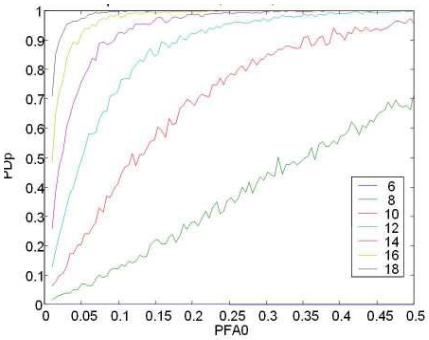


Figure 14. Practical PD_p vs PFA_o with $nu=6, SNR=1, crec=0.01$ for different L values

Regarding the relation between PD_p and *SNR* fitting $L = 10$, *nu*=6, and increase slope *crec* = 0.01, *SNR* varying from 0 to 20 and PFA_0 from 0.5 to 0.01 (see Figure 15.). If the *SNR* increases the *PD* increases, this is a natural result, if we increase the signal level we increase the probability of detecting.

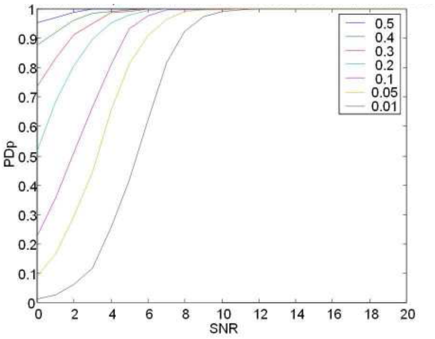


Figure 15. Practical PD_p vs *SNR* with *L=20, SNR=1, crec=0.01*, for different PFA_0 values

Another important conclusion to stress from these simulations is the importance of the decision fusion step to improve the *PD* of only one detector.

In the case of the increase detector, we have no theoretical expressions that connect the PFA_i and PD_i with the model parameters, so we cannot show the curves which compare the theoretical and practical results.

Finally we show the practical and simulated operation of the final decision fusion as a fusion of the persistence detector and the increase detector.

We have also represented the evolution of PFA_t with *L* in Figure 16. and with PFA_0 in the Figure 17. for a fixed value of *nu*.

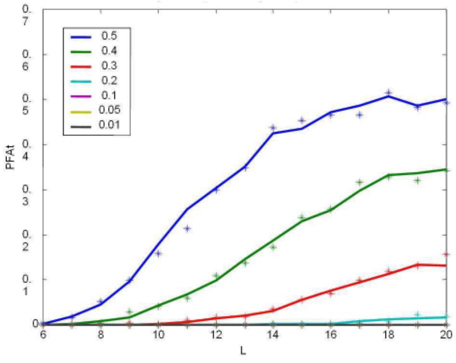


Figure 16. Theoretical and practical PFA_t vs *L* with different PFA_0 values

As expected, we can see in Figure 16. , as if we increase the value of segments *L* we have more positive decisions, for a fixed *nu*, and this increases the PFA_t.

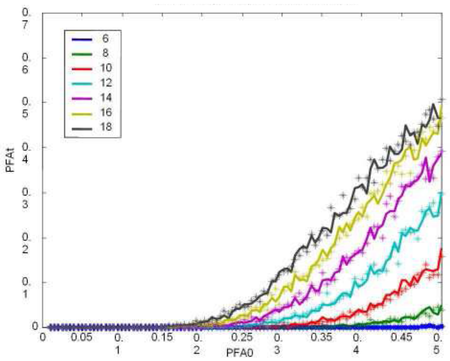


Figure 17. Theoretical and practical PFA_t vs PFA_0 for different L values

In the case of Figure 17. occurs something very similar, increasing the probability of false alarm of each independent detector PFA_0 increases the probability of false alarm of the fusion PFA_t and more significantly by increasing *L*.

In the same way, in Figures 18, 19 and 20 we have represented the evolution of PD_t with *L*, PFA_0, and *SNR* thus corroborating the concordance between the practical and theoretical results.

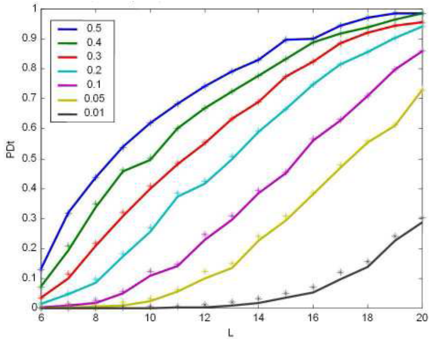


Figure 18. Theoretical and practical PD_t vs *L* with *SNR=1, crec=0.01* for different *PFA_0 values*

In Figure 18, if we increase the number of segments *L* we have more local decisions and this increases the PD_t .

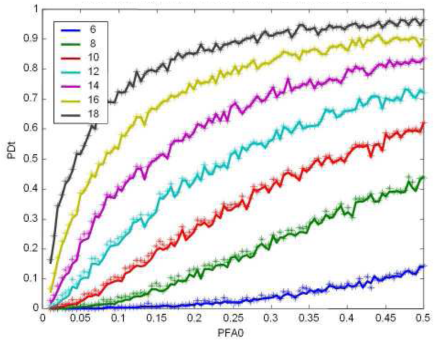


Figure 19. Theoretical and practical PD_t vs PFA_o with nu=6, *SNR=1, crec=0.01* for different *L values*

In Figure 19, if we increase the probability of false alarm of each independent detector PFA_0 the PD_t increases.

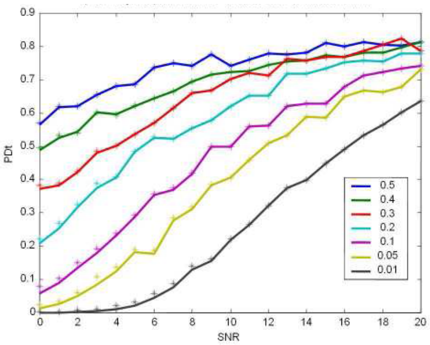


Figure 20. Theoretical and practical PD_t vs *SNR* with *L=10, crec=0.01* for different PFA_0 *values*

In Figure 20, if we increase the signal to noise ratio *SNR* the PD_t increases because we increase the probability of detecting.

Moreover, if we compare the continuous and dotted lines we can notice that the practical results are perfectly fitted to the theoretical ones.

Equations (7) and (8) also verify thus, indicating that the assumed uncorrelation between the increase and persistence detector is an appropriate hypothesis.

Another important issue to take into account is the capability of the system to make an early detection of fire. The curve depicted in Figure 21 have been computed for a fixed L=10, and PFA_0, PFA_i varying from 10^{-3} to 10^{-6} and show the delay that the system requires to generate the first fire alarms from the beginning of the fire.

We can conclude that for a fixed value of L, if *nu* is decreased L-nu increases and the persistence decision fusion is less restrictive and the fire can be detected earlier. For a fixed value of *L-nu*, if the PFA_0 is increased, the system response speed increases.

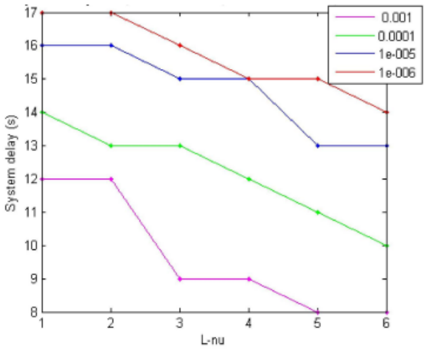


Figure 21. System delay vs L-nu, for fire detection with diferent PFA_0

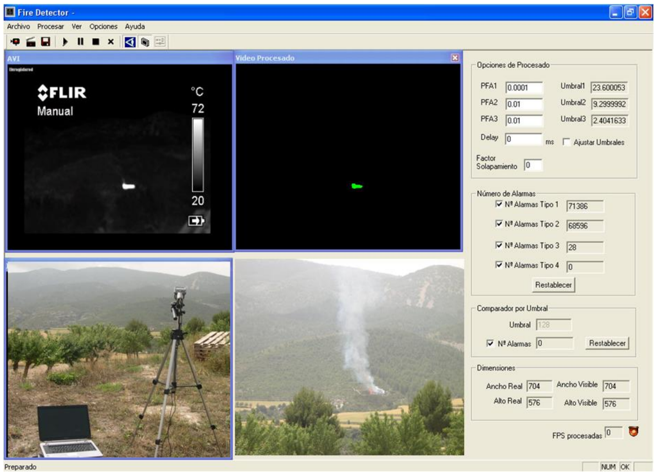
We can conclude that the system performs an earlier detection like tens of seconds, if we compare the system response with that of the average watchman.

V. RESULTS IN A REAL ENVIRONMENT

Some results in a real environment were shown to demonstrate the correct operation of the proposed system.

First, the system was tested under various real scenarios by performing the capture of infrared images from real area-controlled forest fires under the supervision of several firemen. Specifically in two tests with controlled fires made in Ayora and Alcoy parks.

The signal processing parameters were: L=10, *nu*=2, PFA_0 = PFA_i = 10^{-3}. One of the processing testing resut was shown in Figure 22. In this test we can see how the processing system is capable of discerning possible unwanted effects due to other hot zones detected by the thermal camera. In this example, the fire was detected at 1500 m of direct vision.



Opciones de Procesado

PFA1 0.0001 Umbral1 23.600053
PFA2 0.01 Umbral2 9.2999992
PFA3 0.01 Umbral3 2.4041633
Delay 0 ms
☐ Ajustar Umbrales
Factor Solapamiento 0

Número de Alarmas

☑ Nº Alarmas Tipo 1 71386
☑ Nº Alarmas Tipo 2 68596
☑ Nº Alarmas Tipo 3 28
☑ Nº Alarmas Tipo 4 0

Comparador por Umbral

Umbral 128
☑ Nº Alarmas 0

Dimensiones

Ancho Real 704 Ancho Visible 704
Alto Real 576 Alto Visible 576

FPS procesadas 0

Figure 22. *System tested in a real scenario*

Finally the system was installed in the natural park named "El Carrascal de la Font Roja" in the Spanish city of Alcoy. The Font Roja natural park area is about 2500 hectares, and its maximum altitude is 1352 m.

Specifically it was installed in the tower of the “Sanctuary of the Red Font” building, see Figure 23.

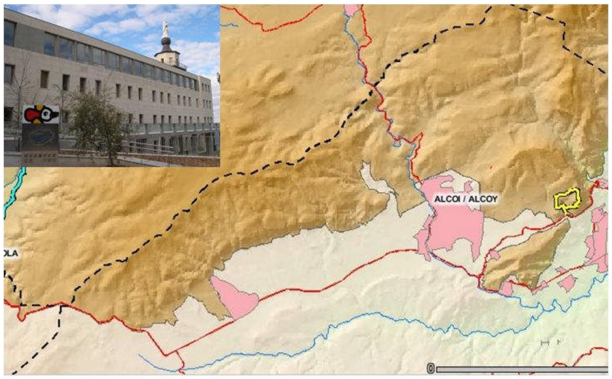


Figure 23. Natural park named “El Carrascal de la Font Roja”

During its normal operation, the system has generated true detections (see Figure 24.) and false alarms (see Figure 25.).

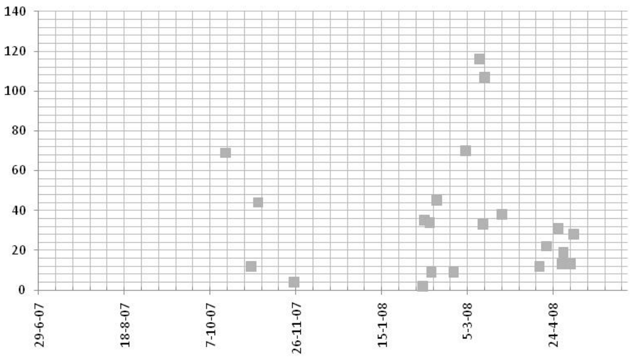


Figure 24. True alarms/day

A true detection is defined as an alarm that can be associated a posteriori to a true fire (whether or not the areas is under human control). On the other hand, a false alarm is one that cannot be associated to a real fire. And it is therefore a system error.

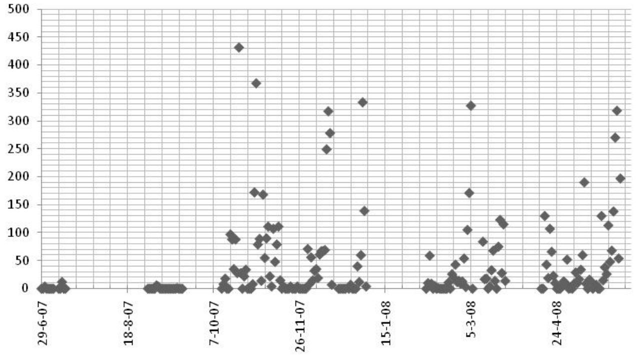


Figure 25. False alarms/day

A small average number of 40 false alarms per day were produced, which is in concordance with the configured value of PFA_t of approximately 8E-9.

Some possible explanations can be found for the false alarms. Many of them were produced during the night or in especially cool weather conditions in the presence of fog. Other false alarms were produced during the night by the lights of the city of Alcoy (part of the scanning angle included this city). Finally, some false alarms were observed on sunny days that had alternate periods of clouds.

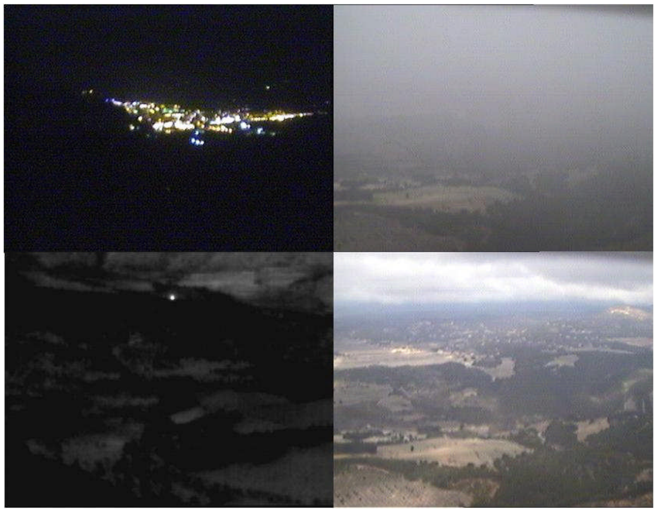


Figure 26. Thermal and visible images of false alarms examples

These results are due to the automatic calibration of the infrared camera. When the average infrared energy level is very low, the camera increases the sensitivity, so more false alarms are to be expected.

Appropriate control of the automatic calibration of the camera by means of a temperature sensor could avoid most of these false alarms. Then, if these problems are resolved, this amount of false alarms will be reduced drastically, theoretically getting average values of 14 false alarms per day or a PFA_t of 3E-9.

On the other hand the system has generated many true fire alarms (see examples in Figure 27.).

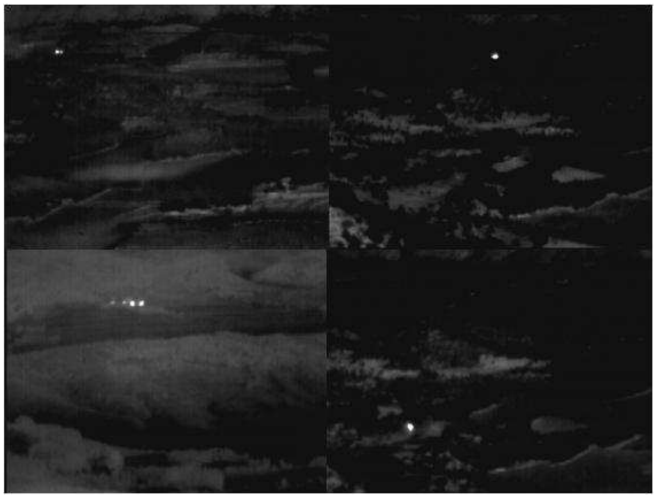


Figure 27. Thermal images of detection examples

Most of the true fires were detected from a significant distance approximately 1 to 10 km. To our knowledge, no fires went undetected during this period.

VI. CONCLUSIONS

A system for early forest fire detection based on images obtained from an infrared sensor network, has been presented. The system is based on the use of several sensors strategically located to render a required coverage.

We have focus on processing the images captured from the infrared sensors. Particularly interesting is the fusion of two decisions respectively corresponding to persistence and an increase detector. In essence both detectors try to reproduce the behavior of a human watchman: an uncontrolled fire must persist at short-term, and must increase at long-term.

The correct operation of the system has been justified verifying that the final *PFA* may be controlled by appropriate selection of the detector parameters. We have also evaluated that the theoretical *PD* for the total fusion decisions is consistent with practical simulations, and we can stress from here the advantage of decision fusion.

The functionality of the system has been verified in diverse controlled real tests and in normal operation in order to authenticate the accuracy of the proposed system. We have shown the temporary evolution of false alarms and true detections to evaluate the performance of the system in the real environment. We have also evaluated the delay of the system in order to generate an alarm corresponding to a real fire.

The obtained results show large potential interest for this system to replace human surveillance. Furthermore, and once it has been justified the correct operation of each single sensor, it is important to emphasize the importance of a network sensor in order to increase the coverage area of the system. [14]

ACKNOWLEDGMENT

This work has been supported by Generalitat Valenciana, under grant GVEMP06/001.

REFERENCES

[1] I. Bosch, S. Gómez, and L. Vergara, “Automatic forest surveillance based on infrared sensors”, Proc. 2007 International Conference on Sensor Technologies and Applications, (SENSORCOMM 2007), pp 572-577. 0-7695-2988-7/07

[2] E. Pastor, L. Zárate, E. Planas, and J. Arnaldos, “Mathemaical models and calculation systems for the study of wildland fire behaviour”. Prog Energy Combust Sci, 29, pp 139 – 152. 2003.

[3] Y. Rauste, E. Herland, H. Frelander, K. Soini, T. Kuoremaki, and A. Ruokari “Satellite-based forest fire detection for fire control in boreal forests”. International Journal of Remote Sensing, Vol. 18, no. 12, pp. 2641 - 2656, 1997.

[4] M.J. Carlotto, “Detection and analysis of change in remotely sensed imagery with applications to wide area surveillance”, IEEE Trans. Image Procees. Volume 6, no 1, pp. 189-202, 1997.

[5] Jin Li, Qingwen Qi, Xiuping Zou, Hu Peng, Lili Jiang, and Yajuan Liang, “Technique for automatic forest fire surveillance using visible ligth image”. Proc. on the IGARSS ´05. Volume 5, pp. 3135 - 3138, July 2005

[6] Y. Dedeoglu, B. Toreyin, U. Güdkübay, A. Enis, and C. Etin, “Real-time fire and flame detection in video”. IEEE 30th International Conference on Acustics, Speech and Signal Processing (ICASSP), pp: 669-672. 2005

[7] B.C. Arrue, A. Ollero, and J.R. Matinez de Dios, “An intelligent system for false alarm reduction in infrared forestfire detection”. IEEE Intelligent Systems. Volume: 15, no 3, pp. 64-73, 2000.

[8] J. Vicente and P. Guillemant, “An image processing technique for automatically detecting forest fire”, Int J. Therm. Sci, 41, pp: 1113-1120, 2002.

[9] S. Briz, J.A. De Castro, J.M. Aranda, J. Meléndez, and F. López, “Reduction of false alarm rate in automatic forest fire infrared surveillance systems”, Remote sensing of Envionment, 86, pp: 19-29. 2003

[10] L.Vergara and P.Bernabeu, “Automatic Signal Detection applied to fire control by infrared digital signal processing”. Signal Processing. Volume 80, pp. 659-669, 2000.

[11] L.Vergara and P.Bernabeu, “Simple approach to nonlinear prediction” Electronic Letters. Volume 37, no 14, pp. 926-928. July 2001.

[12] P.Bernabeu, L.Vergara, I.Bosch, and J.Igual, “A prediction/detection scheme for automatic forest fire surveillance”. Digital Signald Processing. Volume 14, pp. 481-507, 2004.

[13] I. Bosch and L. Vergara, “Forest Fire Detection by Infrared Data Processing”. Data Fusion for Situation Monitoring, Incident Detection and Response Management, 198 NATO Science Series: Computer & Systems Sciences. Editor E. Shahbazian, G. Rovina, P. Valin, IOS Press, Volume 6, pp. 931-944, 2006.

[14] J. Lloret, M. Garcia, D. Bri, and S. Sendra, “A Wireless Sensor Network Deployment for Rural and Forest Fire Detection and Verification”, Sensors, Vol. 9, Issue 11, Pp. 8722-8747. October 2009.

A new Wireless Sensor for Intravenous Dripping Detection

Paul Bustamante[1,2], Gonzalo Solas[1], Karol Grandez[1], Unai Bilbao
Electronics & Communications Department
[1]CEIT and [2]Tecnun (University of Navarra)
Manuel Lardizábal 15, San Sebastian, Spain
pbustamante@ceit.es, gsolas@ceit.es, kgrandez@ceit.es

***Abstract*— Nowadays the use of wireless technologies provides a great benefit to the society. This article shows a wireless sensor network for Intravenous Dripping System, which can detect when an intravenous liquid, provided to patients in hospitals, run out, as well as detecting obstructions in the catheter. This way, the attention in sanitary centers is more efficient and immediate, as the observation of the state of the container will not need human supervision. Also a novel algorithm has been simulated in order to improve the network with its own mobility.**

Keywords- Wireless Sensor, Dripping, Network, Device, low power consupmtion.

I. Introduction

The reduction of costs that the wireless technology is experiencing has incremented the number of applications which are able to implement this technology, in which medicine is one of the fields and is also known as "eHealth". This has caused many companies to dedicate their research into the development of wireless sensor platforms for generic use, which entails having complex operating systems, meaning a high consumption of energy.

The sensor network we are proposing has been developed focusing on the application, the detection of the intravenous dripping, for which the cost and the energy consumption have been optimized.

This sensor network is formed by different wireless devices, which are placed in the drip chamber of the patients, and by a central device located in every room of the sanitary center. The wireless device consists of four modules: the sensor module, the radio module, the feeding module and the microprocessor.

It is possible to emphasize that this article includes the study of the dripping detection techniques, as well as the development and implementation of the sensor and the base station. In this moment, the project is focused on the implementation of the location and routing algorithms of the sensor network.

A. State of art

There are different techniques for dripping detection; therefore a practical study on the effectiveness of these techniques has been carried out, in order to detect the dripping within the intravenous drip chamber. This will allow us to choose the most suitable method for our sensor.

As a reference of the different detection methods, several patents dealing with all the ways of dripping detection have been consulted and tested. These methods can be divided in the following sections: ultrasounds[1][2], piezoelectric materials [3], applying capacitive electrodes [4] and optical methods [5][6][7], which can be described follow.

1) Ultrasonic method

This dripping detection method consists of the detection of the tension variation produced by a drop that crosses a beam of ultrasounds. The ultrasound emitter generates a wave at a frequency of 25 KHz, which is detected by the receiver. The effect produced by the crossing of a drop between the emitter and receiver has been analyzed. The scheme of the assembly is shown in Figure 1.

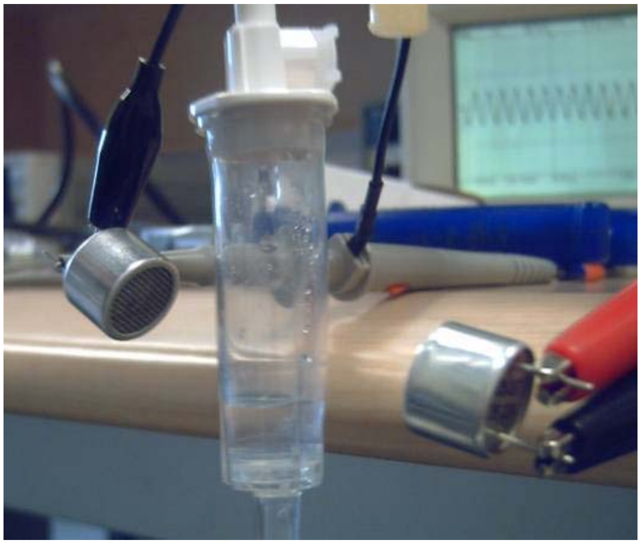


Figure 1 Ultrasonic method

It has been observed that the amplitude of the tension in the receiver varies depending on the obstacles that are between emitter and receiver.

This method may present some more disadvantages, like the fact that when the patient moves, moving the dripping chamber at the same time, the ultrasonic detection may be inaccurate, or even not work.

2) Piezoelectric method

Sheets of piezoelectric material have been used for the implementation of this method (see Figure 2). The piezoelectric material has the characteristic of varying the voltage between its tips due to some mechanical deformation to which it is exposed. Thus, external vibrations will be able to be detected by a minimum deformation in this material.

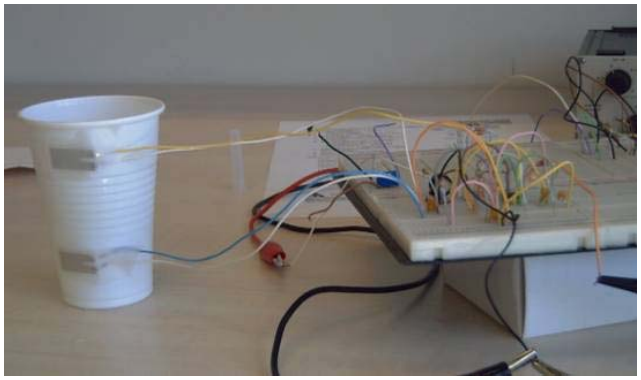


Figure 2 Measurement assembly with piezoelectric material

The technique of using piezoelectric materials can be optimized in order to obtain the dripping detection. However, we can see that the external vibrations affect the measures, thus making this method invalid for the mobile dripping systems.

3) Capacitive electrodes

The capacitive electrodes method consists in the detection of the capacity variation between two copper sheets produced by the dielectric change generated by the crossing of a drop through the generated field. In Figure 3 the assembly can be seen.

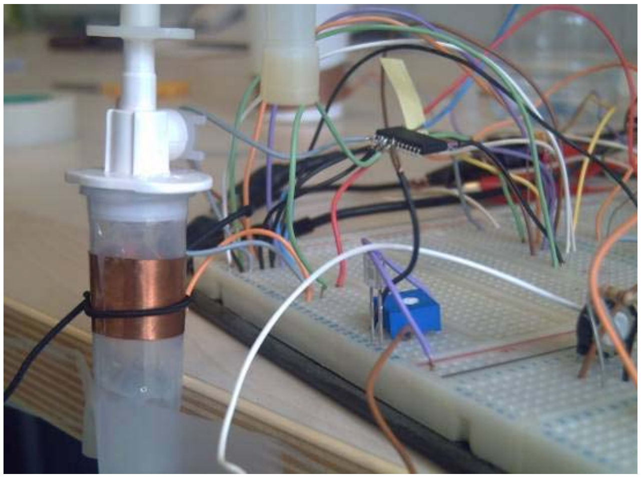


Figure 3 Measurement assembly of capacitive electrodes

The capacity between electrodes follows this equation:

$$C = \frac{k.\varepsilon_0.A}{d} \qquad (1)$$

Where:
C = capacity in Faradays.
A = area of copper sheet in m2.
d = distance between sheets in meters.
k = dielectric constant of the inter-sheet material.
ε0 = permitivity in free space (8.85x10-12 F/m).

In order to ease the calculations when the materials between the plates are different, the capacity also could be defined as:

$$C = \varepsilon_0 \frac{A}{l_e} \ \text{ανδ} \ l_e = \frac{l_1}{\varepsilon_1'} + \frac{l_2}{\varepsilon_2'} + \ldots + \frac{l_n}{\varepsilon_n'} \qquad (2)$$

Where l_e is the equivalent thickness, ε_n' is the relative permitivity of the material and l_n the length. If we consider that the media is isotropic, homogenous and linear $k = \varepsilon'$.

Having in mind the material of the drip chamber (polyurethane) and making the suitable operations, the capacity between the electrodes varies a bit with the passage of the drop, which makes the detection circuits more complex.

4) Optical method

The optical methods try to detect a variation in the tension level in a phototransistor or photoelectric cell, produced by the deflection of the beam of light generated by the passage of the drop. To achieve that, visible LEDs (Light-emitting Diode) of diverse colors, infrared LEDs and lasers have been used as emitters, whereas, phototransistors and photoelectric cells have been used as receivers. In Figure 4 one of the assemblies of the optical method can be seen.

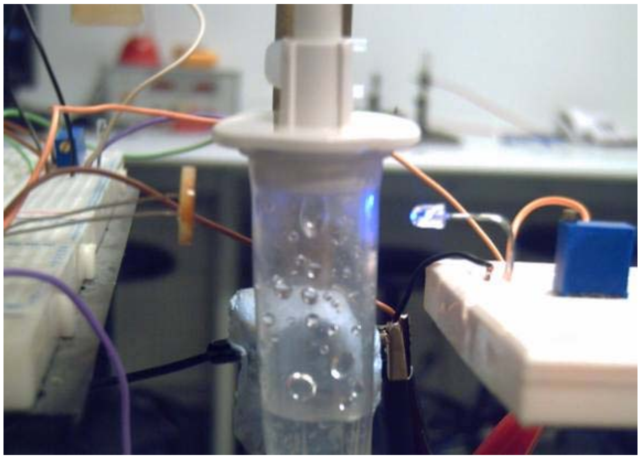


Figure 4 Method of optical measurement

Using these optical methods the dripping detection through the intravenous drip chamber is possible. The use of this method for the implementation of our sensor has been chosen, which has fundamentally lower cost, minor circuital complexity and low energy consumption, because the sensor will be battery-fed.

B. Aims of the system

The overall objective of the system is the dripping detection of all the patients in a sanitary center and to contact with a control center, so that the personnel in charge attending to the patients will be more efficient. To do that, the system is equipped with wireless sensors for dripping detection and central devices in charge of receiving and redistributing the signals received by the wireless sensors.

Besides, the system is optimized to minimize the energy consumption as well as the cost. The energy consumption optimization is done in each module, achieving a considerably long battery life.

There are other similar thematic articles which are not centered in the energy consumption [8].

II. HARDWARE IMPLEMENTATION

The hardware architecture of the wireless device has been divided in four modules: the sensor module, the radio module, the feeding module and the microprocessor. In Figure 5 the scheme of this new platform can be observed. As it is possible to see, in order to save energy, the sensor module, in the left part of the figure, is fed by a switch, which is activated by the microprocessor when it is desired to make a measurement.

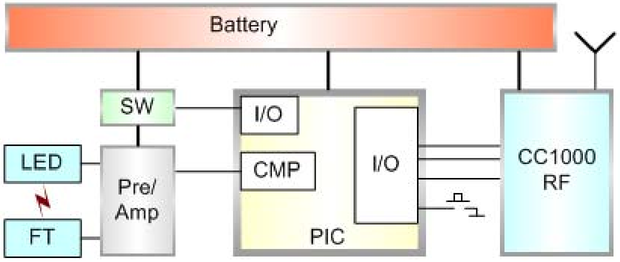


Figure 5 Sensor Scheme

The hardware design, at electronic scheme level as well as at PCB "layout" level, has been made using the OrCAD tool [9].

In the next paragraphs we will do a brief description of the functionalities of each module:

A. Sensor module

This module consists of two stages: on the one hand we have the sensorial stage, composed by the emitting LED and the phototransistor, and on the other hand is the stage of preparation and amplification of the detected signal.

The study of dripping detection techniques has determined that the beam of light used for the sensor is the optical one. The passage of the drop turns aside the beam and the light intensity that arrives at the phototransistor decreases, so a small fall of tension takes place.

That tension difference is the signal we will have to prepare and amplify for its insertion into the microprocessor. For that purpose, a double operational amplifier is used; one of them is used to stabilize the reference tension and the other to amplify the variation of tension produced by the drop. In Figure 6 the block diagram of this module can be seen, where the comparator part is done inside the microprocessor, which have two comparators.

On one hand, the reference level is introduced to one of the pins of the internal comparator of the microprocessor and on the other hand, the reference signal plus the amplified signal is introduced in the other pin.

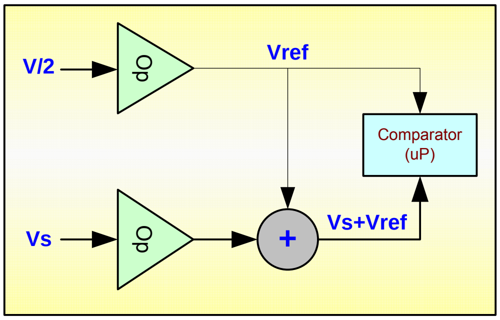


Figure 6 Signal Preparation

B. Radio module

The radio used for the wireless communication is the CC1000 of Chipcon [10]. The handling of the radio is carried out by the microprocessor of the serum sensor, to that purpose, we connect 6 pins of the microprocessor to PALE, PDATA, PCLK, DCLK, GAVE and CHP_OUT/LOCK pins of the CC1000 radio.

In order to achieve the objective of low power consumption of the wireless device, the radio remains in low consumption mode most of the time, only changes to TX mode to transmit strictly necessary messages, as they are the warning of dripping end and low battery.

The sensor has a button to activate the configuration mode of the wireless device. In that configuration mode the wireless sensor communicates with the central device asking for one unique ID. This way, the central device assigns one unique ID to the sensor, and it is known what room it belongs to. Thus, the switch insertion problems are avoided and any sensor can be operative under any central device of the different rooms.

C. Microprocessor

The microprocessor module is the core of the system. It receives the dripping signal and determines if the intravenous infusion system is dripping or not. The microprocessor also controls the battery level of the device.

However, the microprocessor changes the operation mode into configuration mode if it detects an interruption produced by the configuration button.

The microprocessor also controls the radio communication; in such way that the first routine it must do is configuring the radio, which means, to choose the correct frequency, the correct power and so on. Another process is to check the information of the received signals, and to send the corresponding message to the central device using the designed communication protocol.

The energy saving is very important in the wireless sensor device, so the microprocessor also is in charge of increasing the energy saving using a listening algorithm. To do that, it is able to switch-off the sensor module when the dripping monitoring is not important. This software saving obtains a great increase of battery life.

For the design of the wireless device, the microprocessor PIC16F688 of Microchip has been used [11], which is made with the new nanoWatt technology, so it has a very small consumption in "running" mode, around 300uA with 4 MHz clock, and in "Sleep" mode it consumes near 80nA.

In the next figure, Figure 7, an image of the prototype of the wireless sensor device can be seen.

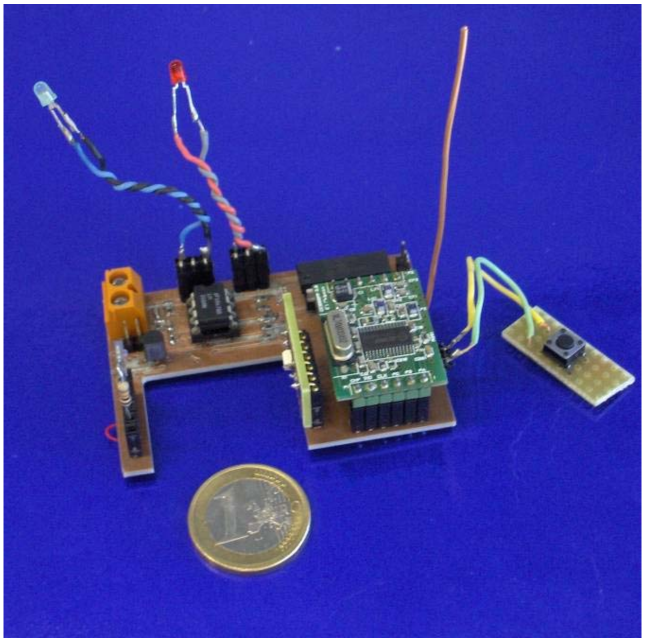


Figure 7 Wireless sensor device prototype

D. *Feeding module*

The feeding module is in charge of providing adapted tensions to each part of the sensor and also controls the battery level. When the battery level is low, it warns to the microprocessor so it sends an alert signal of low battery, then the corresponding person changes the battery of the device.

The battery life in a wireless sensor is the key for its acceptance in the market, so here is a study of the lithium battery load duration of the CR2 of Panasonic; its rated load is 750mAh.

Figure 8 shows the obtained results of the evaluated battery during 100 hours with a continuous unloading current of 10mA. It is observed that both voltages in the graph agree at any moment. During the first 80 hours the voltage provided by the battery is stable and superior to 2.5V. With the applied unloading cycle, around 30.000 unloading cycles each 24 hours are considered, which supposes about 100.000 cycles in 80 hours. These consumption maximums have been of 25mA during 25ms and 10mA during 2.5s. Therefore, the energy consumed in these 80 hours is of 730mAh approximately, which is equivalent to a battery yield of 93%.

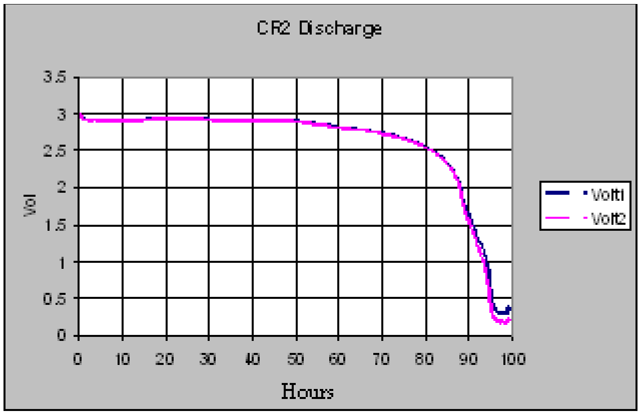


Figure 8 Curve of unloading of the battery

III. Software implementation

A. *Software implementation of the Wireless Sensor Device*

The correct implementation of software is a key point when communicating all the modules described in the hardware section. The chosen language to program the microprocessor is C.

The software developed to be embedded in the platform has a modular scheme. This design allows the software being independent from the platform and also gives flexibility.

The whole software structure is divided in 4 layers as it is depicted in Figure 9. The layers are separated by dotted lines and a short description for each one is given below:

- *Physical level*: is the lowest level and it depends on the hardware directly. The modules present in this level are corresponding to the physical modules of the node; these are analog and digital sensors, Serial Parallel Interface (SPI bus), Timers, Memory Access, etc.
- *Controller level*: this layer is the interface between controller and application. The functions developed in this level permit the application level to invoke controller functions. The ADC (Analog Digital Converter) module converts analogical signals from the analog sensors to digital, and the SPI allows communications of the CPU with the CC1000 RF chip. In this layer are also included the set of RTC (Real Time Clock) functions.
- *Interface level*: this layer contains the main functions that the sensor node performs during its duty cycle. These functions range from reading ADC channels or communicating through the SPI interface, to sending and receiving data from RF chip and access to the comparator module. Programmed interrupt routines functions are also developed in this layer.
- *Application level*: this is the top level layer and executes related actions according to interruptions received (external switches or internal interruptions).

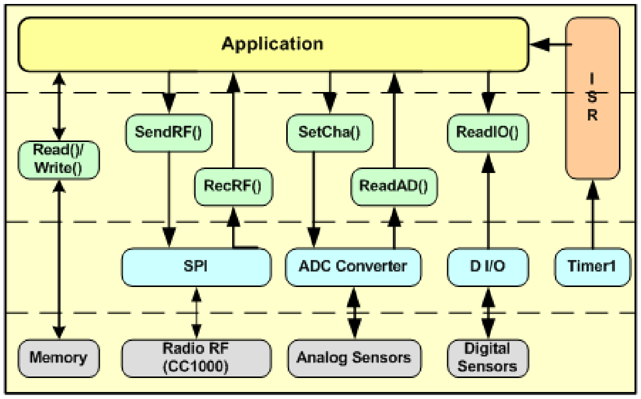


Figure 9 Embedded Software

In Figure 10 the flow chart of the Application layer can be seen, which shows the wireless device operation, summarized in the following lines: once the device is ON, the program sets up the radio to operate at 868.3 MHz and also sets up other conditions like power output, etc. Later the wireless device asks for the unique ID to the central device and keeps it in memory. This unique ID will identify it throughout the session. It would be possible to change the unique ID in the same session by pressing the configuration mode button.

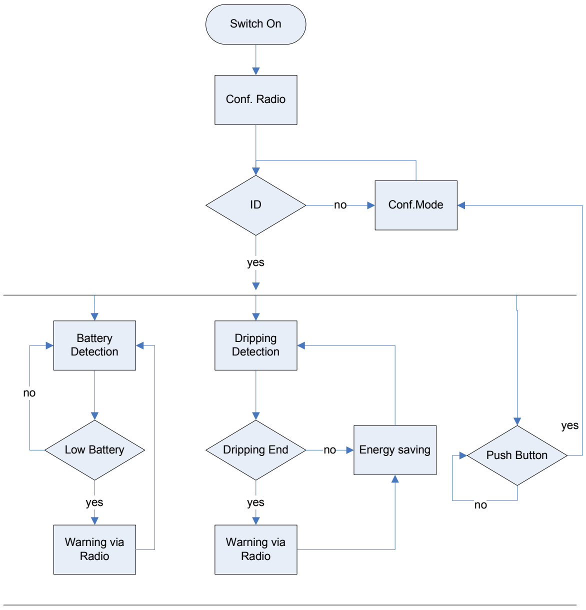


Figure 10 Software diagram

When the whole set up is done the microprocessor is pending of three tasks. One of the tasks is the dripping detection and warning of the end of dripping in case it is necessary.

Apart from that, the microprocessor is pending to see the battery state to warn about the need of change if it is low.

Finally, the microprocessor is pending of the external interruption generated by the configuration mode button pulsation.

B. *Energy saving algorithms*

The end of the dripping detection in the intravenous systems is not essential to be immediate, so a margin period from the end of the dripping to the renewal exists. This margin makes it possible to accomplish a listening period to verify if the intravenous system is dripping or if it has stopped.

The listening periods make that the time when the sensor module is ON to be minimized, which implies an energy saving.

Figure 11 shows that the energy consumption can be reduced to 50% (or less), making listening and extinguished periods equitable o minors.

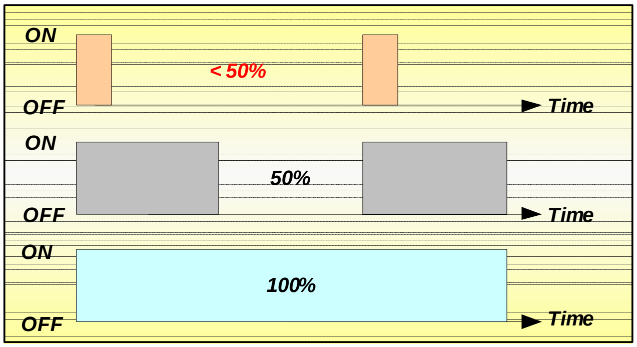


Figure 11 Energy Saving algorithm

Making studies had determined that more energy can be saved if during the listening periods we count the number of drops. Thus, if during listening we count a fixed number of drops, for example 3, we already can know that the dripping exits, so we do not have necessity to continue listening the dripping the rest of the listening period, then the saving is increased in more than a 50%, depending on the dripping frequency (first graphic in Figure 11).

C. *Protocol Implementation*

Figure 12 shows the composition of the data bytes that are sent from WS (Wireless Sensor) to the Room Base Station (RBS). The number of bytes to send is 18, which can be described in the next paragraphs:

- Preamble: is composed by 32 bits and it allows to the RF chips to start the communication. I our case the double word chosen for preamble was 0xAAAAAAAA.
- ID: there are 2 bytes (16 bits) reserved for the ID, which can be unique for the devices.
- The Payload is composed of 64 bits (8 bytes) as shown in the figure. The content of the payload is the following bits:

  - The Time: each sensor has a Real Time module and it serve to know the exact time at which sensor has an alarm.
  - Status bits are the most relevant, and are composed for the status of the dripping system (empty or not), the status of the battery, etc.
  - Reserved: there are 16 bits reserved for future applications.
- CRC: is used to control that the data that is send to the RBS station reach this target.

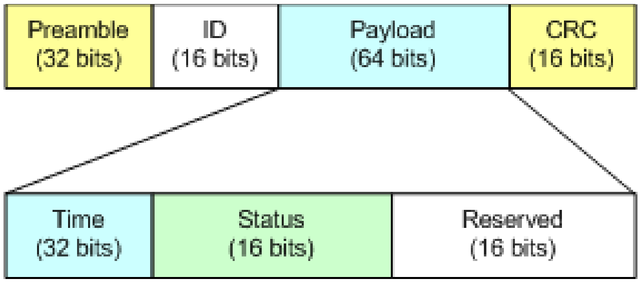
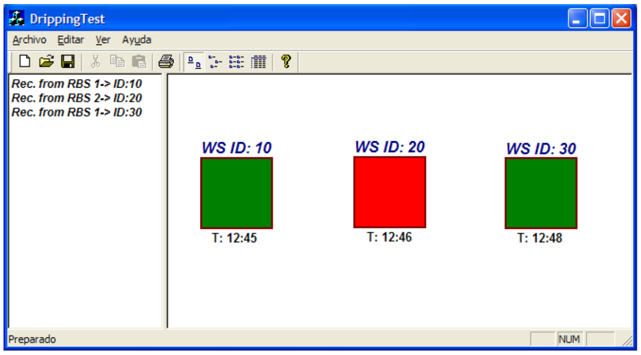


Figure 12 Protocol Implementation

D. *PC application*



Rec. from RBS 1-> ID:10
Rec. from RBS 2-> ID:20
Rec. from RBS 1-> ID:30

WS ID: 10 — T: 12:45
WS ID: 20 — T: 12:46
WS ID: 30 — T: 12:48

Figure 13 PC Program Application

The PC application is a windows based program, which was designed only to test the devices. This environment has been developed in Visual C++ platform and has the following functionalities:

- It allows knowing how many wireless sensors (WS) are in the system.
- To see when each WS is empty or not, in which case it sends an alarm to the Room Base Station (RBS) located nearer. In that case, the PC program can represent it using colors (in our case, red color), which means that the dripping chamber is empty and will be replaced immediately.
- To configure the sensors: set the date and some configuration parameters in the node, like the frequency channel for wireless communication, power of transmission, ID, etc.
- To store the information in a local database for further analysis

IV. Network topology

Prior to describing the network topology of the dripping detection wireless sensor network, a separation between two kinds of dripping systems has to be done.

In every hospital we can find patients which cannot move and are lain down on the bed all the time. Apart from those, there are also patients which can move around and walk during the day, in any moment, and with a high degree of freedom.

Each of these patients may need an intravenous dripping system. Thus, two systems of this kind can be found in hospitals currently:

- Static intravenous dripping system: these are located in each room, next to the beds, and are used by those patients which cannot move and spend all the hospitalization time in bed.
- Mobile Intravenous dripping system: these are used by those patients who have the ability to move around, and consist of a human-height-size metallic bar, equipped with small rollers or wheels. On top of it is where the serum bag and the dripping detection system are located.

In both cases the need of automatic detection of a run put of intravenous solution is present. As a consequence, a different network topology has to be defined for each case, determined by the mobility of the patient.

A. *Static network topology*

As mentioned before, the static intravenous dripping systems are those which are located in each room of the hospital, next to the beds in which the patients lie down. Figure 14 shows a picture of one of these systems. And there is one for each patient, so it can be established a one-to-one equivalence between patients and dripping systems. This eases the identification of the dripping systems which need to be replaced.

Figure 14 Static intravenous dripping system

The implemented static network topology can be observed in Figure 15. The wireless sensor devices (WS) communicate via RF with the central devices of each room (RBS). RBS devices are in a LAN (Local Area Network), communicated with the central server in which the control of the dripping state of all the intravenous systems will be carried out.

The configuration mode of WS sensors makes it possible to a sensor from a room to be transferred to another one and the system will continue working in the correct way, because the changed WS will contact with the new RBS in the present room.

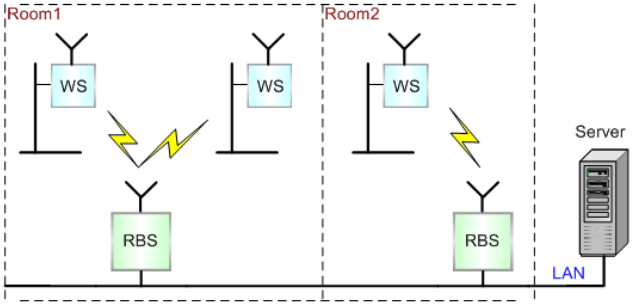


Figure 15 Diagram of the System

B. *Mobile network topology*

As stated before, the mobile intravenous dripping systems are those which can be carried along the hospital corridor by patients capable of moving around. Figure 16 shows a picture of one of these systems.

Figure 16 Mobile intravenous dripping system

For the case of the mobile system, simulations have been carried out. The simulated network topology for this case can be seen in Figure 17.

The diagram represents a patient carrying a mobile intravenous dripping system along a hospital corridor. The dripping system is equipped with a wireless sensor, and along the corridor, several Base Stations (BS) are located.

The mayor challenge in this case is to achieve a continuous and seamless connectivity, in order to be ale to carry out two functions:

- Send an alert of intravenous liquid replacement in any moment, and anywhere the patient is in that moment.
- Localize the patient in the hospital, using the Base Station Identifier of the BS to which it is connected in that exact moment.

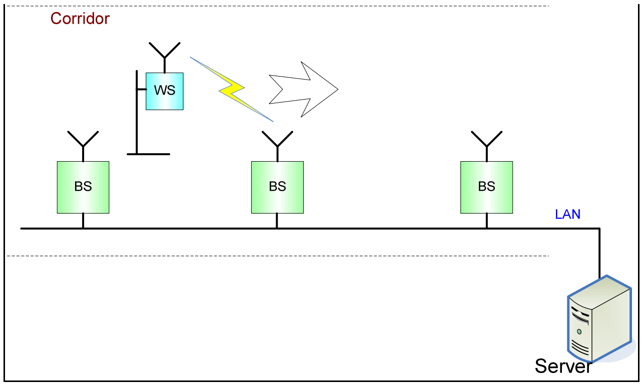


Figure 17 Diagram of the Mobile system

V. MOBILITY

This section describes the mobility algorithm designed for the mobile intravenous dripping system, as well as the simulations carried out in order to validate the system.

Traditionally, in the field of wireless sensor networks, only three types of mobility were defined:

- Node mobility: is the case in which the wireless nodes themselves are mobile. The network must be capable of reorganize itself frequently enough to be able to function correctly.
- Sink mobility: the most interesting case involving sink mobility is the one in which the sink isn't part of the sensor network.
- Event mobility: in applications like event detection and in particular in tracking applications, the cause of the events or the objects to be tracked can be mobile.

The application presented in this research work corresponds to the first type of mobility listed above.

A. *State of the Art in node mobilty*

Nowadays, the wireless sensor node mobility principles are studied mainly under the umbrella of MANET (Mobile Ad-hoc Network) related research work.

A mobile ad hoc network (MANET), sometimes called a mobile mesh network, is a self-configuring network of mobile devices connected by wireless links. Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic.

The limited resources in MANETs have made designing an efficient and reliable routing strategy a very challenging problem. An intelligent routing strategy is required to efficiently use the limited resources while at the same time be adaptable to the changing network conditions such as network size, traffic density, and network partitioning.

One of the most popular methods to distinguish mobile ad hoc network routing protocols is based on how routing information is acquired and maintained by mobile nodes. Using this method, mobile ad hoc network routing protocols can be divided into proactive routing, reactive routing, and hybrid routing.

In a reactive routing protocol, routing paths are searched only when needed. A route discovery operation invokes a route-determination procedure. The discovery procedure terminates when either a route has been found or no route is available after examination for all route permutations. In a mobile ad hoc network, active routes may be disconnected due to node mobility. Therefore, route maintenance is an important operation of reactive routing protocols.

For the research work presented in this paper, a similar strategy to that used by reactive routing protocols has been chosen. As the wireless sensor moves along the coverage of different base stations, the active route or link is determined and selected reactively, based on the signal power received from the base stations.

B. Design of the algorithm

Figure 18 shows the flow diagram which determines how the mobility algorithm operates.

The determination of the active link is based on the received signal strength. Among all the received signals coming from the base stations, the wireless node selects that which has the higher received power.

The algorithm, after the initialization stage, measures the received power of all the signals coming from the reachable base stations, and with that it builds a data base, including the identification of the base station and the received signal strength. With that information, the algorithm is able to identify which of the base station has the highest received signal strength, which will be used by the next stage of the algorithm.

The next stage establishes the link using the information provided by the previous steps of the algorithm. The whole process is repeated every second, in order to update the status of the connection.

This algorithm allows the achievement of the two main objectives determined before:

- It allows the mobile wireless sensor to be permanently connected to the hospital LAN, and thus, send an intravenous–liquid-replacement alert anytime and anywhere the patient is.
- It allows the localization of any wireless sensor in the hospital, due to the characteristic of each of them being permanently attached to a base station. Looking at the identifiers of the wireless node and base station that form a certain link, the task can be performed.

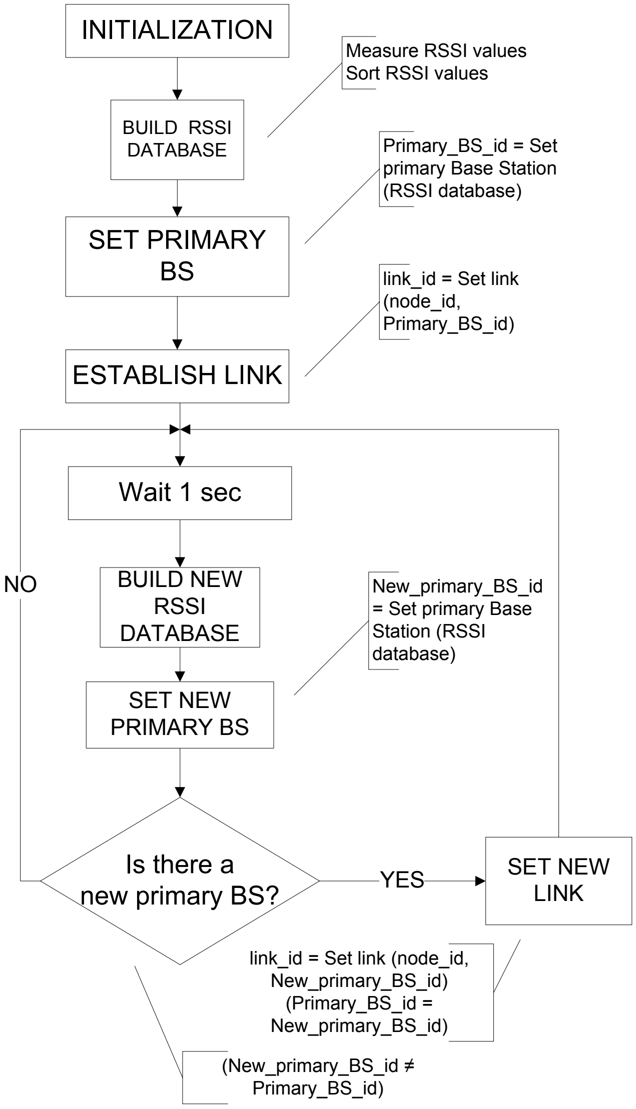

INITIALIZATION
Measure RSSI values
Sort RSSI values
BUILD RSSI DATABASE
Primary_BS_id = Set primary Base Station (RSSI database)
SET PRIMARY BS
link_id = Set link (node_id, Primary_BS_id)
ESTABLISH LINK
Wait 1 sec
NO
BUILD NEW RSSI DATABASE
New_primary_BS_id = Set primary Base Station (RSSI database)
SET NEW PRIMARY BS
Is there a new primary BS?
YES
SET NEW LINK
link_id = Set link (node_id, New_primary_BS_id) (Primary_BS_id = New_primary_BS_id)
(New_primary_BS_id ≠ Primary_BS_id)

Figure 18 Flow diagram of the mobility algorithm

VI. CONCLUSIONS

In this section we will summarize the most important conclusions which can be extracted from this project. It is necessary to emphasize the following aspects obtained with the present design:

- Reduction in consumption in comparison to other wireless devices with the same characteristics.
- Low cost of the device.
- The device size, the sensor dimensions are considerably smaller.

- Network forming flexibility, being able to use any sensor in any room.
- Great contribution in the improvement of effectiveness and security in the health care of the patients.
- Easy scalability of the system.

Regarding the mobility algorithm, there are two main tasks pending, which are planned as future work. On the one hand, obtaining concrete parameter measurements from the simulations, e.g. bitrate, number of lost packets, or switching time.

On the other hand, the implementation of the mobility algorithm is planned. It will be carried out using the sensor nodes described in this paper, and real life measurements are going to be done using a backbone network deployed along a concrete area of a building.

We also can mention the great ease of the system evolution. The future work can be focused on these topics:

- Location systems.
- Use of the wireless sensor itself as a node, therefore they could resend signals from other nodes.
- Greater advantage of the processing capacity of the wireless sensors.

References

[1] Drip Managing Device. JP08098884.

[2] Ultrasonic Air and Blood Foam Detector. US4068521.

[3] Indirect Piezoelectric Drop Counter And Method. US4583975.

[4] Drip Monitoring Device. JP07289635.

[5] Method And Apparatus For Electrical Signal Coding. GB8900132.

[6] Intravenous Drip Rate Control Device. US4493710.

[7] Device For Monitoring And Controlling An Intravenous Infusion System. EP0588427.

[8] Wireless Sensor Network for Patient IV Monitoring .Wenfeng Li and OnChing Yue. Mobile Technologies Centre (MobiTeC). The Chinese University of Hong Kong. Shatin, Hong Kong, China.

[9] ORCAD www.orcad.com (last access: July 2010)

[10] Chipcon AS, Gaustadalléen 21, NO-0349 Oslo, NORWAY www.chipcon.com (last access: July 2010)

[11] Corporate Headquarters Microchip Technology Inc. 2355 West Chandler Blvd. Chandler, Arizona, USA www.microchip.com (last access: July 2010)

A Distributed Network Management Approach to WSN in Personal Healthcare Applications

Karla Felix Navarro, Elaine Lawrence,
Doan Hoang
School of Computing and Communications
University of Technology Sydney
Sydney, Australia
Karla| elaine| dhoang@it.uts.edu.au

Yen Yang Lim
University of Technology Sydney
Sydney, Australia
Brian.lim@uts.edu.au

Abstract—**This paper describes the development of a Wireless Sensor Network personal health monitoring system called Medical MoteCare which uses a combination of medical and environmental sensors. SNMP and CodeBlue agents are incorporated in the system as is the network management software JaguarSX. Network management models and tools provide an alternative, scalable and affordable solution to WSN health monitoring applications that allow for data storage correlation and dissemination as well as timely alerts when parameters are breached. This work forms part of a large grant aimed at providing assistive healthcare for the elderly.**

Keywords: Wireless Sensor Networks; Network Management; Data Correlation.

I. Introduction

This paper describes the development of a Wireless Sensor Network (WNS) Personal Healthcare application capable of monitoring a patient's vital signs such as temperature, heart rate and blood oxygen level. The WSN based medical application uses network management tools and models to store and correlate the collected data and issue warnings if thresholds are breached. The Medical MoteCare prototype aims to assist all stakeholders in the personal healthcare arena to improve the monitoring and, therefore, the health outcomes for the aged, the chronically ill and the infirm. It is the latest in a series of 4 prototypes that have been developed so far [1], [2], [3], [4].

An ageing population trend is particularly noticeable in developing countries and, overall it is forecast that, between the years 1990 and 2020, the proportion of people aged between 65– 74 will increase by 74% [5]. The impact of these population changes means that governments and relatives/caregivers will need to spend increasing amounts of money and time to care for this cohort. Wireless sensor networks offer some hope for decreasing the cost of monitoring the elderly and sick in their own, or in nursing, homes. Wireless Sensor network nodes (Motes) equipped with sensors can be used to monitor such items as temperatures in the environment or on the person, cameras may be linked to motes to enable remote monitoring and thresholds can be set to ensure that action is taken if a particular limit is reached.

The justification for this research is spread across economic and social imperatives, such as helping governments to limit the rising costs for caring for the sick and the elderly whilst not jeopardizing the standard of care. One overarching area of interest of the healthcare industry is in heterogeneous and autonomous data acquisition systems. It is important to develop cost effective treatments which optimize patient safety, while minimizing treatment cost and the possibility of malpractice litigation.

This paper describes Medical MoteCare as a proof of concept of a network management based healthcare monitoring system and reports on the outcomes. The next section provides background information while section 3 outlines health sensing and monitoring. Section 4 describes the architecture of the Medical MoteCare based on an adaptation of the Three-Tier and SNMP Proxy Network Management Organizational Models, using the network management tool JaguarSX. Section 5 describes the system and a scenario setting is outlined in Section 6 and results are displayed and discussed in Section 7. The conclusions are drawn in Section 8.

II. Background

Given the increase in the numbers of the elderly and the current and projected nursing shortages, health care organizations must find ways to meet demands with fewer resources [6]. The ability to integrate sensors for fall detection, video surveillance, sleep disorder monitoring, heart attack identification and problems with obesity, will improve the usefulness of the Medical MoteCare prototype. People will be able to move about in their own home secure in the knowledge that they are being monitored. Just as airplanes are monitored by air traffic controllers, our patients may be monitored remotely by healthcare providers or carers who will have access to the patient's information via a web server and, in the case of an emergency or at a predetermined time, via audio and video links [7].

The following statistics illustrate the potential savings that can be gained by using home monitoring systems that allow patients to leave hospital early. Table I provides the following figures from 2004 for the United Kingdom:

TABLE I. ECONOMICS OF UK PATIENT CARE [8]

Explanation	Amount
Average cost of care per week/per person in a hospital	GB Pounds 805
Average cost of care per week/per person in a nursing home	GB Pounds 337
Average cost of care per week/per person in own home	GB Pounds 120

Other statistics from [8] further illustrate the savings that can be made if mobile health devices are used by early discharged patients in Germany (Table II). The figures exclude administrative costs that are also applicable if using a mobile monitoring system.

TABLE II. ECONOMICS OF GERMAN PATIENT CARE AND MOBILE HEALTH DEVICES [8]

Explanation	Amount
Early discharged hospital patients (mobile services (20% of total))	3.3 million Euros
Average costs for one hospital day	150 Euros
Average number of hospital days saved through early discharge	3 days
Total yearly cost savings through early discharge	1.5 billion Euros

Thus health economics reveal that potential savings can be made by using mobile and wireless home monitoring devices. Hopefully patients will not be admitted to hospital unnecessarily; they may be discharged earlier; the burden on emergency services could be reduced; medical personnel could remotely assess patients, the patient and relations could reduce their travel times and home visits by experts could be reduced.

III. HEALTHCARE SENSING AND MONITORING

A sensor is a device that detects the presence and/or the variation of some physical phenomena and converts the sensed quantity into a useful signal that can be directly measured and processed, such as voltage or current. A smart sensor is a sensor that provides extra functions beyond those necessary for generating a correct representation of the sensed quantity. Often they possess processing, storage, and decision making capabilities. Table III below outlines sensing properties and examples.

TABLE III. SENSING PROPERTIES AND EXAMPLES [9]

Sensing	Examples
Physical properties	pressure, temperature, humidity, and flow
Motion properties	position, velocity, angular velocity, and acceleration
Identification	by personal features
Biochemical	by biochemical agents
Contact properties	strain, force, torque, slip and vibration

For health monitoring, wearable medical sensors are of particular interest as these devices are used to monitor a set of key ambulatory parameters in domains such as oncology, paediatrics, and geriatrics. Some of these parameters include: heart rate for cardiac function; acceleration during walking and running for activity; body temperature for illness; virtual capacity for severity of airway obstruction in chronic obstructive pulmonary disease; blood glucose for vascular or neurological complications; EEG for seizure disorders, confusion, head injuries, brain tumours, infections, degenerative diseases, and metabolic disturbances that affect the brain; ECG waveform for cardiac arrhythmias; blood pressure; arterial oxygen saturation for sleep disorder; and body weight [10].

The Medical MoteCare system sensing can be achieved with on-body and off-body sensors. With on-body sensors, the elderly can be mobile and the on-body sensors can form a body area network (BAN) which reacts when the conditions recorded imply that the elderly needs immediate assistance (Fig.1). In Medical MoteCare, the network management software suite Jaguar SX is utilized for correlating the data and alerting the caregivers if problems arise.

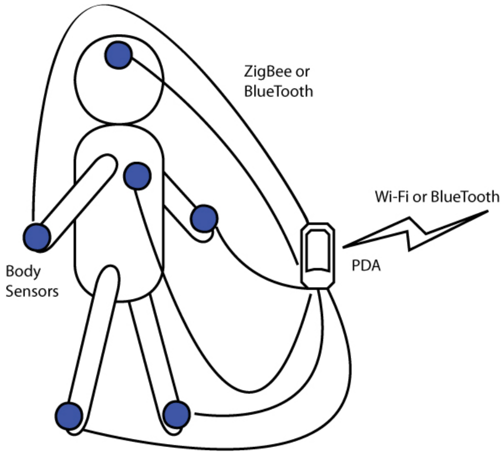


Figure 1. Body Sensors and Communications.

IV. MEDICAL MOTECARE DESCRIPTION USING JAGUARSX

The strength of the Medical MoteCare system is its innovative deployment of a network management model in a scalable, flexible way to manage multiple sensors and networks of sensors reliably and securely. The building of the Medical MoteCare system brings the integration and testing of specialized sensors for healthcare monitoring (pulse oximeters, light and temperature sensors with Zigbee motes [10]) through the use of an enhanced version of the Harvard CodeBlue software suite [11], a Simple Network Management Protocol (SNMP) Agent, and the Network Management (NM) tool JAguarSX for scalability, performance improvement and data correlation.

The Medical MoteCare system was based on the adaptation of the Three-Tier and SNMP Proxy Network Management Organizational Models [12]. The SNMP Proxy enhancements enabled this system to improve the scalability, modularity and flexibility of the system by potentially bringing to the developer communities the freedom of selecting from a vast range of existing SNMP based NM tools to fit their specific WSN application requirements. The adapted Three-Tier NM Organizational model can be seen in Fig.2.

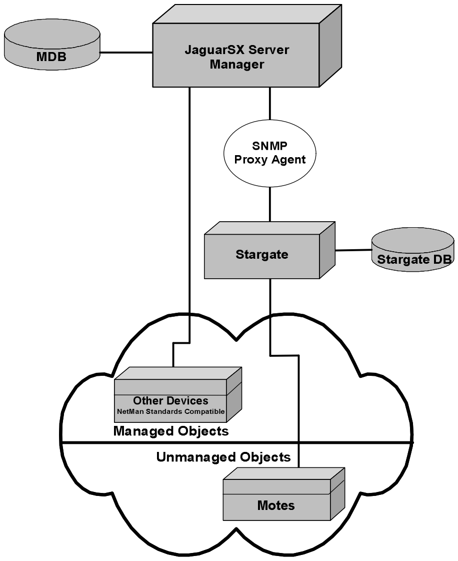


Figure 2. Adapted Three-Tier NM Organization Model to the Medical MoteCare System [12].

The enhancements in the adapted Three-Tier NM Organization model with a tailored SNMP proxy agent shown in Fig.2 allow for the natural implementation of any SNMP-capable NM tool in the Medical MoteCare system. This makes the system not only capable of having more than one manager entity without the risk of over polling the managed devices, but also makes it platform independent by allowing SNMP standard communication with any number of NM systems acting as Manager, Middle Manager or Agent Entities.

The integration of an SNMP agent in conjunction with the CodeBlue Communication module enabled the system to be accessed via any network management tool that understands SNMP. This ensured that the system was able to scale up easily and provide a robust and well tested, standardized environment.

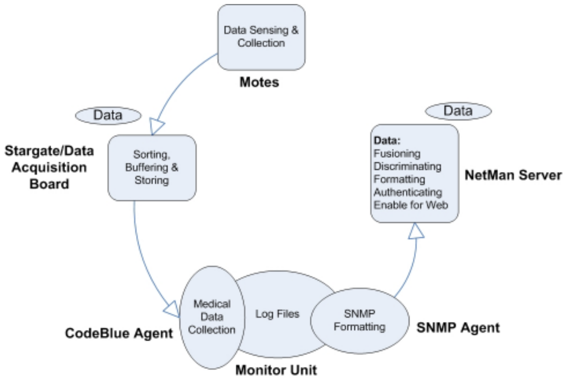


Figure 3. System Process Overview for the Medical MoteCare System [13].

The system process overview of the Medical MoteCare System in Fig.3 depicts the high level system processes for each of the system components (labeled in bold). A more detailed explanation of these components and their processes is further explained in this section.

The WSN architecture of this prototype was based on MicaZ motes [2] which added robustness to the system communication (as a result of the use of the Zigbee protocol). The sensors include the Pulse Oximeter, a medical sensor from BCI, Inc. consisting of a standard finger or ear sensor, a pulse oximetry module (BCI Micro Power Oximeter board) and a mote [6].

Figure 4. MicaZ Mote [10].

Blood oxygen saturation (SpO2), pulse and plethysmogram waveform data are relayed via a serial interface to the MicaZ node. The pulse oximeter mote device periodically transmits packets containing the measured samples.

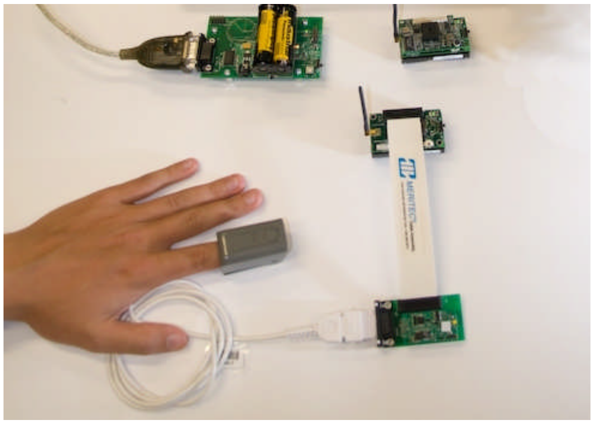


Figure 5. The Pulse Oximeter [14].

The MTS310CA flexible sensor board has a number of sensors such as light and temperature which are incorporated into the Medical MoteCare system, to illustrate the potential importance of environmental impacts on the physical condition of a monitored patient. For example, an elderly depressed person might need adequate lighting to alleviate mental health problems. The light sensor could prove that the light is appropriate. Alternatively, if a person, who has suffered a heart attack recently, suddenly starts sweating, the symptoms could signal another attack or it may be that the room is too hot.

The Data Acquisition Board is the Stargate Personal Server processing board sourced from Crossbow. The Personal Server, Stargate has the ability to connect wirelessly to other devices using Wi-Fi, GSM and GPRS. This hardware allows a system to move from a centralized network management system to a distributed one.

The Monitor Unit comprises the enhanced CodeBlue software that incorporates log files of data recordings captured by the sensors on the MicaZ motes and an SNMP agent to enable it to deal with standardized network communication using the SNMP protocol. The Monitor Unit connects to the Gateway Mote either via a USB connection, a serial connection or Internet Protocol (IP) [2].

On the Monitor Unit side CodeBlue captures the data and stores it locally on log files. It is then sent through an SNMP agent after a request is made by the Network Management Server. The CodeBlue system with the incorporated SNMP agent provides access to the widely accepted TCP/IP protocol suite. As motes have extremely limited processing power, battery life and transmission budgets, SNMP was not implemented on the motes themselves as the additional overhead would have dramatically reduced battery life [2]. In the Monitor Unit of the Medical MoteCare system, using the CodeBlue component, an SNMP agent was set up to act as an IP-based Data Delivery Unit (DDU). The DDU is able to act as an agent and deliver information collected on the WSN into a TCP/IP based network [2]. In keeping with the philosophy of the team to utilize as much existing and freely available code and protocols as possible, several Object Identifiers (OIDs) from RFC 1213 were used for the Medical MoteCare system (see Table IV). An object identifier is a label used to name an entity. Structurally, an OID consists of a node in a hierarchically-assigned namespace, formally defined using the ITU's Telecommunication Standardization Sector (ITU-T) Abstract Syntax Notation 1(ASN.1) standard. Successive numbers of the nodes, starting at the root of the tree, identify each node in the tree. Designers set up new nodes by registering them under the node's registration authority [15]. Standard and customized Management Information Bases (MIBs) for the Medical MoteCare system are set out in Table IV. The customized MIB was developed by the mHealth team at UTS creating specific OIDs for the polling of health and environmental data from the WSN nodes [3].

TABLE IV. OBJECT IDENTIFIERS USED FOR THE MEDICAL MOTECARE

OID	Variable Name	Utilized for
1.3.6.1.2.1.2.2.1.1	ifIndex	mote ID as number
1.3.6.1.2.1.2.2.1.2	ifDesc	mote ID as string (mote-xx)
1.3.6.1.2.1.2.2.1.6	ifPhysAddress	mote ID as hex value
1.3.6.1.2.1.2.2.1.10	ifInOctets	pulse sensor value
1.3.6.1.2.1.2.2.1.11	ifInUcastPkts	light sensor value
1.3.6.1.2.1.2.2.1.16	ifOutOctets	oxygen sensor values
1.3.6.1.2.1.2.2.1.17	ifOutUcastPkts	temperature sensor values

The Medical MoteCare system incorporated an SNMP agent. The development of more sophisticated SNMP versions such as SNMPv3 allows for more security which is important when dealing with medical data. The benefits of SNMPv3 for the Medical MoteCare system are that data can be collected securely from SNMP devices without the fear of the data being tampered with or corrupted. This is vital in any healthcare system. As well, confidential information, for example, the SNMP Set command packets that change a router's configuration, can be encrypted to prevent its contents from being exposed on the network. The incorporation of the SNMP agent to the MoteCare Monitor Unit was facilitated by the use of SNMP4J, an enterprise class, free open source and up-to-date SNMP implementation for Java [3].The system therefore utilized an agent derived from the SNMP4J standard example which provided a static SNMP table to hold the sensor data and allow for SNMP requests [2].

The log file is used to communicate between the CodeBlue component and SNMP agent in the Monitor Unit. The structure of the files consists of time and data stamps, Mote/patient ID and sensors available. The standard format is as below:

<date stamp> <time stamp>:<Mote/patientID number type of sensor = data from the sensor>

An example is set out below:
<14 Jan 2008 10:25:16.939 EST : <patientID= 102
PULSE= 92>

Upon detection, the motes will continuously send data from the sensors with IDs on a first come, first served basis.

After the motes are detected, data with an ID code are sent and logged into the Monitor Unit [2].

This Medical MoteCare system utilized commercial software in the form iReasoning MIB Browser for MIB browsing and MIB customisation and SysUpTime Network Monitor and Jaguar SX for alert configuration and event correlation. iReasoning MIB browser allowed the team to load standard and proprietary MIBs for the Medical MoteCare system and issue SNMP requests to retrieve agents' data, or make changes to the SNMP agent. JaguarSX allowed for correlation of data so that alarms could be issued on more than one parameter.

See Fig. 6 for a screen shot of the iReasoning MIB Browser, which will be explained in the section describing the implementation of the Medical MoteCare system.

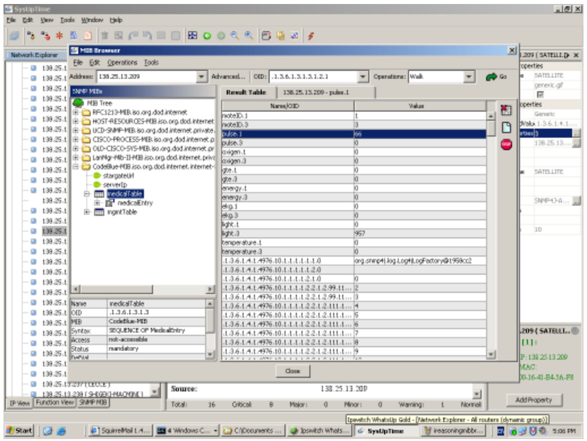


Figure 6. iReasoning MIB Browser.

With the iReasoning MIB Browser and SysUpTime Network Monitor, all network information is stored in a relational database such as FirebirdSQL, PostgreSQL or SQL Server to enable easy and efficient device management and reporting. In a similar manner, the data monitored from the wireless sensor based healthcare application will be stored in such a database to facilitate its manipulation and provide secure storage, discrimination and fusion of data as specified as a requirement of such a healthcare monitoring system.

The SysUpTime Network Monitor can alert network administrators or, in the Medical MoteCare implementation, the health carer, by email, sound, the running of a script, or triggering, as well as taking corrective actions such as executing remote Windows or Linux commands. The data is stored in a relational database that can be accessed via the Internet. Refer to Fig. 7 for a graphical overview of SysUpTime Network Monitor.

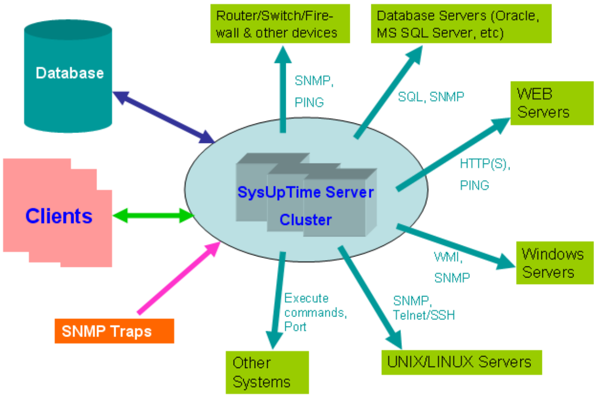


Figure 7. Overview of SysUpTime Network Monitor. Source: http://www.ireasoning.com/sysuptimeoverview.shtml

Using SysUpTime Network Monitor as a NM tool further illustrates the flexibility of using a variety of NM software packages in the Medical MoteCare system. Fig. 8 illustrates the user interface of SysUpTime Network Monitor. In the Performance Graph section of the screen shot, the heart rate of the tester is shown increasing and decreasing in value – the horizontal line indicates the pre-set threshold that will ensure that a warning is given once the threshold is reached. The remote SNMP agent properties are displayed on the right hand side of the screen and show the OID 1.3.6.4.1.4976 that corresponds to the pulse oximeter sensor on the mote as well as the name and IP address of the host (Satellite and 138.25.13.209 respectively) where the agent is residing. It also shows the SNMP version number.

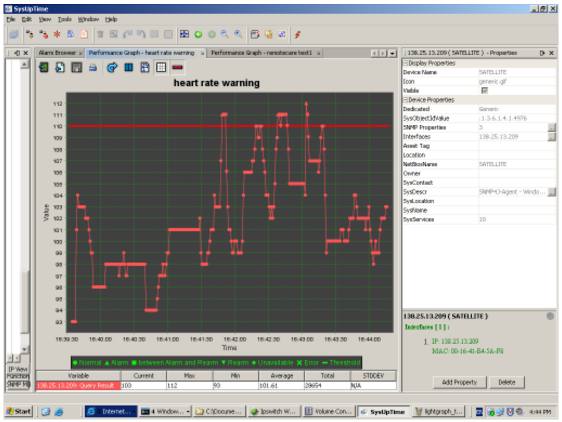
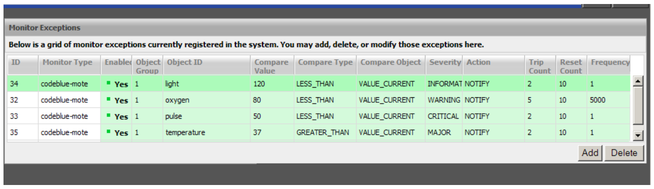


Figure 8. SysUpTime Network Monitor.

As can be seen in Fig. 8 the Medical MoteCare system using the NM tools iReasoning MIB Browser and SysUpTime Network Monitor was able to display medical data in graphical form in real time as well as provide alarm triggering. However the author wanted to illustrate some of the data fusion aspects of NM tools by implementing correlation of medical data and alarm triggering and for that the author used the NM tool Jaguar SX. This will be further discussed in the following section.

Event correlation techniques embedded in various network management tools are expected to allow for the development of personalized applications specific to certain health conditions. Current techniques such as rule-based reasoning (RBR), code-based reasoning (CBR) and case-based reasoning [12] are utilized to add intelligence to system and network management tools by correlating variables and prioritizing them based on thresholds for labeling, various algorithms and hierarchical structures associated to them. For instance if a sufficient number of 'events' in the system are triggered at the same time, e.g. temperature of the person and temperature in the environment, the system will check correlation and prioritize in order to trigger an alarm, message or action [16].

The NM tool Jaguar SX was implemented in the Medical MoteCare prototype with the purpose of adding intelligence to the system by making use of network management correlation techniques to interpret multi-variable collected data and automatically react or prevent harmful health conditions. The system was tested for alarm triggering functionality. Fig. 9 shows a partial view of the Jaguar SX Monitor Exceptions window in which the multi-variable alarm triggering conditions are configured. The compared values and conditions from variables identified by Object ID on column five can be correlated by grouping them into the same Object Group in column four. In order for Jaguar SX to trigger an alarm or notification, all the compared value conditions with the same Object Group have to be "True" (refer to Compare Value and Compare Type on columns six and seven).



Monitor Exceptions

Below is a grid of monitor exceptions currently registered in the system. You may add, delete, or modify those exceptions here.

ID	Monitor Type	Enablec	Object Group	Object ID	Compare Value	Compare Type	Compare Object	Severity	Action	Trip Count	Reset Count	Frequency
34	codeblue-mote	■ Yes	1	light	120	LESS_THAN	VALUE_CURRENT	INFORMAT	NOTIFY	2	10	1
32	codeblue-mote	■ Yes	1	oxygen	80	LESS_THAN	VALUE_CURRENT	WARNING	NOTIFY	5	10	5000
33	codeblue-mote	■ Yes	1	pulse	50	LESS_THAN	VALUE_CURRENT	CRITICAL	NOTIFY	2	10	1
35	codeblue-mote	■ Yes	1	temperature	37	GREATER_THAN	VALUE_CURRENT	MAJOR	NOTIFY	2	10	1

Figure 9. Partial View of Jaguar SX Monitor Exceptions Window.

V. Medical MoteCare System Description

In Fig. 10 below the implementation of the Medical MoteCare System is illustrated. For the implementation of Medical MoteCare the researcher set up the system as follows. A tester was fitted with pulse oximeter device, which was connected to the sensor board on the mote via a serial cable. As the data are being collected it is sent wirelessly to the Stargate Personal Server. The Stargate then sends the data through an Ethernet port to the monitor unit where it is stored in the form of log files created by the CodeBlue component. This step enables the agent to access the collected data and transform it into a SNMP format which makes it accessible to any network management tool that is SNMP-compatible, such as the iReasoning MIB Browser, SysUpTime Network Monitor, Jaguar SX or any other commercialized or open source network management tool.

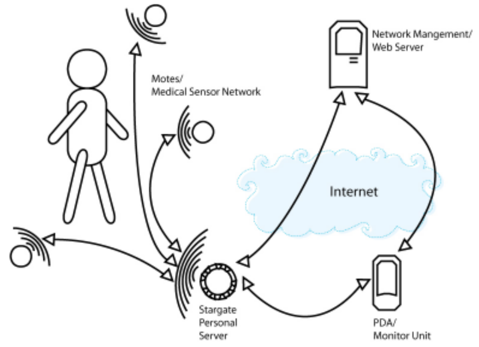


Figure 10. System Overview for the Medical MoteCare System [13].

This step enables the agent to access the collected data and transform it into a SNMP format which makes it accessible to any network management tool that is SNMP compatible, in this instance with iReasoning MIB Browser, SysUpTime Network Monitor and JaguarSX. See Fig. 11 for advantages of this system.

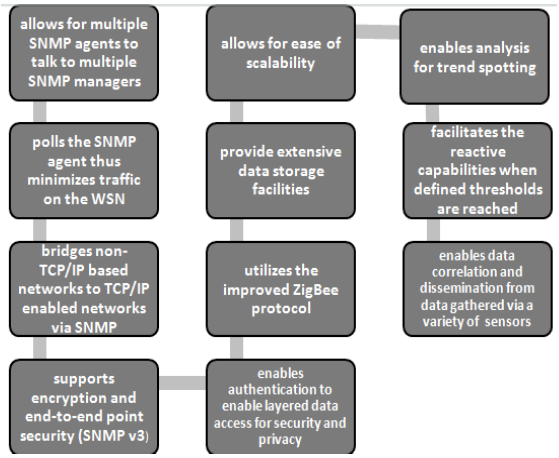

allows for multiple SNMP agents to talk to multiple SNMP managers
allows for ease of scalability
enables analysis for trend spotting
polls the SNMP agent thus minimizes traffic on the WSN
provide extensive data storage facilities
facilitates the reactive capabilities when defined thresholds are reached
bridges non-TCP/IP based networks to TCP/IP enabled networks via SNMP
utilizes the improved ZigBee protocol
enables data correlation and dissemination from data gathered via a variety of sensors
supports encryption and end-to-end point security (SNMP v3)
enables authentication to enable layered data access for security and privacy

Figure 11. Advantages of Medical MoteCare System [2].

VI. Scenario Setting

In this section, a typical clinical scenario that serves for the testing and simulation of the system is described. A middle aged woman has recently recovered from the after-effects of a severe car accident. She had to spend two months in hospital and has now returned to her home. She immediately began to experience flashback and panic attacks as she relived her accident. The doctor suspected tachycardia and suggested she wore a pulse oximeter while resting during the afternoon and at night so that he can monitor her condition and ascertain whether the medication he has prescribed is succeeding in regulating her heart rate. The doctor has asked her to use the pulse oximeter for one week so that he can establish enough evidence to decide whether he needs to change her medication or test for other conditions such as Wolff Parkinson White syndrome. Wolff-Parkinson-White syndrome (WPW) represents an abnormality of the heart's electrical system. In patients with WPW, there is an extra electrical connection between the atria (the upper chambers) and the ventricles (the lower chambers). This abnormal electrical connection can cause episodes of rapid heart rhythms called paroxysmal atrial tachycardia, pre-existed tachycardia, or pre-existed atrial fibrillation [17]. The aim of the monitoring is to ascertain if the heart rhythm reverts to normality during this week of monitoring and medication.

VII. Testing

In order to test the above scenario in a laboratory setting the author set up the following implementation using five (5) participants to test the Medical MoteCare system (refer to Fig. 10). The participants ranged in age from 20 to over 60 years old.

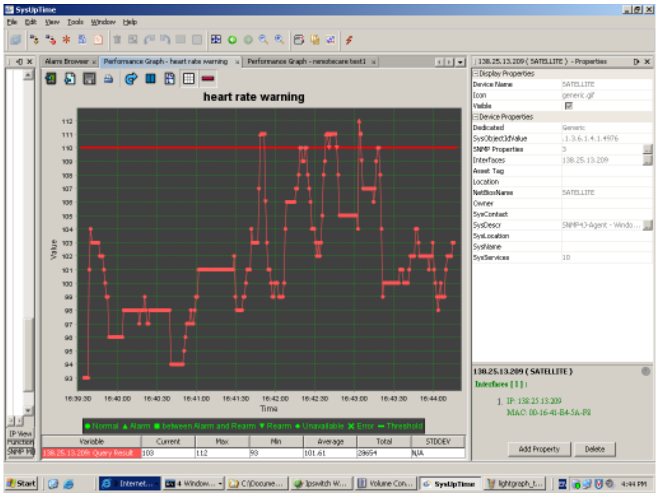
With the testing, the author was attempting to establish that the system would detect the variations in the heart rate and blood oxygen levels as if the monitored person was starting to experience a panic attack. The system would send a critical alarm via a red flashing screen and an email to the carer notifying them that a particular threshold was reached. The author tested the system on five participants performing a variety of activities that could change the heart rate while connected to the system in a testing setting as described below.

The system captured the readings of the heart rate and blood oxygen levels as the participants carried out the different activities. Furthermore the system was expected to react to predefined thresholds and alert the carers to potential problems. The system was able react to a series of correlated thresholds – for example if the heart rate is over a particular threshold but the blood oxygen level is normal a warning notification is generated whereas if the heart rate is high and the blood oxygen is low a critical alarm is produced.

A. *Experiments to Simulate Increased Heart Rate*

In this scenario, two testers were each attached to a pulse oximeter and given the following instructions. For the first minute they had to sit calmly while the resting heart rate figure was established. They then were asked to stand and walk on the spot slowly while still attached to the system. Pulse oximeters are supposed to be used on a patient at rest but to increase the heart rate artificially, the researcher had the testers walk quickly on the spot. At the end of these two minutes they were instructed to walk quickly for two minutes. The graph in Fig. 12 below shows the changes in the heart rate with the threshold marked to show when the system was to react – in this case, for the purposes of testing, the figure of 110 bpm was selected.

Fig. 12 illustrates the results of one experiment where the heart rate went over the 110 bpm and the alarm trigger screenshot is shown in Fig. 13.



Variable	Current	Max	Min	Average	Total	STDDEV
138.25.13.209 Query Result	103	112	93	101.61	28654	N/A

138.25.13.209 (SATELLITE) - Properties

⊟ Display Properties

Device Name	SATELLITE
Icon	generic.gif
Visible	☑

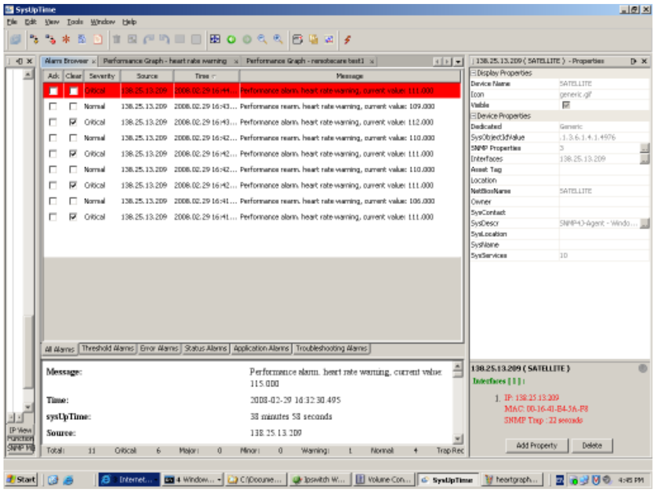
⊟ Device Properties

Dedicated	Generic
SysObjectIdValue	.1.3.6.1.4.1.4976
SNMP Properties	3
Interfaces	138.25.13.209
Asset Tag	
Location	
NetBiosName	SATELLITE
Owner	
SysContact	
SysDescr	SNMP4J-Agent - Windo...
SysLocation	
SysName	
SysServices	10

138.25.13.209 (SATELLITE)
Interfaces [1] :
1. IP: 138.25.13.209
MAC: 00-16-41-B4-5A-F8

Figure 12. Graph from iReasoning MIB Browser.

In Fig. 13 a list of triggered alarms, illustrating different levels of severity, is shown. In the top row alarm, a critical event, highlighted in red, is shown. Column Severity can indicate 'Critical', 'Normal' or 'Warning'. The next column indicates the IP address of the source where the event was polled. Time and date stamps are indicated in the next columns and in the last column, the current value of the heart rate plus a description of the alarm is located.



Ack	Clear	Severity	Source	Time	Message
□	□	Critical	138.25.13.209	2008.02.29 16:44...	Performance alarm. heart rate warning, current value: 111.000
□	□	Normal	138.25.13.209	2008.02.29 16:43...	Performance rearm. heart rate warning, current value: 109.000
□	☑	Critical	138.25.13.209	2008.02.29 16:43...	Performance alarm. heart rate warning, current value: 112.000
□	□	Normal	138.25.13.209	2008.02.29 16:42...	Performance rearm. heart rate warning, current value: 110.000
□	☑	Critical	138.25.13.209	2008.02.29 16:42...	Performance alarm. heart rate warning, current value: 111.000
□	□	Normal	138.25.13.209	2008.02.29 16:42...	Performance rearm. heart rate warning, current value: 110.000
□	☑	Critical	138.25.13.209	2008.02.29 16:42...	Performance alarm. heart rate warning, current value: 111.000
□	□	Normal	138.25.13.209	2008.02.29 16:41...	Performance rearm. heart rate warning, current value: 106.000
□	☑	Critical	138.25.13.209	2008.02.29 16:41...	Performance alarm. heart rate warning, current value: 111.000

Message:	Performance alarm. heart rate warning, current value 115.000
Time:	2008-02-29 16:32:30.495
sysUpTime:	38 minutes 58 seconds
Source:	138.25.13.209

Total: 11 Critical: 6 Major: 0 Minor: 0 Warning: 1 Normal: 4 Trap Rec

138.25.13.209 (SATELLITE) - Properties

⊟ Display Properties

Device Name	SATELLITE
Icon	generic.gif
Visible	☑

⊟ Device Properties

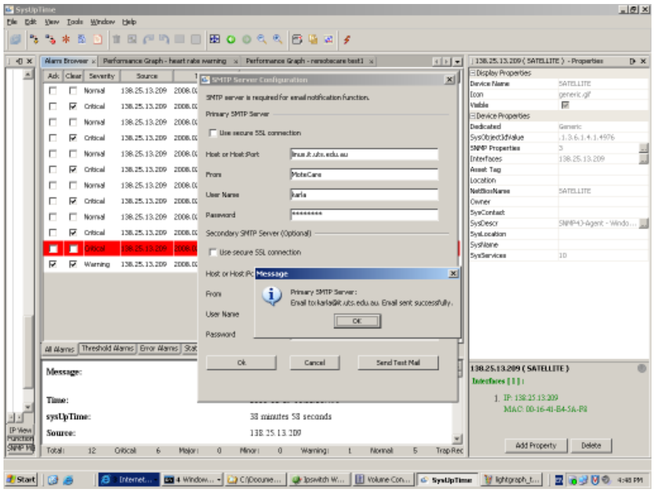
Dedicated	Generic
SysObjectIdValue	.1.3.6.1.4.1.4976
SNMP Properties	3
Interfaces	138.25.13.209
Asset Tag	
Location	
NetBiosName	SATELLITE
Owner	
SysContact	
SysDescr	SNMP4J-Agent - Windo...
SysLocation	
SysName	
SysServices	10

138.25.13.209 (SATELLITE)
Interfaces [1] :
1. IP: 138.25.13.209
MAC: 00-16-41-B4-5A-F8
SNMP Trap : 22 seconds

Figure 13. SysUpTime Network Monitor Alarm Triggering.

The screen shot of Fig. 14 shows the Simple Mail Transfer Protocol (SMTP) configuration window used by

SysUpTime Network Monitor to send email alerts once a preset threshold has been reached.


SysUpTime

File Edit View Tools Window Help
Alarm Browser
Performance Graph - heart rate warning
Performance Graph - remotecare test1

Ack	Clear	Severity	Source	
□	□	Normal	138.25.13.209	2008.0
□	☑	Critical	138.25.13.209	2008.0
□	□	Normal	138.25.13.209	2008.0
□	☑	Critical	138.25.13.209	2008.0
□	□	Normal	138.25.13.209	2008.0
□	☑	Critical	138.25.13.209	2008.0
□	□	Normal	138.25.13.209	2008.0
□	☑	Critical	138.25.13.209	2008.0
□	□	Normal	138.25.13.209	2008.0
□	☑	Critical	138.25.13.209	2008.0
□	□	Critical	138.25.13.209	2008.0
☑	☑	Warning	138.25.13.209	2008.0

All Alarms
Threshold Alarms
Error Alarms
Stat

Message:
Time:
sysUpTime: 38 minutes 58 seconds
Source: 138.25.13.209

Total: 12 Critical: 6 Major: 0 Minor: 0 Warning: 1 Normal: 5 Trap Rec

SMTP Server Configuration

SMTP server is required for email notification function.

Primary SMTP Server

□ Use secure SSL connection

Host or Host:Port: linus.it.uts.edu.au
From: MoteCare
User Name: karla
Password: ********

Secondary SMTP Server (Optional)

□ Use secure SSL connection

Host or Host:Po
From
User Name
Password

Message

Primary SMTP Server:
Email to:karla@it.uts.edu.au. Email sent successfully.

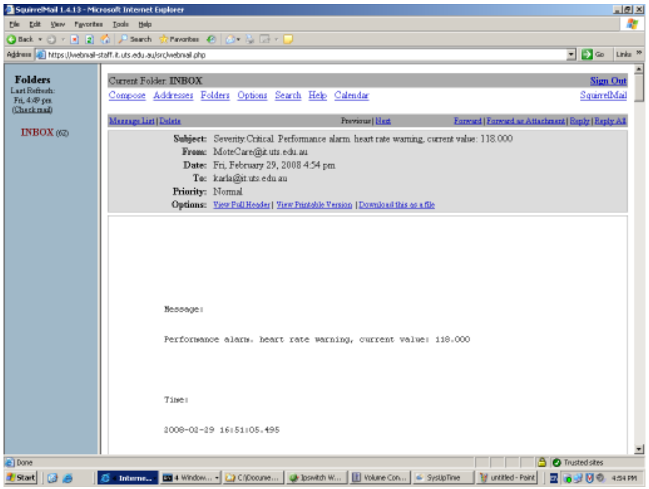
138.25.13.209 (SATELLITE) - Properties

⊟ Display Properties
Device Name: SATELLITE
Icon: generic.gif
Visible: ☑
⊟ Device Properties
Dedicated: Generic
SysObjectIdValue: .1.3.6.1.4.1.4976
SNMP Properties: 3
Interfaces: 138.25.13.209
Asset Tag
Location
NetBiosName: SATELLITE
Owner
SysContact
SysDescr: SNMP4J-Agent - Windo...
SysLocation
SysName
SysServices: 10

138.25.13.209 (SATELLITE)
Interfaces [1] :
1. IP: 138.25.13.209
MAC: 00-16-41-B4-5A-F8

Figure 14. SysUpTime Network Monitor Email System Alert.

Fig. 15 shows the email sent to the carer indicating that a heart rate warning has been triggered and shows the current heart rate value (in this case 118 – well over the threshold of 110 bpm)


SquirrelMail 1.4.13 - Microsoft Internet Explorer

File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links

https://webmail-staff.it.uts.edu.au/src/webmail.php

Folders
Last Refresh:
Fri, 4:49 pm
(Check mail)

INBOX (62)

Current Folder: INBOX

Sign Out
Compose Addresses Folders Options Search Help Calendar
SquirrelMail
Message List | Delete
Previous | Next
Forward | Forward as Attachment | Reply | Reply All

Subject: Severity:Critical. Performance alarm. heart rate warning, current value: 118.000
From: MoteCare@it.uts.edu.au
Date: Fri, February 29, 2008 4:54 pm
To: karla@it.uts.edu.au
Priority: Normal
Options: View Full Header | View Printable Version | Download this as a file

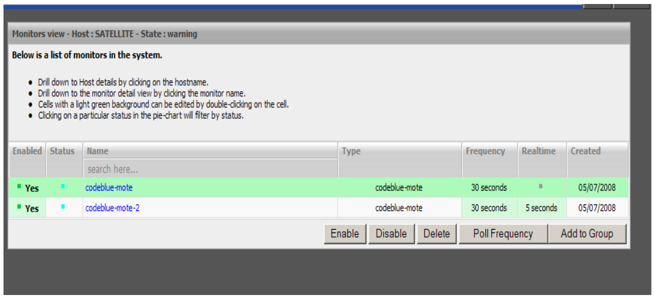
```
Message:

Performance alarm. heart rate warning, current value: 118.000

Time:

2008-02-29 16:51:05.495
```

Figure 15. SysUpTime Network Monitor Email System Alert Scenario to Simulate Increased Heart Rate.

The author was able to demonstrate that the Medical MoteCare was capable achieving the requirements of monitoring healthcare variables via network management tools. However to make the system more robust and useful the author utilized Jaguar SX to demonstrate event correlation for more than one healthcare variable.

B. *Data Fusion with NM Event Correlation*

The Medical MoteCare system implemented Jaguar SX by incorporating a customized MIB in the system engine of Jaguar SX and by performing the appropriate discovery and polling of the medical and environmental motes via the SNMP proxy. See Fig. 16 for a view of the discovered system devices via JaguarSX and refer to Fig. 17 for a graphical view of the collected medical data in real time.

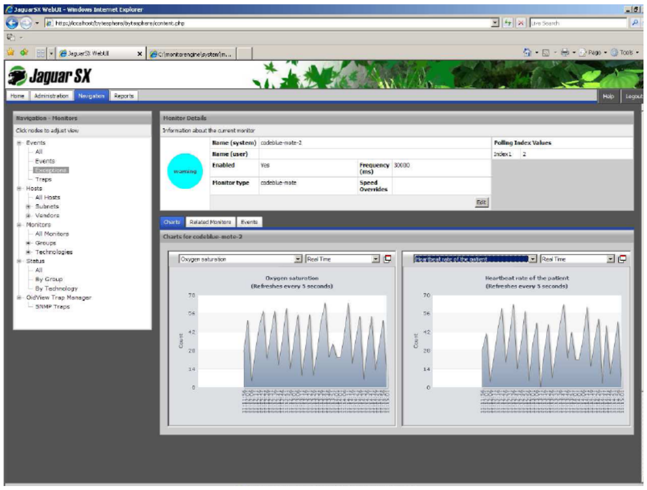

Monitors view - Host : SATELLITE - State : warning

Below is a list of monitors in the system.

- Drill down to Host details by clicking on the hostname.
- Drill down to the monitor detail view by clicking the monitor name.
- Cells with a light green background can be edited by double-clicking on the cell.
- Clicking on a particular status in the pie-chart will filter by status.

Enabled	Status	Name	Type	Frequency	Realtime	Created
		search here...				
■ Yes	■	codeblue-mote	codeblue-mote	30 seconds	■	05/07/2008
■ Yes	■	codeblue-mote-2	codeblue-mote	30 seconds	5 seconds	05/07/2008

Figure 16. JaguarSX WebUI Discovery of the Motes.


JaguarSX WebUI - Windows Internet Explorer

http://localhost/bytesphere/bytesphere/content.php

Navigation - Monitors

Click nodes to adjust view

- Events
  - All
  - Events
  - Exceptions
  - Traps
- Hosts
  - All Hosts
  - Subnets
  - Vendors
- Monitors
  - All Monitors
  - Groups
  - Technologies
- Status
  - All
  - By Group
  - By Technology
- OidView Trap Manager
  - SNMP Traps

Monitor Details

Information about the current monitor

Name (system): codeblue-mote-2
Name (user):
Enabled: Yes
Frequency (ms): 30000
Monitor type: codeblue-mote
Speed Overrides:

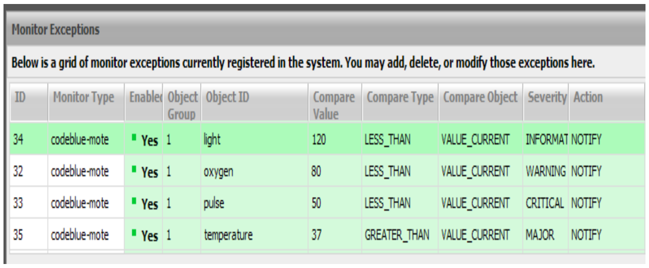
Polling Index Values
Index1: 2

Charts for codeblue-mote-2

Oxygen saturation
Real Time
Oxygen saturation
(Refreshes every 5 seconds)
Count
70
56
42
28
14
0
Heartbeat rate of the patient
Real Time
Heartbeat rate of the patient
(Refreshes every 5 seconds)
Count
70
56
42
28
14
0

Figure 17. JaguarSX WebUI Data Collection of Heart Rate and Oxygen Level in Real Time.

As can be seen in Fig. 17 both blood oxygen level and heart rate measurements are being measured in real time with the resulting graphs being refreshed every five seconds. A false sensor was used to generate extreme variations of data to test the responsiveness of the system to sudden fluctuations in the monitored data. There may be limitations regarding security here. As in real world scenarios, there could be falsification of measurement values. If falsified data is delivered to the sensor (like higher temperatures, etc.) a consistency check may need to be done. This will be explored further in our next iteration. The Monitor Exceptions module of Jaguar SX was used to setup the conditions for multiple variable event correlation. In Fig.8 the monitor exceptions from the variables oxygen, pulse, light and temperature have been configured to trigger an alarm in the event of their preset conditions, corresponding to the "Compare Values" in column six and "Compare Types" in column seven.



Monitor Exceptions

Below is a grid of monitor exceptions currently registered in the system. You may add, delete, or modify those exceptions here.

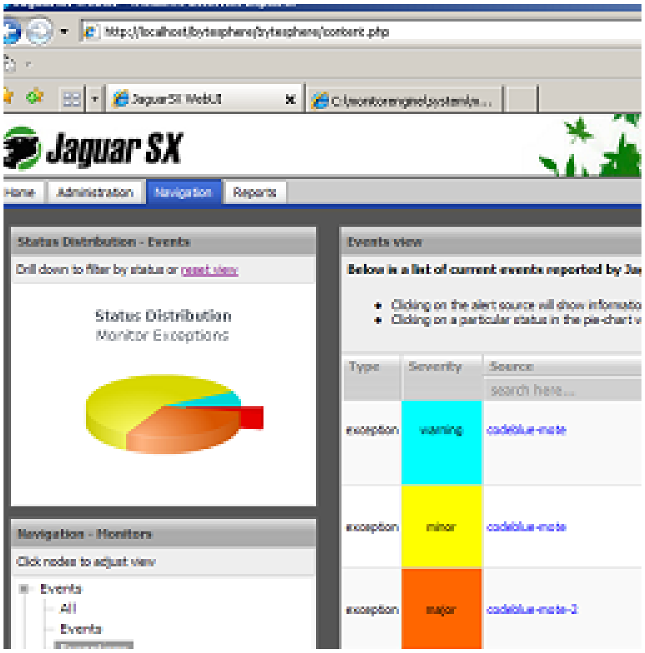
ID	Monitor Type	Enablec	Object Group	Object ID	Compare Value	Compare Type	Compare Object	Severity	Action
34	codeblue-mote	Yes	1	light	120	LESS_THAN	VALUE_CURRENT	INFORMAT	NOTIFY
32	codeblue-mote	Yes	1	oxygen	80	LESS_THAN	VALUE_CURRENT	WARNING	NOTIFY
33	codeblue-mote	Yes	1	pulse	50	LESS_THAN	VALUE_CURRENT	CRITICAL	NOTIFY
35	codeblue-mote	Yes	1	temperature	37	GREATER_THAN	VALUE_CURRENT	MAJOR	NOTIFY

Figure 18. Jaguar SX WebUI Monitor Exceptions Configuration Page.

Alarm triggering is colour coded depending of its level of criticality, for example a critical alarm is bright red whereas a minor or warning alarm is colored yellow or blue respectively (Fig. 19).

As can be seen in Fig. 20, the Medical MoteCare system triggering alarm module from the Jaguar SX window shows a diverse set of alarm notifications generated from preset multi-variable conditions. For example, long term reports show a patient's history of oxygen saturation levels for a predefined period of time, such as, in this case for the last hour. Different types of graphs can be selected by the medical personnel as can a number of different reporting styles. We are now able to customize the triggers and alarms of the Jaguar SX system to individualize alarm triggering, based on a patient's medical history and their current medical condition.

With a ubiquitous Medical Monitoring System, we can use data mining techniques to create models and templates to assist in the prediction, prevention and treatment of a wide variety of possible medical conditions. This will form the focus of our ongoing work.



Status Distribution - Events

Drill down to filter by status or reset view

Click nodes to adjust view

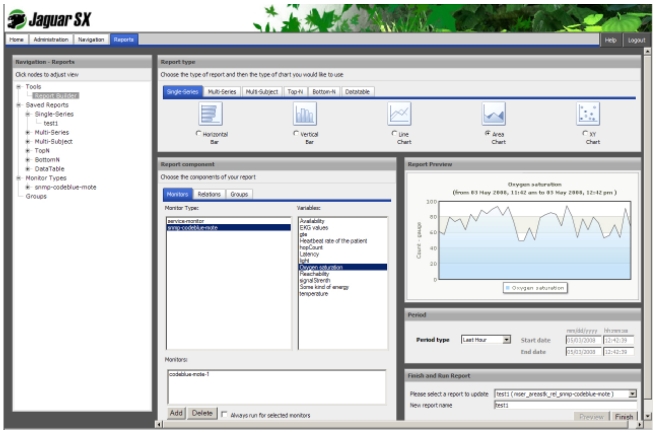
Events view

Below is a list of current events reported by Ja

- Clicking on the alert source will show informatio
- Clicking on a particular status in the pie-chart v

Type	Severity	Source
		search here...
exception	warning	codeblue-mote
exception	minor	codeblue-mote
exception	major	codeblue-mote-2

Figure 19. Jaguar SX Alarm Triggering for Heart Rate & Oxygen Level.



Click nodes to adjust view

Choose the type of report and then the type of chart you would like to use

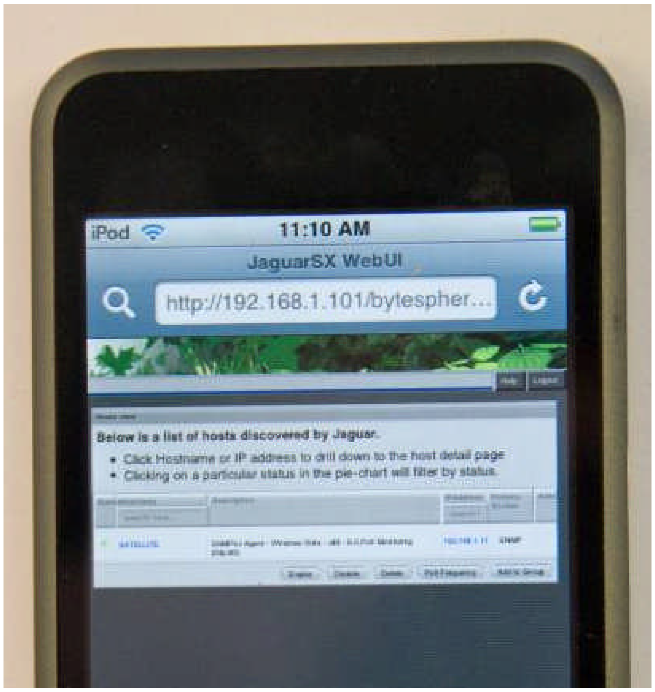
Choose the components of your report

Please select a report to update: test1 (mser_areastk_rel_snmp-codeblue-mote)

New report name: test1

Figure 20. Jaguar SX Reports Module for Oxygen Saturation.

To further illustrate the versatility of the system, the users can access the Medical MoteCare system through the use of a PDA or a smart phone (Fig. 21). This would enable carers to move around while still being able to monitor patients remotely.



Below is a list of hosts discovered by Jaguar.

- Click Hostname or IP address to drill down to the host detail page
- Clicking on a particular status in the pie-chart will filter by status

Figure 21. Mobile Medical MoteCare.

Fig. 21 shows that JaguarSX has discovered the Medical MoteCare system via SNMP and is able to push this information to a smart phone or PDA.

VIII. Conclusion

The Medical MoteCare system demonstrates the feasibility of using traditional network management systems for monitoring personal health parameters in an inherent NM distributed environment. The incorporation of medical sensors to the prototype allowed for the feasibility testing of the system for the handling of medical data. The use of an SNMP agent and a tailored MIB enhanced the scalability, modularity and flexibility of the system by potentially bringing to the researcher or developer communities the freedom of selecting from a vast range of existing SNMP based NM tools to fit their specific WSN application requirements. This was exhibited by the implementation and testing of NM tools of varied complexity and purpose, namely iReasoning MIB Browser, SysUpTime Network Monitor and JaguarSX. The Medical MoteCare System developed is a proof of concept that the use of standard network management tools and models. iReasoning MIB Browser, SysUpTime Network Monitor and Jaguar SX can assist in the development and implementation of robust distributed, wireless sensor networking applications for monitoring healthcare parameters for patients. This NM tool and, consequently any other NM tool that utilizes the SNMP NM standard protocol, can be successfully integrated into WSN applications, and, in this case, of a personal health care monitoring nature. The implementation of JaguarSX added intelligence to the system by utilizing NM correlation techniques to interpret the collected events automatically and react, or sometimes even anticipate from the collected statistical data, harmful health conditions.

Acknowledgment

The authors would like to thank to the present and former research assistants, group members and research collaborators of the multicultural M-Health Research Group in the Faculty of Engineering and Information Technology at UTS for their hard work, team spirit and brilliant contributions, Professor Elaine Lawrence, Josep Riudavets, Marco Messina, Einstein Lubrin, Brian Lim, Leena Kowser Ganguli, Martin Fischer from Koblenz University and Dr. Frank Kargl from the Institute of Media Informatics at Ulm University in Germany. Particular thanks must also go to Professor Matt Welsh from the School of Engineering and Applied Sciences at Harvard University, Nicholas Saparoff from ByteSphere and Professor Victor C.M. Leung from the University of British Columbia in Canada for their invaluable collaborations in the accomplishing of the research objectives of this project.

References

[1] Lubrin, E., Lawrence, E. & Navarro, K.F.: Wireless Remote Healthcare Monitoring with Motes. IEEE International Conference on Mobile Business 2005, Sydney. pp. 235 - 241. 2005,

[2] Lim, YY., Messina, M., Kargl,F., Ganguli, L, Fischer,M. and Tsang,T.:, SNMP-Proxy For Wireless Sensor Network, IEEE ITNG Conference, Las Vegas, 7-9 April, 2008

[3] Fischer, M., Lim, Y.Y. Lawrence, E., Ganguli, L.: ReMoteCare: Health Monitoring with Streaming Video, IEEE International Conference on mBusiness, Barcelona, July 9-11, 2008.

[4] Felix Navarro, K, Lawrence, E and 2. Lim, B.: Medical MoteCare: A Distributed Personal Healthcare Monitoring System, submitted to IEEE Telehealth Conference, IEEE Iaria, Cancun 2009

[5] Dipert, B. The Human Touch keeps the Elderly and Disabled Technology connected. EDN, pp. 47-57. 2004 From: www.edn.com

[6] Messina, M., Lawrence ,E., Felix Navarro, K and Lubrin, E.: Towards Assistive Healthcare: Prototyping Advances In Wireless Sensor Network (WSN) Systems Integration And Applications, IADIS International Conference e-Society 2007 Lisbon, Portugal 3-6 July 2007

[7] Peter Leijdekkers, Valérie Gay, Elaine Lawrence: Smart Homecare System for Health Tele-monitoring. IEEE International Conference on Digital Societies CDS 2007: 3 Guadaloupe, January 3-9 2007

[8] Healthservice24. (2007).
http://www.healthservice24.com/healthservice24/images//HS24_Project%20Presentation.pdf.

[9] Hoang, Doan and Lawrence, Elaine: An Active Grid Infrastructure for Elderly Care, Tromso telemedicine and eHealth Conference (TTeC08) Tromso, Norway 9-11 June 2008

[10] Crossbow Technology Inc. 2005, [http://www.xbow.com].

[11] Malan, D., Fulford-Jones, T., Welsh,,M and Moulton, S.: (CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care, In Proceedings of the MobiSys 2004 Workshop on Applications of Mobile Embedded Systems (WAMES 2004), Boston, MA, June, 2004. (Also presented at the International Workshop on Wearable and Implantable Body Sensor Networks, London, April 2004.)

[12] Subramanian, M. Network Management:Principles and Practice, Addison Wesley, 2000.

[13] Lawrence, E.M., Felix Navarro, K.M., Hoang, D.B. & Lim, Y., 'Data Collection, Correlation and Dissemination of Medical Sensor Information in a WSN', Fifth International Conference on Networking and Services Valencia, IEEE, Australia, pp. 402-408, April 2009.

[14] Felix Navarro, K.M., Lawrence, E.M. & Lim, B., 'Medical Motecare : A Distributed Personal Healthcare Monitoring System', International Conference on eHealth, Telemedicine, and Social Medicine eTELEMED 2009, Cancun, IEEE Computer Society Conference Publications, Piscataway, USA, pp. 25-30, February 2009.

[15] Anon, .A Brief Guide to OIDs, middleware.internet2.edu/a-brief-guide-to OIDs.doc

[16] Felix Navarro, K, Lawrence, E & Lubrin, E, ‘A network management approach to wireless sensor networks in personal healthcare environments’, Communication Systems and Networks, p. 543, 2006.

[17] Medicinenet, http://www.medicinenet.com/wolff-parkinson-white_syndrome/article.htm, 2007.

Design and Implementation of Multi-User Wireless Body Sensor Networks

José A. Afonso, Pedro Macedo, Helder D. Silva, José H. Correia, Luis A. Rocha
Department of Industrial Electronics
University of Minho
Guimarães, Portugal
{jose.afonso, pmacedo, dsilva, higino.correia, lrocha}@dei.uminho.pt

Abstract—**This paper describes the development of two low power wireless body sensor network (BSN) prototypes that allow the real-time monitoring of wearable sensors data from multiple users with a single base station. The prototypes present differences in the architecture, wireless network hardware and implemented protocols. These systems integrate inertial sensors that allow the monitoring of users' posture and goniometric development, as well as heart rate and respiratory rate sensors. The design of body sensor networks is challenging due to the requirements of quality of service provisioning and low energy consumption, combined in many cases with high traffic loads. The solutions presented in this paper address these challenges, with particular focus on the implemented medium access control protocols, which overcome some drawbacks presented by standard protocols when applied to BSN scenarios. The performance of the developed systems is presented, compared and discussed using theoretical, experimental and simulation methods. Results show that the proposed mechanisms provide good bandwidth efficiency and decrease the delivery error rate without significant increase in the energy consumption.**

Keywords-body sensor networks, medium access control, wireless sensor networks, quality of service

I. Introduction

A body sensor network (BSN) is characterized by the presence of sensors nodes attached to a user's body in order to acquire physiological signals and a central monitoring unit, which receives the data from the sensors, either directly or through intermediate nodes. These networks may allow the monitoring of several kinds of signals, such as electrocardiogram (ECG), oximetry, temperature, heart rate, blood pressure and respiratory rate, as well as the motion of segments of the body. Conventional wired BSN monitors have been used in places like hospitals over the last decades; however, these systems impose significant restrictions on the users' mobility.

Recent advances in microelectronics, wireless communications and sensors technology, with the corresponding reduction in size and cost, foresee a rapid expansion on the development of wireless body sensor networks over the next years, enabling mobile and continuous monitoring of users even during their daily life activities, with applications in the areas of healthcare, sports and entertainment.

In the healthcare area, although extensive physiological data monitoring is currently available in many hospitals, the monitoring and diagnostic procedure is normally limited in time, restraining the capability of these systems to properly capture the patients' physiological states, since transient abnormalities cannot always be detected in this scenario. For example, many cardiac diseases are associated with relevant episodic abnormalities such as arrhythmias, episodes of myocardial ischemia or transient surges in blood pressure [2]. Since the timing of these abnormalities is impossible to predict, much time and effort is wasted when trying to capture an episode with controlled monitoring using wired systems, making many vital and even life threatening disorders go undetected due to their random behavior. Wireless body sensor networks facilitate the continuous wireless monitoring of users, having the potential to replace life saving but expensive therapies. It is expected that these systems will be used initially to monitor at-risk patients, with their use being gradually extended to benefit a larger proportion of the population.

Despite of the benefits that can result from the adoption of wireless BSNs, there are some challenges that still limit their widespread application. Normally these systems need to satisfy demanding requirements in terms of quality of service (QoS), such as reliable data delivery, sustainable throughput and bounded delay. At the same time, since sensor nodes are normally battery powered, it is necessary to minimize their energy consumption to increase their lifetime. A challenge arises from the fact that some kinds of sensors must be sampled quite often, generating a great amount of data and, consequently, requiring the network to operate under high loads. For example, a motion capture system based in inertial sensors placed on the body needs to sample multiple sensors several times per second to provide satisfactory results (30 frames per second in many applications). Another example is ECG monitoring, which can require sampling rates of up to 250 Hz per lead with a resolution of 12 bits [3]. If contention-based MAC (Medium Access Control) protocols typically proposed for wireless sensor networks are used, the high traffic load generated by these data-intensive applications will lead to frequent collisions between sensor nodes.

This paper describes the design and development of two multi-user wireless BSN prototypes that enable the simultaneous monitoring of the posture and movements of several users in real-time with a single central monitoring unit (base station). An application of this systems is the monitoring of teams of athletes in a gymnasium for sports such as basketball, volleyball, handball and gymnastics, with the goal of providing detailed information in order to enhance the performance of both the athletes individually and the team as a whole. Another application is in the medical field, namely in physiotherapy sessions, where such a system can benefit both the patient, by increasing his levels of confidence, and the therapist, by providing detailed information about the patient evolution. The first system prototype also implements sensors to measure the heart rate and respiratory rate.

This paper is an extended version of [1]. Compared to the original paper, it includes a section about wireless network standards and extends the description of the first developed prototype. In addition, this paper also includes the description of a new prototype, comprising an overview of its design characteristics, the presentation of its enhanced MAC protocol and the presentation and discussion of simulation results comparing the performance of the protocols used by both prototypes.

Next section presents an overview of current wireless standards and discusses their suitability to BSN applications. Section III presents the architecture of the first prototype, the implemented MAC protocol (LPRT) and the developed PC application, as well as a theoretical analysis and experimental results. Section IV presents the architecture of the second prototype, describes its MAC protocol (iLPRT) and presents simulation results. Finally, Section V presents the conclusions.

II. Wireless Network Standards

Most wireless networks can be classified in the following five categories: wireless local networks (WLAN), wireless personal networks (WPAN), satellite networks, mobile cellular networks and broadband wireless access (BWA) networks. Some characteristics of former two differ substantially from the characteristics of the latter three. WLAN and WPAN are relatively short range technologies that normally work in unlicensed frequency bands and where all the network equipment normally belongs to the user. On the other hand, the latter networks tend to work in licensed bands, with the infrastructure belonging to the network operator, which charges the user for the provided services. The characteristics of the former are more convenient for the applications considered in this paper, so those networks will be discussed further in the remaining of this section.

The most common WLANs available in the market are based on the IEEE 802.11 standards [4]. WLANs present higher range and data rates than most WPANs. However, they normally present also higher energy consumption and large protocol overhead with short data packets, which make them less adequate to be used with battery-operated body sensor network nodes. For example, the MAC header and trailer of 802.11 data frames occupy a total of 34 bytes, which represents a significant overhead given that the payload generated by sensor applications is normally composed of few bytes.

Bluetooth [5] is a WPAN technology that works in the 2.4 GHz band using frequency hopping spread spectrum (FHSS). The network, called piconet, is organized in a star topology composed by one central node (master) and a maximum of seven active end nodes (slaves). A piconet has a range from 10 to 100 m, depending on the class of the device. Piconets can be interconnected to form a scatternet. Bluetooth uses a polling based MAC protocol that provides support for both real-time and asynchronous traffic. However, the restriction of only seven slaves per piconet imposes a severe limitation on the number of wireless nodes of the BSN system. While the scatternet functionality provides a way to increase the number of the nodes of the network, its multi-hop multi-channel and single radio architecture tends to increase the complexity and decrease the performance of the network, making it harder to provide QoS guarantees to the application. Another obstacle for the implementation of large scale BSNs based on Bluetooth is that few commercially available devices are able to support the seven slaves per piconet specified by the standard, and even less, if any, are able to properly support scatternets.

The standardization of WPAN networks at the IEEE is under the scope of the 802.15 group. One of these standards, the IEEE 802.15.4 [6] defines the physical and MAC layer of ZigBee [7], which targets wireless sensor network (WSN) applications through the provision of low power and low bit rate WPANs. At the physical layer, the IEEE 802.15.4 relies on direct sequence spread spectrum (DSSS) to enhance the robustness against interference, providing gross data rates of 20/40 kbps, at the 868/915 band, and 250 kbps, at the 2.4 GHz band. The basic IEEE 802.15.4 MAC protocol is a contention-based CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) mechanism, which is not the most suitable option to provide the QoS required by BSN applications, especially under high loads, due to the increased probability of collisions. The performance of the CSMA/CA mechanism is also negatively affected by collisions due to the hidden node problem.

The IEEE 802.15.4 standard also provides a guaranteed time slot (GTS) scheme in order to support devices requiring dedicated bandwidth and low latency transmission. However, only seven GTS allocations per superframe are supported with this scheme, imposing a limitation on the number of wireless nodes of the BSN system. Besides that, the granularity of the transmission periods that can be allocated to the nodes is coarse, since each superframe is composed by just 16 slots, which can result in wasted bandwidth, reducing the bandwidth efficiency.

Given the drawback of the current wireless networks standards for the BSN applications considered in this paper, we decided to design and implement our own MAC protocols. These implementations are based on low power COTS (Commercial Off-The-Shelf) wireless network components that are compliant with the IEEE 802.15.4

standard at the physical layer, in order to benefit from the economies of scale (low cost) and component integration (small size).

III. First System Prototype

An emerging new area of research that combines the strengths and capabilities of electronics, sensors and textiles is the field of electronic textiles (e-textiles) [8]. If these e-textiles are further combined with wireless communication capabilities, wireless smart textiles systems with local monitoring and computation skills and remote data storage become reality [9].

A. Prototype Overview

The architecture of the first multi-user wireless BSN prototype is composed by a personal computer (PC), a base station and several smart shirts (wireless stations), as depicted in Figure 1. Each shirt aggregates data from a heart rate sensor, a respiratory sensor and several posture sensors, which allow the measurement of the user's posture and goniometric development, and transmits this information continuously to the base station using its wireless station. Microcontroller, power supply, RF transceiver and antenna are also attached to the shirt. The base station is connected to a PC, where the data collected from the shirts can be stored and visualized in real-time.

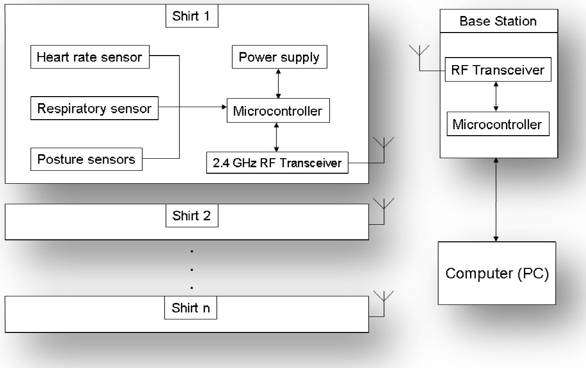


Figure 1. Architecture of the first prototype.

An ear clip infrared sensor is used to measure the heart rate, based on the skin reflectivity to infrared light. Since the reflectivity changes with blood pressure, which in turn changes with the pumping action of the heart, the variations in reflectivity are used to measure the heart rate.

The respiratory frequency is measured with inductance plethysmography, which measures cross-sectional area variation and is commonly used to carry out respiration measurements. The respiratory sensor is made of a copper yarn crocheted in a polyester belt, providing an adjustable stretch due to the addition of rubber yarns. If the belt is sewed on the suit around the abdomen or thorax, circumference and length changes of the belt caused by breathing result in an inductance variation, which can be measured and processed.

The posture monitoring system is composed by one module for each segment of the body requiring monitoring. The developed shirt has sensor modules in the spine, hips and arms. The same modules could also be used to measure the angles of other segments of the body, such as the legs, if desired. Each module contains a 3-axis accelerometer and a 3-axis magnetometer that are used to obtain the pitch, roll and yaw angles. Both the gravitational force and the Earth's magnetic field are used to detect the angles of the segments. The former is used to detect inclination while the later is used to measure the rotation of the body about the axis perpendicular to the gravity field. Figure 2 shows the posture sensor module that was developed. Each module is able to provide angles resolution of around 1 degree. The posture monitoring system is described in patent WO 2008/018810 A2 [10].

Both the base station and the shirt stations are implemented using Crossbow MICAz motes [11] (Figure 3). The MICAz integrates a microcontroller (Atmel ATMega128) and a RF transceiver (Chipcon CC2420) that is compliant with IEEE 802.15.4 at the physical layer.

Each shirt can be seen as a wired BSN where the multiple sensor modules are linked to the MICAz mote through the use of an I^2C serial bus. The shirt electronics is powered by two AA batteries.

The base station is attached to a MIB520 interfacing and programming board, which allows the MICAz to communicate with and receive power from the PC through a USB cable.

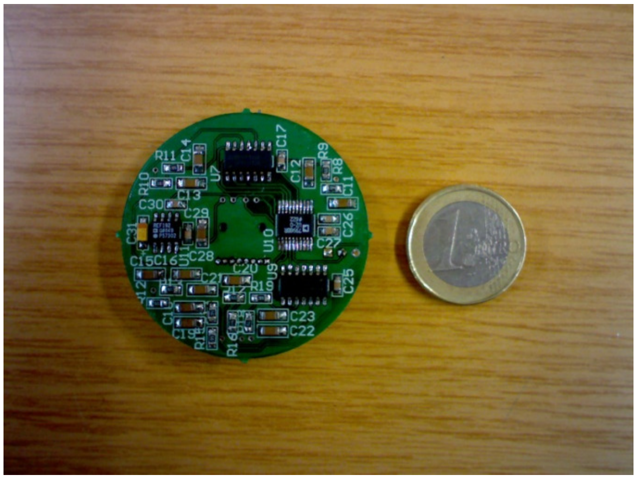


Figure 2. Posture sensor module.

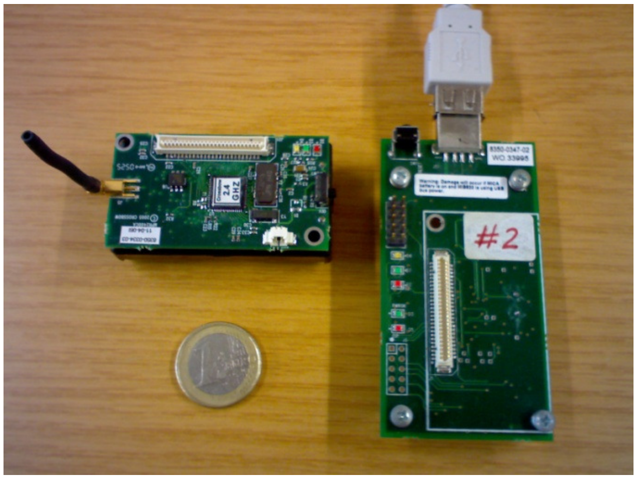


Figure 3. MICAz (left) and MIB520 (right).

B. The LPRT MAC protocol

The LPRT (Low Power Real Time) protocol was implemented in the MICAz motes using the TinyOS operating system and the nesC programming language [12]. LPRT is a hybrid schedule-based dynamic TDMA protocol and contention-based CSMA/CA protocol that is based in the superframe structure presented in Figure 4. Each superframe is divided in a fixed number of mini-slots (1024, in this implementation), and starts with the transmission, by the base station, of the respective beacon frame (B), which is followed by the Contention Period (CP). During the CP any station can transmit using the rules of a CSMA/CA protocol. The CP allows the stations to associate with the base station and to request mini-slots for transmission during the Contention Free Period (CFP). It can also be used to convey non-real-time asynchronous traffic. Stations must not initiate a CSMA/CA transaction if it cannot be completed before the beginning of the CFP.

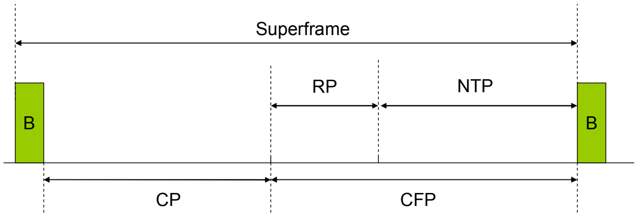


Figure 4. Superframe format of the LPRT protocol.

The Contention Free Period is placed after the CP. Transmissions during the CFP are scheduled by the base station using resource grant (RG) information announced previously in the beacon frame of the current superframe. Since transmissions during the CFP are regulated by the base station, they are not affected by the hidden node problem, unlike transmissions of protocols that rely on carrier sensing.

The CFP is composed by an optional retransmission period (RP) and a normal transmission period (NTP). The RP gives a new opportunity for transmission of data frame when it was not delivered correctly in the previous superframe. The RP is placed after the CP so that a data message is retransmitted away from an eventual burst error condition responsible for the transmission failure occurred during the previous NTP. The RP is placed before the NTP to allow the data message to reach its destination before the next one is transmitted, in order to limit the delay and preserve the sequence of data. This division is not mandatory, however, as retransmissions can be mixed with regular transmissions during the entire CFP period. The retransmission procedure is intended to increase the reliability of the protocol, which is fundamental for applications with low loss tolerance.

MICAz motes use fixed 16 bit long station addresses. Instead of using these addresses in the resource grant (RG) fields of the beacon to identify the stations that are allowed to transmit during the CFP, the LPRT protocol uses a smaller association ID (AID) that is dynamically assigned to the stations when they associate with the base station. This approach reduces the length required for the beacon, making it less sensible to channel errors. The current implementation uses 5 bit long AID addresses, allowing a maximum of 32 associated stations, which is sufficient for the envisioned applications.

Figure 5 shows the structure of the payload of the beacon frame. The superframe duration field gives the duration of the current superframe in multiples of a minimum superframe duration time. It is followed by a list of resource grant (RG) fields, whose quantity is expressed by the RG list length (RLL) field. Each RG is composed by a transmission direction (TD) bit, the association ID (AID) field and an initial transmission slot (ITS) field. The RG allows the scheduling of data transmissions either to or from the station identified by the AID, depending of the value of the TD bit: “0” for downlink and “1” for uplink direction. More than one RG can be granted to a station in the same superframe. The total transmission period granted by a given RG goes from the beginning of respective ITS until before the beginning of the ITS of the next RG on the list. The uplink data can include piggybacked information for alteration of resource allocation parameters, if desired.

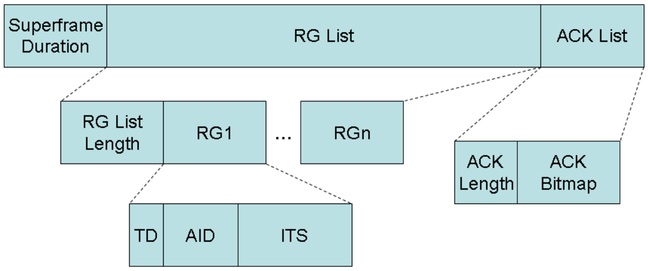


Figure 5. Structure of the beacon frame payload.

For downlink transmissions the ACK frame follows the data transmission, while for uplink transmissions the acknowledgment is made using the ACK list of the next beacon frame. The piggybacking of acknowledgments in the beacon frame eliminates the bandwidth overhead associated with the reception of individual ACK frames for each uplink data frame. As an example, in the 250 kbit/s version of the IEEE 802.15.4 standard, the transmission of an ACK frame (11 bytes) takes 352 µs and the gap between the data and ACK frames can vary from 192 to 512 µs. Even using the minimum gap, the overhead introduced by the use of ACK frames, considering the data frame length used in Section IV-C (43 bytes), would be 40%.

The ACK list is composed by an ACK length (AL) field and an ACK bitmap field containing one bit for each uplink RG of the previous superframe. A successful transmission is indicated by a “1” in the respective bitmap position, while a lost or corrupted transmission is indicated by a “0”.

To reduce the power consumption of the stations, after a station listen to the beacon, it turns off the radio transceiver, turning it on again in the beginning of each scheduled data transmission or reception. After that, it turns off the transceiver again. The station turns on the transceiver again at the beginning of the next superframe to receive a new beacon, and so the cycle repeats. Each time the station receives a beacon it resynchronizes its clock with the base station.

C. Theoretical analysis

This section presents an analysis of the LPRT protocol with uplink data traffic, assuming an AWGN/BSC (Additive White Gaussian Noise/Binary Symmetric Channel) channel, which is later used for comparison with experimental data.

The packet error rate (PER_D) for an uplink data transmission is given by:

$$PER_D = 1 - (1 - BER)^{L_D}, \tag{1}$$

where L_D is the length of the data packet in bits and BER is the channel bit error rate. The packet error rate (PER_B) for a beacon transmission, with L_B as the beacon length in bits, is calculated in a similar way.

The delivery error rate (DER) is defined here as the probability of failure in the delivery of a data message from a given station to the base station. We consider that one data message (e.g., a sample from all sensors of a smart shirt) is generated at each superframe. For the LPRT protocol, the basic frame exchange transaction required for the delivery of an uplink data message consists of the transmission by the base station of a beacon frame at the start of a superframe followed by the transmission by a station of a data frame, at the position announced by the beacon, in the same superframe.

A successful transaction in this case requires the successful transmission of both the beacon and the data frame; therefore the delivery error rate in this case (DER_0) is:

$$DER_0 = 1 - (1 - PER_B)(1 - PER_D)\,. \tag{2}$$

In order to increase the robustness of the communications, the LPRT protocol implements a contention-free retransmission mechanism that works by automatically rescheduling the transmission of a data message in the next superframe whenever the delivery in the current superframe fails. If the attempt of delivery of the data message fails in two consecutive superframes the message is dropped. The improved delivery error rate with one retransmission attempt is:

$$DER_1 = DER_0{}^2\,. \tag{3}$$

One can notice that the rescheduled transmission in the RP can be either a retransmission, if the data transmission in the previous superframe failed, or the first transmission of the data, if instead the beacon in the previously superframe was not received, since it precludes the corresponding data transmission. For the sake of simplicity, however, the rescheduled transmission will continue to be referred here as a retransmission.

The sensors samples included in the data message are taken just before its transmission, so the delay without retransmission is just the data transmission time, while the delay for a retransmitted data message is always less than one superframe period.

The average current consumption of the LPRT protocol without retransmissions (I_0) is given by:

$$I_0 = \frac{T_B + G_B + T_D + G_D}{T_{SF}}(I_{ON} - I_{OFF}) + I_{OFF} \tag{4}$$

where T_B and T_D are the beacon and data transmission times, G_B and G_D are the respective guard times, I_{ON} is the current consumption in the active state and I_{OFF} is the consumption in the sleep state. The guard times are the periods between the transition to the active state and either the effective transmission of data or the reception of the beacon.

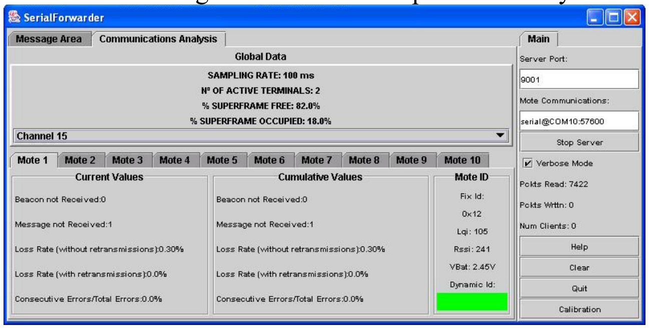
The average current consumption with retransmissions increases by a factor that depends on the DER, and is given by:

$$I_1 = I_0 + \frac{T_D + G_D}{T_{SF}}(I_{ON} - I_{OFF})DER_0 \tag{5}$$

D. PC application

The TinyOS installation CD includes a java application called SerialForwarder that reads packets from the serial port and forwards those through a given TCP port to a network client application, and vice versa. Based on this java application, we developed an application that introduces several new functionalities.

The main window provides information about the state of the connection between the base station and each station in real-time (Figure 6), including the number of beacons not received by a station and the number of data frames not received by the base station, as well as the DER with and without retransmissions. The user interface presents this data both for the current values (evaluated from the last N samples) and cumulative values (since the start of data collection). The receiver signal strength indicator (RSSI) and the battery voltage of the station are also presented. The data can also be registered in a file for posterior analysis.



SAMPLING RATE: 100 ms
Nº OF ACTIVE TERMINALS: 2
% SUPERFRAME FREE: 82.0%
% SUPERFRAME OCCUPIED: 18.0%
Channel 15

Current Values:
Beacon not Received:0
Message not Received:1
Loss Rate (without retransmissions):0.30%
Loss Rate (with retransmissions):0.0%
Consecutive Errors/Total Errors:0.0%

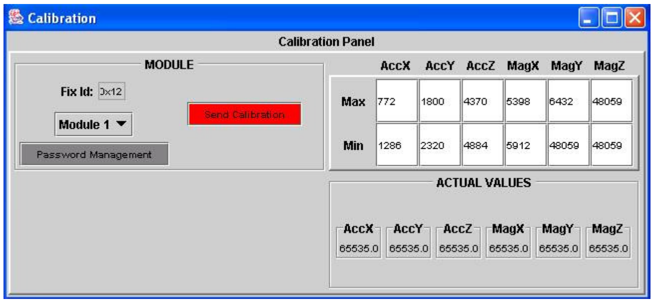
Cumulative Values:
Beacon not Received:0
Message not Received:1
Loss Rate (without retransmissions):0.30%
Loss Rate (with retransmissions):0.0%
Consecutive Errors/Total Errors:0.0%

Mote ID:
Fix Id:
0x12
Lqi: 105
Rssi: 241
VBat: 2.45V
Dynamic Id:

Server Port:
9001
Mote Communications:
serial@COM10:57600
☑ Verbose Mode
Pckts Read: 7422
Pckts Wrttn: 0
Num Clients: 0

Figure 6. Main user interface of the PC application.

The developed PC application also includes a functionality that allows the sensors to be remotely calibrated though the wireless sensor network (Figure 7), thereby avoiding the requirement of physical connection of the data station to a programming board.



Fix Id: 0x12
Module 1

	AccX	AccY	AccZ	MagX	MagY	MagZ
Max	772	1800	4370	5398	6432	48059
Min	1286	2320	4884	5912	48059	48059

ACTUAL VALUES

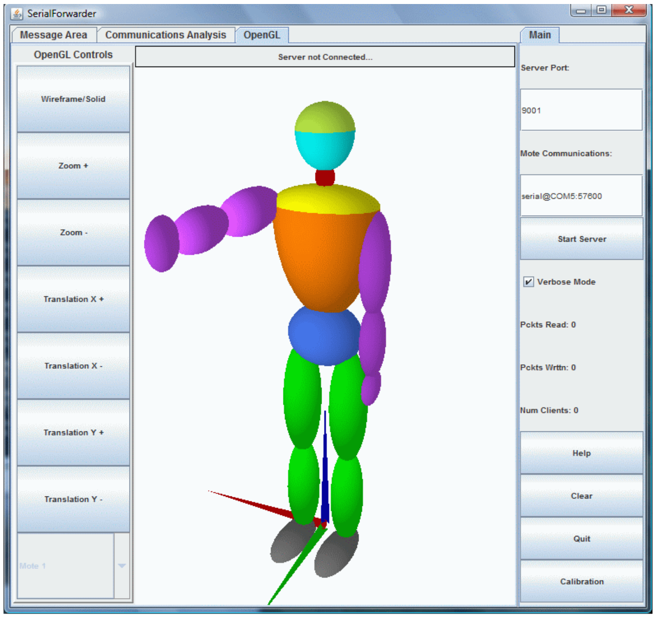
AccX	AccY	AccZ	MagX	MagY	MagZ
65535.0	65535.0	65535.0	65535.0	65535.0	65535.0

Figure 7. Calibration interface.

The application receives the raw posture sensors data through the wireless network, performs the calculation of the angles of the articulations and forwards this information to the client application, which acts as a data base and allows 3D visualization of data. Figure 8 presents an OpenGL application that was developed for real-time visualization of the posture and movements of the users.

E. Experimental results

First results in this section present the current consumption of the MICAz motes as a function of time, which include both the consumption of the transceiver and of the microcontroller. The evaluation scenario is composed by a base station and four stations. Each station collects 36 byte samples (corresponding to the data generated by four posture sensor modules) periodically, with a sampling rate of 5 Hz, which results in a packet inter-arrival time of 200 ms. The superframe period was chosen to be equal to the packet inter-arrival time, so one sample is transmitted per superframe. Figure 9 presents the current consumption for one of the stations. The current consumption of the microcontroller (I_{OFF}) is 8 mA while the consumption with the transceiver turned on (I_{ON}) is around 28 mA.



Server not Connected...
Mote 1
Server Port:
9001
Mote Communications:
serial@COM5:57600
☑ Verbose Mode
Pckts Read: 0
Pckts Wrttn: 0
Num Clients: 0

Figure 8. OpenGL application.

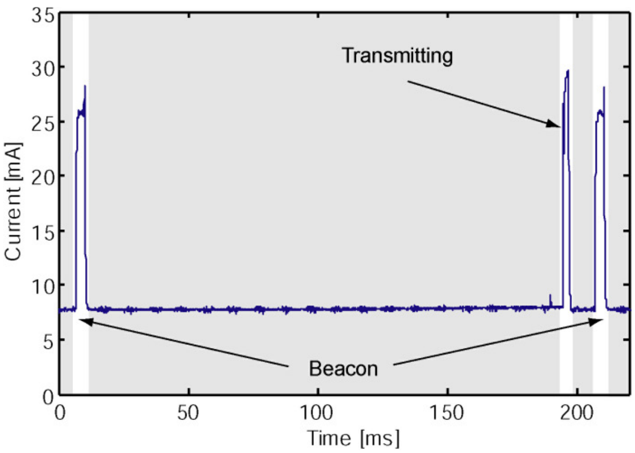


Figure 9. Current consumption of the LPRT protocol as a function of time for one particular station.

Figure 10 presents a closer look to the current consumption charts of the four stations in the RG list, superimposed, where the transmission of each station and the reception of the beacon of the next superframe can be seen. Each superframe is divided in 1024 mini-slots, allowing a high level of granularity in the allocation of transmission time to the stations. With these data frame size and without channel errors, the LPRT protocol allows up to 60 stations per superframe. It can achieve high throughput with high loads without losses due to collisions that are typical of contention based protocols.

The evaluation setup for the second experiment is composed by two networks operating in different channels (11 and 26) at the same time in the same place. Each network is composed by only one base station and one station located 3 meters apart and using the same transmission power (0 dBm). The superframe period is T_{SF} = 100 ms. The samples are 75 bytes long, so the data frame length, including headers is L_D = 712 bits while the beacon length is L_B = 160 bits, which means that the respective transmission times, given that the network data rate is 250 kbit/s, are T_D = 2.85 ms and T_B = 0.64 ms.

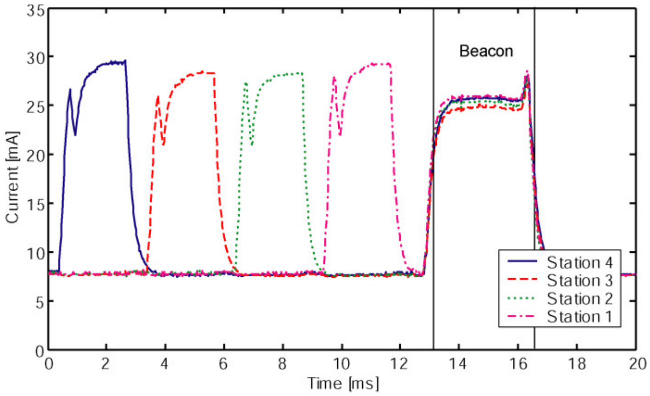


Figure 10. Current consumption of the LPRT protocol as a function of time for four stations.

Each station was powered by two 2300 mAh Ni-MH AA batteries. Both networks started to operate in a Thursday at 17:00h. The channel 11 network ceased to operate when the batteries drained out. The operation of channel 26 network lasted longer but was terminated due to a power outage in the laboratory.

MICAz is compliant with the 2.4 GHz version of the IEEE 802.15.4 standard at the physical layer, which provides 16 non overlapping channels (11 to 26) spaced 5 MHz apart. The center frequency for channel 11 is 2405 MHz, while for channel 26 the center frequency is 2480 MHz.

The signals of three IEEE 802.11 access points operating on channel 1 were detected at the place of the experiment. Channel 1 is centered on 2412 MHz and has a width of 22 MHz, so it overlaps with channel 11 of 802.15.4. 802.11 access points operating in other channels were also detected, but there were no overlapping with 802.15.4 channel 26.

Figure 11 presents the DER as a function of time calculated using the previous 1000 samples. Asterisks in the graph identify disconnections, triggered by several consecutive missing data frames, followed by new associations. The number of errors for the network operating in channel 11 was significantly higher (overall DER without retransmissions was 0.66% for channel 11 and 0.02% for channel 26). These results can be explained by interference from nearby 802.11 networks. In fact, the great majority of errors occurred during working hours.

In order to avoid interference, the user interface of the developed PC application allows the operator to change the operating channel (Figure 6). The information about the new channel is sent on the beacon, allowing a seamless channel transition for the stations. Since the state of the links is monitored in real-time by the application, an automatic channel selection mechanism could be easily implemented as well.

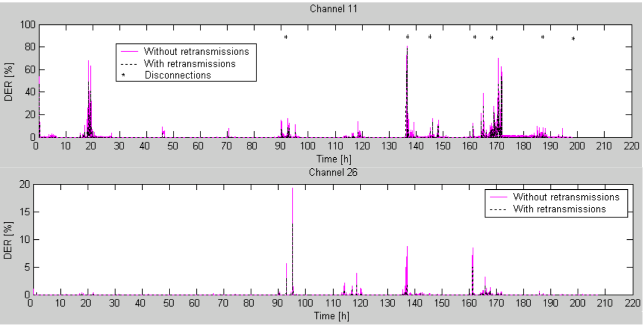


Figure 11. Delivery error rate as a function of time.

Figure 12 presents the relation between the DER with and without retransmissions, comparing the experimental data for channel 11 with the analytical data (equation 3). Even though there is a set of data points around the analytical curve, there is another set of points where the efficacy of the retransmission mechanism is almost none. In this situation, the overall performance is worse than expected, which indicates that future analysis needs to take into account other effects, such as burst errors due to fading and/or interference.

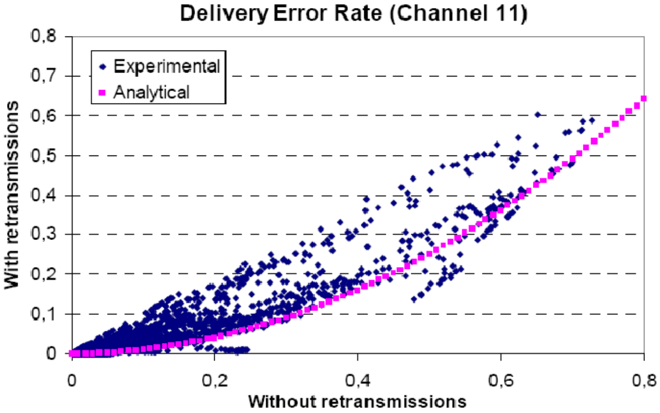


Figure 12. DER with and without retransmissions.

Figure 13 presents the battery voltage as function of time. It drops quickly when the voltage reaches around 2.2 V. The guard times that were used are G_D = 1 ms and G_B = 3.2 ms, thus, according to equation 4, the average current consumption without retransmissions is I_0 = 9.54 mA, so the theoretical lifetime of the station with fully charged 2300 mAh batteries is 241 h. An inspection of the channel 11 chart indicates that the lifetime of the channel 26 station would probably be around 230 h, which is close to the theoretical value, if it was not for the power outage. The lifetime of the station on channel 11 was lower, probably because its batteries were not with full capacity anymore, since the increment in consumption due to retransmissions (equation 5) is small. Even in the worst case (DER_0 = 1) it would be only 8 %. The average current consumption of stations could be reduced to 2.15 mA if the microcontroller was turned off during the sleep state, increasing significantly the lifetime.

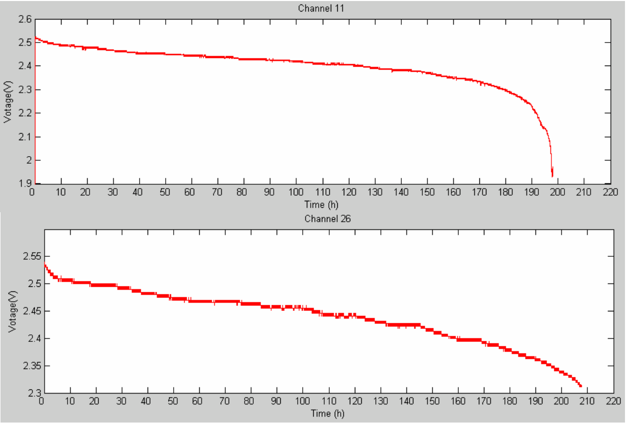


Figure 13. Battery voltage as function of time.

Another set of experiments was made in a hall with one base station and one station. A range (r) of 30 m with moderate channel errors ($DER_0 < 10$ %) was achieved when both the base station and the station were at the chest level. However, when the station was placed at ground level, the errors increased significantly and disconnections were frequent, even for shorter distances. This behavior can be explained by the radio signal power dropping off with r^4 [13] due to ground reflections for short antenna heights.

When the base station was placed over the station, the channel conditions were not affected by the height of the station antenna, even when the station was placed at the ground. This results lead to conclusion that a good place to put the base station is at the ceiling, in the middle of the hall, since it is a central position with relation to the stations and the signal is less affected by the height of the stations antennas.

IV. Second System Prototype

While the architecture of the first prototype comprises a wireless network of suits where the sensor nodes in each suit are connected through a wired BSN to a wireless station that communicates with a base station, the second prototype is based on a fully wireless architecture, i.e., all sensor nodes communicate wirelessly with the base station, as shown in Figure 14.

A. Prototype Overview

The architecture of this second prototype provides more flexibility of use compared to the first prototype, extending the range of applications of the system, since the number of sensor nodes and their distribution through the body can be easily changed according to the needs of the application.

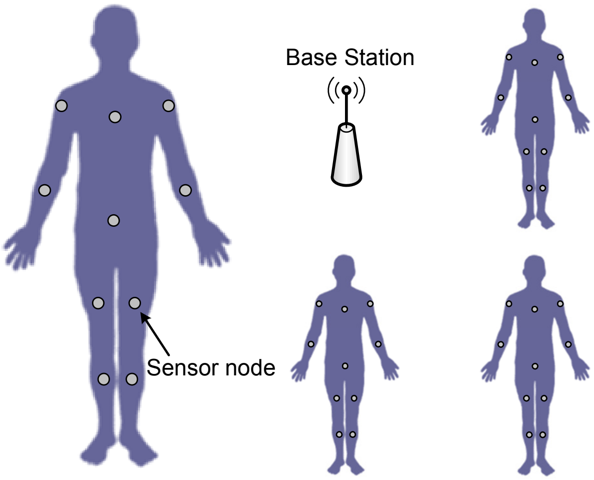


Figure 14. Architecture of the second prototype.

The design of the wireless sensor nodes requires a microcontroller, an analog-to-digital converter (ADC), a transceiver, an antenna and a battery, as well as the sensors. The sensor nodes should be as small and lightweight as possible, since the user may need to use several of them in the body.

The MICAz mote, used in the first prototype, is not adequate in this case, since its size, excluding the battery pack, is 58x32x7 mm. The second prototype is based on the CC2430 [14], a system-on-chip (SoC) from Texas Instruments that integrates a microcontroller, an ADC and a transceiver in a 7x7 mm QLP48 package. Like the MICAz mote, the CC2430 is compliant with the 2.4 GHz version of the IEEE 802.15.4 at the physical layer. Figure 15 presents a CC2430 evaluation module (EM) with external antenna. These modules are part of a development kit [15] used to program and debug the MAC protocol implemented in the second prototype.

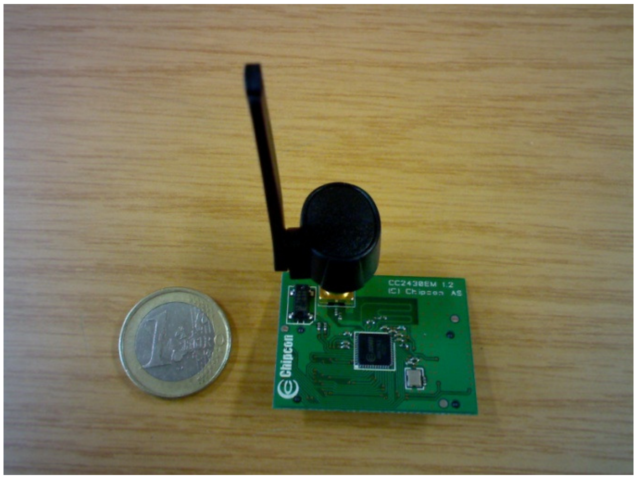


Figure 15. CC2430 evaluation module with external antenna.

The hardware of the prototype of the wireless sensor nodes consists of a PCB integrating a CC2430, an antenna, the sensors and the battery. Instead of using an external antenna, the sensor node will use a small size PCB antenna, described in [16].

B. MAC Protocol

The second prototype implements an improved version of LPRT, called iLPRT [17]. This protocol explores the fact that the traffic of BSN applications is mostly periodic to avoid the transmission of allocation information (RG list) in every superframe (a characteristic of LPRT). Instead of placing the RG list in the beacon, iLPRT includes only the CP size, which is necessary to avoid transmissions during the CFP from nodes that are using CSMA/CA. For transmissions during the NTP, nodes use the same allocation from the previous superframe. For transmissions in the RP, each node computes the period that is reserved for its retransmission with basis on the ACK bitmap information contained in the beacon, using a distributed algorithm that is shared by all nodes.

To ensure fairness in the allocation of retransmission periods when the available RP space is not sufficient to support all retransmissions, the algorithm performs rotation of allocation periods through all network nodes.

Unlike LPRT, if a node does not receive a beacon or a short sequence of beacons, it may continue to transmit new data in the NTP, since the clock drift should allow the node to continue synchronized during a few consecutive beacon intervals. Like LPRT, the node cannot retransmit in the RP when it does not receive the respective beacon, since the ACK bitmap is not available, so it does not know how the RP slots were allocated.

Since the RG list is not included in every beacon, a reconfiguration scheme, such as the algorithm proposed in [17], is required to inform the nodes when a new allocation is made or an existing allocation is released.

The use of shorter beacons tends to decrease the DER and the energy consumption of the sensor nodes. The possibility of transmission during the NTP when the beacon is lost also tends to decrease the DER. This is an advantage also when compared to the GTS mechanism of IEEE 802.15.4, since in this protocol if a device misses the beacon it must not use its allocated slots until it receives a beacon correctly.

C. Simulation Results

Simulation models of the LPRT and iLPRT protocols were implemented using the version 4.0 of OMNeT++, an open source, component-based, modular and open-architecture discrete event simulation environment [18] [19].

Table I presents the parameters related to the posture sensors and their corresponding values. Most motion capture applications operate with a frame rate of 30 fps (frames per second), so the sampling rate of the posture sensors (f_s) was set to 30 Hz. Since each posture sensor module has three accelerometers and three magnetometers, $N_s = 6$.

TABLE I. MOTION CAPTURE PARAMETERS.

Parameter	Name	Value
Number of sensors	N_s	6
Sampling rate	f_s	30 Hz
Sensors sampling resolution	Q_s	12 bits
Battery sampling resolution	Q_b	8 bits

Table II presents the network-related parameters and values. The PHY overhead corresponds to the physical preamble and header of IEEE 802.15.4. The MAC overhead is due to the MAC header and trailer. For data messages, it includes the following fields: FCF (Frame Control Field), data sequence number, PAN address, destination address, source AID, frame type and FCS (Frame Check Sequence).

TABLE II. NETWORK PARAMETERS.

Parameter	Name	Value
PHY data rate	R	250 kbit/s
PHY overhead	OH_{PHY}	6 bytes
MAC overhead	OH_{MAC}	9 bytes
Payload length	L_p	28 bytes
Superframe duration	T_{SF}	100 ms
Mini-slots per superframe	M_{SF}	500

The superframe duration (T_{SF}) was set to 100 ms. With this value, the number of samples (N_{sa}) of each sensor in the posture sensor module, using a sampling rate of 30 Hz, is 3, according to the equation:

$$N_{sa} = f_s T_{SF}. \tag{6}$$

Besides of the multiple samples of the posture sensors, each data message carries also a sample of the battery voltage. The length of the payload (L_p) of the data messages is given by the equation:

$$L_p = N_{sa} N_s Q_s + Q_b, \tag{7}$$

where N_s is the number of sensors, Q_s and Q_b are the sampling resolution of the posture sensors and the battery voltage, respectively. To save space, each two 12-bit samples are compressed into three bytes. With the values obtained from Table II, the payload length results in 224 bits, or 28 bytes.

There is a tradeoff between the delay and the energy consumption and network capacity (number of supported nodes). An increase in the superframe duration would tend to increase the network capacity and decrease the energy consumption, since the payload length would increase, reducing the protocol overhead. However, the delay would also tend to increase.

The maximum PPDU (PHY Protocol Data Unit) size of 133 bytes, imposed by the IEEE 802.15.4 specifications, limit the number of RGs in the beacon and, consequently, the number of transmissions in the CFP for LPRT, to a maximum of 55. On the other hand, iLPRT does not suffer this constraint since the RG list is not included in the beacon.

Comparing with the RG field format used in the first prototype, the length of the AID field was increased from 5 to 6 bits, while the length of the ITS field was decreased from 10 to 9 bits, which means that the RG field maintains the overall length of 2 bytes. With 6 bits in the AID, the addressing capability of the network is increased to 64 nodes. With 9 bits in the ITS field, the superframe can have up to 512 mini-slots. The number of mini-slots per superframe (M_{SF}) was set to 500 instead, because with this value the period of each mini-slot, with the superframe duration of 100 ms, is 200 µs, which is an integer multiple of the resolution of the timer of the CC2430 that is being used (4 µs).

The minimum duration of the CP (CP_{min}) was set to 11 ms, which is enough to support the exchange of two full length frames, including the CSMA/CA backoff and the gap between the frames. The CP is used, for example, for the exchange of association request/response messages between the nodes and the base station and for the transmission of sporadic sensor calibration data. The maximum duration of the CFP (CFP_{max}) is the superframe duration (T_{SF}) minus a reserved period (15.26 ms) composed by the CP_{min} plus the time required to transmit a maximum length beacon (4.26 ms). Therefore, given the used superframe duration, CFP_{max} = 84.74 ms, which corresponds to 423 mini-slots. According to the values defined on Table II, the length of the PPDU (Physical layer Protocol Data Unit) is 43 bytes. That means that the transmission time is 1.376 ms, requiring 7 mini-slots. One additional mini-slot was used as guard time between the transmissions of the sensor nodes, resulting in 8 reserved mini-slots for each data message. Consequently, the maximum number of transmissions that can be carried out in the CFP is 52, either with LPRT or iLPRT.

A brief analysis of the IEEE 802.15.4 GTS mechanism shows that the limited options for the superframe duration combined with the low granularity of the slots result in poor bandwidth efficiency. The IEEE 802.15.4 standard provides only 15 choices for the superframe duration, ranging in a geometric progression from 15.36 ms to 251.658 s. The closest options to the value defined in Table II (100 ms) are 61.44 ms and 122.88 ms. Using the latter value, the average number of samples per superframe with 30 fps would be 3.69 and the average payload length (applying equation 7) would be 34.21 bytes, so the data frame transmission time would be 1.58 ms. Since each superframe is composed by 16 slots, each slot would have a period of 7.68 ms, which means that 79.5% of the available slot space would be wasted if the GTS mechanism was used.

The next three figures present the simulation results for the delivery error rate (DER) as a function of the number of sensor nodes (considering different channel error models and parameter values). For each figure the following cases were simulated:

- LPRT without retransmissions;
- LPRT with one retransmission attempt;
- iLPRT without retransmissions;
- iLPRT with one retransmission attempt.

Simulation results presented in Figure 16 assume an AWGN/BSC channel identical to the used in the theoretical analysis presented on Section III-C. It is also considered that the channel is symmetrical in terms of bit error rate, which means that the BER used in the transmission of the beacon by the base station is the same BER used in the transmission of the data messages by the sensor nodes.

The simulation results for the LPRT protocol are identical to the ones obtained with the theoretical analysis, using equations 2 and 3, for the cases without retransmissions and with one retransmission attempt, respectively. The consistency of results between these two analysis methods provides validation to the simulation model of the LPRT protocol.

As shown by the figure, the DER with LPRT increases with the number of the nodes due to the increase of the beacon length. This is mainly due to the corresponding increase in the RG list length, which takes two bytes per allocated transmission (Figure 5). The beacon of iLPRT does not carry the RG list and its length only increases by one bit per node due to the ACK bitmap, so the DER with iLPRT is practically independent of the number of nodes in the network.

The DER with retransmissions is much better than the DER without retransmissions, for both protocols, when there is sufficient space available in the superframe for all retransmissions. As the number of nodes approaches the capacity of the network, the number of scheduled retransmissions decreases, due to lack of space in the superframe, increasing the DER. For example, the DER with iLPRT starts to increase with more than 48 nodes. In the limit, when the number of nodes reaches 52 and there is no space for retransmissions anymore, the DER with the retransmission mechanism for both protocols reaches the corresponding value of the DER without retransmissions.

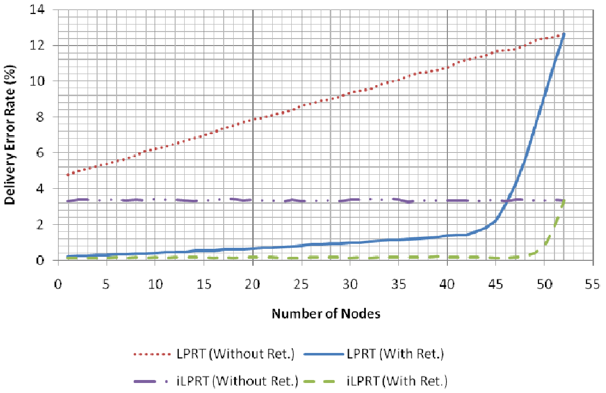


Figure 16. DER using the AWGN/BSC model with BER = 10^{-4}.

As the figure shows, iLPRT improves the DER with respect to LPRT. The main reasons for this improvement are the shorter beacons used by iLPRT, that are less sensitive to channel errors, and the transmission of data by the nodes during the NTP when the beacon is not received, which is allowed with iLPRT but not with LPRT.

A second scenario that takes into account the occurrence of burst errors was considered. The channel errors for this scenario were modeled using the Gilbert-Elliot model [20]. According to this model, the channel alternates between a good state with low bit error rate (BER_{good}) and a bad state, with very high bit error rate (BER_{bad}), with mean dwelling time T_{good} for the good state and T_{bad} for the bad state. The choice of values for these parameters, presented in Table III, is intended to model fast fading, which typically occurs on timescales of milliseconds to tens of milliseconds [21] and can also represent interference from other sources such as IEEE 802.11 transmissions in the same band. The channel errors for the different nodes were made independent, which means that at any moment the channel for some nodes can be in the bad state while for others it can be in the good state. The channel state is the same regardless of the direction of the transmission (from the node to the base station or vice-versa).

TABLE III. PARAMETERS OF THE GILBERT-ELLIOT MODEL.

Parameter	Value
BER_{bad} (nodes to base station)	10^{-2}
BER_{bad} (base station to nodes)	$10^{-2}/10^{-3}$
BER_{good}	0
T_{bad}	10 ms
T_{good}	90 ms

Figure 17 presents the DER curves obtained using the same value of BER_{bad} (10^{-2}) in both directions. The average BER, using the parameters of Table III is 10^{-3}, which is significantly higher than the constant BER used in the previous scenario.

With this value of BER_{bad}, the probability that a beacon or a data message is corrupted by errors when the channel is in the bad state is near 100%, which means that almost all successful transmissions occur only when the channel is in the good state. Since in this scenario all beacons tend to be corrupted when the channel is in the bad state, regardless of their length, the DER using LPRT without retransmissions shows little sensitivity to the number of nodes in the network, unlike what occurs in the previous scenario (Figure 16).

Like in the previous scenario, iLPRT presents an improvement in the DER with respect to LPRT; however, the DER observed in the figure, for both protocols, is worse with small rather than medium number of nodes in the network. The explanation for this behavior relates to the burst errors channel model and the way the transmissions are allocated in the superframe. The scheduling of mini-slots for transmission of data messages starts from the end of the superframe and goes towards to its beginning, as the number of nodes increases, which means that the transmission of the first node is closely followed by the beacon of the next superframe. In a scenario with burst errors, the probability that the bad state observed during the transmission of this data message extends into the reception of the beacon is high. When the beacon is corrupted, the node is unable to make the retransmissions, since it does not have the feedback about the delivery of the data message neither the information about the allocation of mini-slots for retransmission. Therefore, burst errors tend to reduce the effectiveness of the retransmission mechanism for data messages that are scheduled for transmission close to the next beacon. As the number of nodes in the network increases, the average distance between the data transmissions and the following beacon also increases, decreasing the probability that the reception of the beacon is compromised by the bad state observed during the transmission of the data message.

The behavior observed in Figure 17 provides an explanation for the experimental results of DER with retransmissions using LPRT that was presented in Figure 12 (in a scenario with a single sensor node and equal output power in both directions). The experimental results are worse than expected by the theoretical results provided by equation 3, which consider a channel with constant BER, while the simulation results obtained in Figure 17 are closer. This suggests that, in similar circumstances, channel models that take into account burst errors tend to be more realistic than models that assume an AWGN/BSC channel.

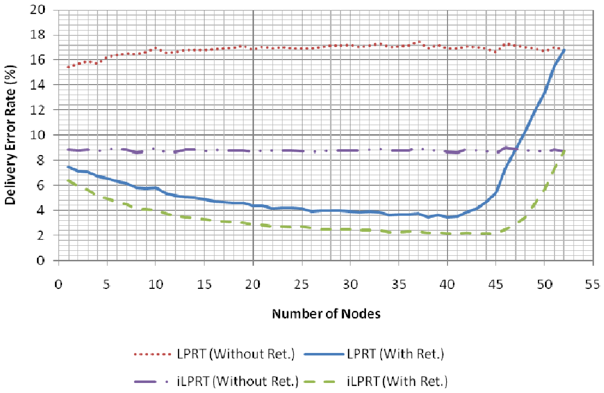


Figure 17. DER using the Gilbert-Elliot model with $BER_{bad} = 10^{-2}$.

The BER at the receiver depends, among other factors, of the modulation used and the SNR (signal-to-noise ratio) [22]. Since the SNR depends of the output power, a solution to improve the robustness of these protocols consists in increasing the output power of the base station transmitter. This increase is not problematic in terms of energy consumption since, unlike the sensor nodes, the base station is not energy constrained.

Figure 18 presents the DER curves obtained in a scenario similar to the previous one. The only difference is that the BER_{bad} for the transmissions of the beacon from the base station to the nodes is reduced to 10^{-3}, to account for an increase in the base station output power, while the BER_{bad} for the transmissions from the node to the base station maintains the value of 10^{-2}. As Figure 18 shows, the increment in the probability of reception of the beacon when the channel is in the bad state has a overall positive effect on the DER with retransmissions for both protocols, but especially when the number of nodes is small, correcting the problem detected in the previous scenario.

Since the BER_{bad} for the beacon in this scenario is lower than in the previous one, the probability of corruption of the beacon becomes significantly dependent of the beacon length again and, consequently, the DER using LPRT without retransmissions shows a dependence on the number of nodes in the network. On the other hand, the DER using iLPRT without retransmissions in this scenario remains the same when compared to the previous scenario, even though the probability of corruption of the beacon is different, since the transmission of data messages during the NTP with iLPRT does not depend of the successful reception of the beacon.

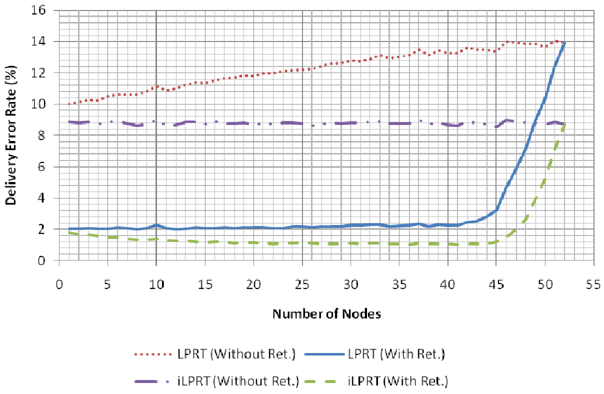


Figure 18. DER using the Gilbert-Elliot model with BER_{bad}= $10^{-2}/10^{-3}$.

Figure 19 shows the average current consumption (per node) of the CC2430 circuitry with the retransmission mechanism enabled, considering the current consumption of 26.7 mA (RX mode) and 26.9 mA (TX mode, 0 dBm) in the active state and 0.5 µA (power mode 2) in the sleep state. The consumption using LPRT increases significantly with the number of nodes, mainly because of the increment in the size of the beacon due to the RG fields (the consumption starts to decrease with more than 45 nodes because the nodes start to drop the retransmissions due to lack of space in the superframe). The consumption of iLPRT increases much less, because the increment on the size of the beacon in this case is only due to ACK bitmap field. The increase in the current consumption due to retransmissions is small, being largely compensated by the benefit of reduction of the DER.

The average current consumption of the CC2430 for the operation of the iLPRT protocol with the retransmission mechanism enabled, considering a scenario of utilization with 45 sensor nodes, is 0.6 mA. The ADC sampling time for each sensor is 132 µs, which means that the total sampling time for the six sensors in the posture sensor module is 792 µs. With a sampling rate of 30 Hz, that results in a duty-cycle for the ADC of 2.38%. Since the consumption of the microcontroller during the ADC operation is 12.3 mA, the average current consumption due to the ADC using this duty-cycle is 0.29 mA.

Most of the consumption of the sensor node is due to the posture sensors circuitry, which consume around 11.5 mA. The total current consumption including the operation of the iLPRT protocol and the ADC is therefore 12.39 mA. Considering the use of a rechargeable Lithium-Ion battery with capacity of 300 mAh, the lifetime of each posture sensor module will be in the order of 24.2 h of continuous operation. The lifetime of the nodes could be significantly increased by switching off part of the posture sensors circuitry between samples.

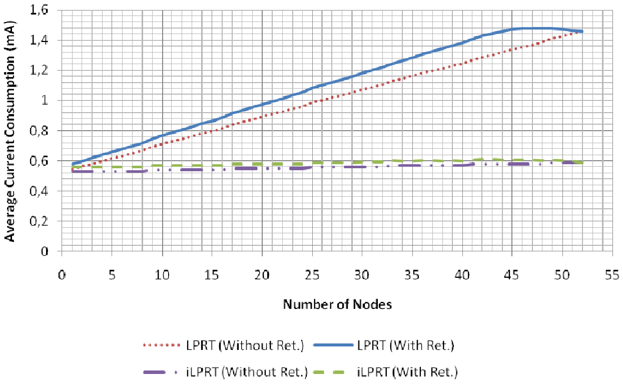


Figure 19. Current consumption with BER_{bad}= $10^{-2}/10^{-3}$.

V. CONCLUSIONS AND FUTURE WORK

This paper describes the design and development of two multi-user wireless BSN prototypes where the posture and movements of multiple users can be monitored simultaneously in real-time using a single base station. The first prototype is implemented using MICAz motes and the second one is based on the CC2430 integrated circuit.

The developed prototypes use the physical layer of the IEEE 802.15.4 protocol, but replace the MAC layer with LPRT/iLPRT. The main advantages of these protocols over the IEEE 802.15.4 CSMA/CA protocol are the collision-free operation and the immunity to the hidden node problem. Advantages of LPRT/iLPRT over the GTS mechanism are: support for more than seven nodes; higher granularity in the allocation of transmission periods; piggybacking of the ACK feedback in beacon frames and automatic scheduling of collision-free retransmissions. These advantages tend to increase the bandwidth efficiency. Another advantage of iLPRT over the GTS mechanism is that the nodes can make the normal data transmission in the superframe even if the beacon is not received, which tends to decrease the delivery error rate.

Future work includes the porting of the base station code of the second prototype from the CC2430 to the CC2530 [23] integrated circuit, which can supply more 4.5 dB of output power than the CC2430, providing a simple alternative to increase the probability of reception of the beacon by the sensor nodes and, consequently, enhance the performance of the system. After that, a set of laboratory and field tests will be performed to evaluate the performance of the system in different scenarios.

ACKNOWLEDGEMENT

This work is supported by Portuguese Foundation for Science and Technology (FCT) project PTDC/EEA-TEL/68625/2006.

REFERENCES

[1] J. A. Afonso, H. R. Silva, P. M. Oliveira, J. H. Correia, and L.A. Rocha, "Design and Implementation of a Real-Time Wireless Sensor Network", *International Conference on Sensor Technologies and Applications (SENSORCOMM 2007)*, Valencia, Spain, October 2007.

[2] B. Lo and G. Z. Yang, "Key Technical Challenges and Current Implementations of Body Sensor Networks", *2nd International Workshop on Body Sensor Networks (BSN 2005)*, London, UK, April 2005.

[3] M. Paksuniemi, H. Sorvoja, E. Alasaarela, and R. Myllylä, "Wireless Sensor and Data Transmission Needs and Technologies for Patient Monitoring in the Operating Room and Intensive Care Unit", *27th Annual International Conference of the Engineering in Medicine and Biology Society (IEEE EMBC 2005)*, Shanghai, China, September 2005.

[4] IEEE Std 802.11-2007, IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June 2007.

[5] Bluetooth SIG, "Specification of the Bluetooth system", November 2003. Retrieved from http://www.bluetooth.org.

[6] IEEE Std 802.15.4-2006, IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), September 2006.

[7] ZigBee Standards Organization, "ZigBee Specification", 2006. Retrieved from http://www.zigbee.org.

[8] D. Marculescu, R. Marculescu, S. Park, and S. Jayaraman, "Ready to Ware", *IEEE Spectrum*, Vol. 40, Iss. 10, pp. 29-32, October 2003.

[9] M. Catrysse et al., "Towards the Integration of Textile Sensors in a Wireless Monitoring Suit", *Sensors and Actuations A: Physical*, Vol. 114, Iss: 2-3, pp. 302-311, September 2004.

[10] J. A. Afonso, J. H. Correia, H. R. Silva, and L. A. Rocha, "Body Kinetics Monitoring System", International Patent WO/2008/018810A2 , February 2008.

[11] Crossbow Technology Inc, "MPR/MIB User's Manual", April 2005. Retrieved from http://www.xbow.com.

[12] D. Gay et al., "The nesC Language: A Holistic Approach to Networked Embedded Systems", *ACM SIGPLAN 2003 conference on Programming language design and implementation*, pp. 1-11, San Diego, CA, June 2003.

[13] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the World with Wireless Sensor Networks", *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, Salt Lake City, UT, May 2001.

[14] Texas Instruments, "CC2430 Data Sheet (rev. 2.1)", 2007. Retrieved from http://www.ti.com.

[15] Texas Instruments, "CC2431DK Development Kit User Manual Rev. 1.5", 2007. Retrieved from http://www.ti.com.

[16] Audun Andersen, "Small Size 2.4 GHz PCB antenna", Application Note AN043, 2008. Retrieved from http://www.ti.com.

[17] O. Gama, J. A. Afonso, P. Carvalho, and P. M. Mendes, "An Improved MAC Protocol with Reconfiguration Scheme for Wireless e-Health Systems Requiring Quality of Service", *Wireless VITAE 2009*, Aalborg, Denmark, May 2009.

[18] A. Varga, "The OMNeT++ Discrete Event Simulation System" *European Simulation Multiconference (ESM 2001)*, Prague, Czech Republic, June 2001.

[19] OMNeT++ Community Site. http://www.omnetpp.org.

[20] J. Ebert and A. Willig, "A Gilbert-Elliot Bit Error Model and the Efficient Use in Packet Level Simulation", *TKN Technical Report TKN-99-002*, March 1999.

[21] A. Willig, "Recent and Emerging Topics in Wireless Industrial Communications: A Selection*", IEEE Transactions in Industrial Informatics*, Vol. 4, No. 2, May 2008.

[22] M. Zuniga and B. Krishnamachari , "Analyzing the Transitional Region in Low Power Wireless Links", *1st IEEE Annual Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004)*, pp. 517-526, Santa Clara, CA, October 2004.

[23] Texas Instruments, "CC2530 Data Sheet", April 2009. Retrieved from http://www.ti.com.

Human-Assisted Calibration of an Angulation based Location Indoor System with Preselection of Measurements

Jürgen Kemper, Nicolaj Kirchhof,
Markus Walter
Robotics Research Institute
Technische Universität Dortmund
Otto-Hahn-Str. 8
44227 Dortmund, Germany
Email: {juergen.kemper, nicolaj.kirchhof,
markus2.walter}@tu-dortmund.de

Holger Linde
Ambiplex GmbH & Co. KG
Idastr. 11
44388 Dortmund, Germany
Email: holger.linde@ambiplex.com

***Abstract*—Besides aspects like accuracy and cost, a simple setup process is essential for the adoption of location systems. However, existing systems often require a time-consuming calibration process to determine sensor node positions or orientations. In this paper we present a software aided approach for the calibration of an angulation-based indoor location system, which only demands from the user to walk through the room. The novelty of the presented approach is that a completely passive sensor technology is applied whereas the calculations are exclusively based on angular measurements. The node localization is realized without any prior knowledge of sensor positions, orientations or the location of the moving person. Nevertheless, it can be shown that an accurate localization of the sensor nodes is possible. The presented algorithm is based on the Newton-Raphson method for solving non-linear equation systems. In order to improve the calibration results, a preselection of the calibration measurements is processed that realizes the identification and non-consideration of unreliable measurements. The accuracy of the approach, its convergence probability and its runtime are evaluated by several simulations. Furthermore, real-world tests, using a location system that exploits the thermal radiation of humans for localization, are carried out to verify the simulation results.**

***Keywords*-Human Assisted, Calibration, Localization, Infrared, Thermal, Thermopile, Tracking, Auto-Calibration**

I. Introduction

The ability to locate people is an essential prerequisite to enable *location-based services*. Therefore, in recent years, several indoor location systems for the field of ubiquitous computing have been developed [2]–[4]. Dependent on the used technology like infrared [5], [6], radio [7]–[9] or ultrasound [10], [11] these systems differ in aspects like accuracy, scalability and cost, whereas especially the latter have direct impact on the acceptance of such systems.

Indeed, the costs of a system do not only depend on the required hardware but also on the installation effort, which corresponds to the time for setting up a system. This time is influenced by the number of sensors, which have to be installed and the time, needed for system calibration, that is, to determine the sensor nodes positions and where needed, their orientation. The number of required nodes depends on their field of view (FoV) and their range. For a cost-efficient localization solution as few sensor nodes as possible should be used.

The calibration effort, on the other hand, is strongly related to the accuracy of the systems. For example, the accuracy of proximity-based location systems such as Active Badge [5] is often limited to room size. Here, the sensor node position is not crucial and therefore, deployment efforts are low.

Contrary to this, lateration-based systems require exact knowledge about the node positions in order to enable accurate localization. Hence, often a time-consuming manual measurement process is required. When additionally applying angulation, besides the exact position also the orientation of the sensor nodes has to be known. Due to this costly calibration, many of the proposed location systems are not practical and therefore suffer from the lack of acceptance.

Automated node localization can help to reduce the effort of calibration significantly. Instead of measuring the sensor coordinates manually, pairwise node distances or other geometrical relations are ascertained by the sensor nodes themselves. From these measurements the sensor coordinates are subsequently calculated. To enable automated calibration, nodes must be able to detect each other. For instance, in a radio location system, they must be equipped with both receivers *and* transmitters.

In the past, especially in the field of *wireless sensor networks*, an essential contribution to autonomous node localization was made. However, these approaches are mostly intended for sensor networks that consist of many nodes and cover a huge area. Indoor location systems on the other hand are often limited to one or a few rooms. Consequently, the requirements for node localization are quite different indoors. In particular, a high position accuracy of the sensor nodes is demanded. This fact implicates that automated calibration is infeasible in certain approaches. E. g. radio location systems estimating distances based on received signal strength (RSS) measurements usually yield a location error of far above one meter. Due to this low accuracy, calibration must be performed manually.

If a passive positioning technology is used, which implies that the sensor nodes are only equipped with receiving elements, distance measurements between sensor nodes are not feasible at all. A promising approach to overcome this problem is *mobile assisted* calibration [12]. Following this approach, a mobile source moving through the room is used to enable the required measurements. However, in that case, not the pairwise geometrical relations between the nodes are measured but those between the mobile source and the nodes.

In this paper, a mobile assisted calibration approach for an angulation-based location system is presented. This system exploits the thermal radiation of humans for localization. It is entirely passive, which means that the sensor nodes are equipped with infrared *sensors* only. Hence, an assisted node localization is required, in which a human acts as *mobile source* himself. Thus, we speak of human-assisted calibration. Furthermore, as infrared radiation does not penetrate walls, the location system is limited to one room and the number of required sensors is small.

For calibrating the system, the node positions and the node orientations have to be determined. The novelty of the presented approach is that a completely passive sensor technology is applied whereas the calculations are exclusively based on angular measurements. The node localization is realized without any prior knowledge of sensor positions, orientations or the location of the moving person. Nevertheless, it can be shown that an accurate localization of the sensor nodes is possible.

The aim of the proposed approach is to significantly reduce the calibration effort of the system by automating the process of node localization. Without automation the calibration is very costly, especially the determination of the sensor orientation. The described approach, however, enables the user to calibrate the system by walking through the room.

The remainder of the paper is structured as follows. In Sect. II related work is discussed. In Section III the ideas and the algorithm to realize human-assisted calibration are described. The process of calibration and some essential software requirements are presented in Sect. IV. In Sect. V experimental and simulation results are discussed. Finally, in Sect. VI a conclusion is drawn and an outlook on future work is given.

II. Related Work

As already stated, in recent years, several indoor location systems have been developed. However, the aspects of calibrating such systems have rarely been addressed.

General approaches were discussed for large-scale sensor networks. Mostly, they are based on distance measurements between sensor nodes. Unknown node positions are calculated via trilateration based on known node positions and the measured distances between these nodes and the unknown ones. Anchor-based algorithms start with a few known sensor positions to calculate all unknown ones with respect to a global coordinate system. In contrast to that, anchor-free algorithms try to determine the network structure without initial anchor nodes. Thus, the calculated system geometry is ambiguous, therefore translation, rotation and flipping is still possible. Both types of algorithms suffer the problem that for spare distribution of nodes no unique structure can be determined. The calculations themselves can either be done incrementally or concurrently. In the prior case, the algorithm starts with a set of three or four nodes to calculate unknown node positions and obtains a new node position in every iteration step. In the latter case, all node positions are concurrently optimized. The advantage of this approach is that the propagation of measurement errors and the probability of being stuck in a local minimum is lower. Finally, centralized (computation on a workstation) and decentralized (by the nodes) approaches have to be distinguished.

Motivated by the *Smart Dust* project, Doherty et al. [13] developed a centralized, anchor-based approach. It derives unknown node positions from proximity and angular constraints between beacons, given by their connectivity. A global solution is found by a linear-programming algorithm working on a network representation of a set of convex position constraints.

A distributed, anchor-based approach (APS) was described by Niculescu and Nath [14] that applies RSSI (**R**eceived **S**ignal **S**trength **I**ndication) and trilateration. For node localization they discuss different methods. In their *DV-hop method* the number of hops between anchor nodes are measured to calculate an average hop distance. Afterwards, the node positions are calculated by trilateration exploiting the hop and distance information. The resulting accuracy is limited to 45 % of the radio range. In contrast to that, the *DV-distance method* estimates the distances between nodes by RSSI measurements and reduces the error by 50 %. An additional paper by both authors [15] deals with a similar approach but instead of trilateration triangulation is used to determine the network structure. The resulting non-linear problem is reduced to a linear one, again based on trilateration.

A mobile-assisted, anchor-free node localization approach was proposed by Sichitiu and Ramadurai [16]. They used a PDA equipped with GPS and WLAN to measure the distance to different nodes at several positions based on RSSI and applied a method using Baysian inference for information processing, afterwards. Experimental results revealed a node localization error of less than 3 m.

Another anchor-free approach, using mobile robots for node localization was presented by Pathirana et al. [17] By applying RSSI measurements and a Robust Extended Kalman Filter-based state estimator for node localization, they were able to realize an accuracy of approximately 1 m, which demands the availability of precise odometry information of a robot.

A general survey of different approaches for indoor location systems was given by Scott and Hazas [18]. Using the *Active Bat* system, they examined three different data gathering methods: Placing several nodes on a mobile frame, whereas the relative locations are known, placing several nodes on the floor and gathering measurements data while a human moves around. For the calculation of node positions non-linear regression was applied. Experiments showed that the frame-based approach in combination with non-linear regression provides the highest accuracy with a mean error of 3 cm. In contrast to that, the human-assisted method exhibits a mean accuracy of 19 cm.

A distributed, anchor-based indoor localization approach using collaborative multilateration was described by Savvides et al. [19]. They showed that the use of ultrasound in combination with ToA (**T**ime **o**f **A**rrival) and trilateration is a sufficient candidate for fine-grained localization. Starting with several anchor-nodes, the unknown node positions are calculated with an over-constrained set of equations. Experiments showed that node position errors of less than 20 cm with a sufficient number of nodes are possible. However, due to the incremental calculation process, error propagation has to be considered.

A mobile-assisted, anchor-free approach without knowledge about the position of the mobile unit was proposed by Priyantha et al. [12]. The authors used the well known *Cricket* system and a mobile robot equipped with a cricket node for their evaluation. By utilizing the robot, they were able to overcome the problems of line-of-sight obstruction and ambiguity due to sparse node deployment. In order to gather appropriate distance samples to solve the localization problem certain movement constraints were developed. Experimental results showed that the median pairwise distance error is less than 1.5 % of the distance between the nodes.

III. Theory of calibration

Before describing the developed calibration algorithm in detail, first of all, the used location system and the resulting requirements are outlined.

A. Passive Infrared Localization

Passive infrared location systems exploit the thermal radiation emitted by humans to determine their position. As sensing elements so called thermopiles are used, which consist of a series connection of thermocouples. If several of those thermopiles are combined to a line or array sensor, whereas every pixel exhibit a slightly different field of view, it is possible to measure the angle under which an object is seen. As illustrated in Fig. 1, the basic principle for determining this angle of arrival (AoA) is to choose that one, which corresponds to the line-of-sight of the pixel with the highest outcome.

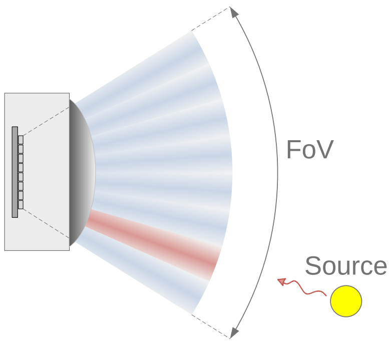


Figure 1. Principle of AoA estimation

The location system described by Kemper et al. [4] consists of special thermopile modules developed by *Ambiplex* [20]. These modules are capable of detecting heat sources within an angular range of 48° and 90°, respectively. In the second case two line sensors with a field of view of 48° are attached to the module (double sensors). At a typical room temperature of 22 °C, the angle between the sensor and a heat source – typically a human being – can be detected up to a distance of 10 m with a angular accuracy of ±2°. For environment temperatures up to 26° the overall system accuracy stays nearly constant, with a mean position error of 15 cm. Higher temperatures than 26° lead to a rapidly increasing position error, because the received radiation is almost indistinguishable from the natural senor noise.

In order to calculate the AOA, a more elaborate algorithm is used that yields a higher resolution by exploiting the fact that the radiating source is typically seen by more than one pixel. Besides the angle of arrival it also calculates a quality measure for the AoA, denoted the *Score*.

In order to set up a location system, several of these modules have to be deployed in a room, e. g. in the corners at chest height. For a room of up to 7 × 7 m, four double sensors with a field of view (FoV) of 90° are sufficient as illustrated in Fig. 2.

B. The node localization problem

After describing the used location system, subsequently we take a closer look on the resulting localization problem:

The one-dimensional angle of arrival provided by the sensors results in a two-dimensional localization problem. A human-assisted node localization is required, as the sensor nodes do not emit signals that can be exploited for mutual detection. In addition, this type of system requires a centralized approach as the sensors' computational power is extremely limited. Moreover, after installing the sensor

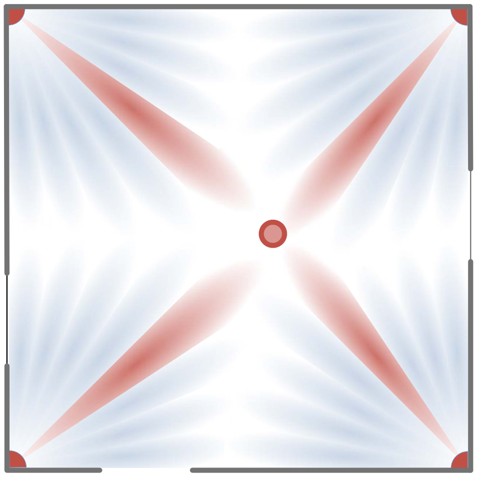


Figure 2. Typical sensor placement

nodes neither their position nor their orientation is known. Consequently, a concurrent, centralized and anchor-free approach seems to be the best choice to realize the calibration. The drawbacks of a centralized approach like the required computational power are negligible since the calibration has to be processed only once. Additionally, in comparison to most of the approaches described in Sect. II, the number of required nodes is small, as thermal radiation does not penetrate walls and the calibration has to be done room by room.

C. Human-Assisted Node Localization

As comfort was the main objective for the development of an automated calibration procedure, the contribution demanded from the user to calibrate the system was supposed to be as small as possible. Therefore, the idea of calibrating a system by walking through the room was considered to be the easiest and most practicable way. The developed algorithm to realize this kind of calibration is described subsequently.

Fundamental Geometrical Relations: Before explaining the used algorithm, first of all, the fundamental geometrical relations of the localization problem have to be described. Figure 3 illustrates these relations and the denotation of variables. Sensors are denoted S_i, where $0 \leq i < N$ and N is the number of sensors. Their positions and orientations with respect to the x-axis are given by $(x_{S_i} | y_{S_i})$ and Θ_{S_i}, respectively. During calibration, the system notifies the user to remain still at a M arbitrary positions P_j, with $0 \leq j < M$. These positions are referred to as *source locations*. Their coordinates are $(x_{P_j} | y_{P_j})$. Finally, the angle under which source location P_j is "seen" by sensor S_i is denoted $\varphi_{S_i P_j}$. The geometrical relation between S_i and P_j can be expressed using the tangent:

$$\tan(\theta_{S_i} + \varphi_{S_i P_j}) = \frac{y_{P_j} - y_{S_i}}{x_{P_j} - x_{S_i}} \tag{1}$$

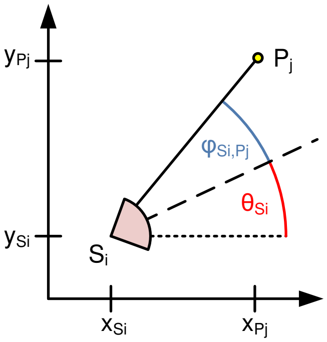


Figure 3. Geometric relations and denotation

After applying some mathematical transformations and the substitution of $\tan(\varphi_{S_i P_j})$ by t_{ij}, a non-linear equation for the calibration algorithm is obtained. This equation exhibits five unknowns x_{P_j}, y_{P_j}, x_{S_i}, y_{S_i} and Θ_{S_i}

$$\begin{aligned} 0 \;=\; & -t_{ij} x_{S_i} + y_{S_i} + (x_{P_j} + y_{P_j} t_{ij}) \cdot \tan\theta_{S_i} \\ & -x_{S_i} \tan\theta_{S_i} - t_{ij} y_{S_i} \tan\theta_{S_i} + x_{P_j} t_{ij} - y_{P_j} \end{aligned} \tag{2}$$

Solvability: If the equations for every sensor-source location combination is set up, a non-linear system with a maximum number of equations $K_E = N \times M$ is obtained. However, in practice, the overall number of equations is lower since, dependent on the sensor setup, not all positions are in the field of view (FoV) of any sensor. Furthermore, for non-linear systems of equations (NLSE) no general prediction of solvability is possible. Nevertheless, for our further considerations, we assume that a solution exists, if the number of independent equations is equal or greater than the number of unknowns. The latter is given by

$$NR_u = 3N + 2M, \tag{3}$$

since every sensor module exhibits three and every source location two unknowns.

To be a bit more problem specific, for every sensor at least three measurements at different positions are necessary (three unknowns per sensor), whereas at every source location the measured object has to be seen by more than two sensors as the source locations are unknown. Table I shows the difference between the number of variables and the unknowns when all positions are in the FoV of every sensor. Unfortunately, this cannot be fulfilled in general. However, the table clarifies that only measurements seen by more than two sensors have a contribution in the reduction of unknowns. Hence, if only these measurements are considered and at least nine valid measurements for every sensor exist, it should be possible to find a solution. In most cases fewer measurement positions are required if at least one of them is in the FoV of more than three sensors. In practice, the number of unknowns is reduced by four since the position and orientation of one sensor should be set as

		N (Sensors)				
		1	2	3	4	5
M (Positions)	4	-7	-6	-7	-4	-3
	5	-8	-6	-4	-2	**0**
	6	-9	-6	-3	**0**	**3**
	7	-10	-6	-2	**2**	**6**
	8	-11	-6	-1	**4**	**9**
	9	-12	-6	**0**	**6**	**12**

Table I
DIFFERENCE BETWEEN NUMBER OF EQUATION AND VARIABLES

reference as well as the distance between this sensor and another one to obtain an unambiguous solution.

If using double sensors instead of single ones as done in the experiments later on, the number of unknowns per module increases to four. Consequently, the number of required source location rises, too.

The Calibration Algorithm: Due to its complexity the resulting NLSE cannot be solved analytically but numerically. The Newton-Raphson method is one of the most efficient approximation procedures for differentiable mappings [21] and can also be applied to multi-dimensional equation systems. The basic idea of this method is to find the root of an equation system of the form $\vec{f}(\vec{x}) = \vec{0}$ by an iterative approximation. Starting with an initial guess, the function is approximated by its derivative at this position. In a second step the root of this derivative is calculated. If the initial guess is near the real solution, the current approximation will typically be better than the former one.

For every step the following LSE has to be solved to obtain the current iteration step $\Delta\vec{x}$:

$$J(\vec{x}_i) \cdot \Delta\vec{x} = -\vec{f}(\vec{x}_i), \tag{4}$$

where $J(\vec{x}_i)$ is the Jacobian matrix and contains the partial derivatives of $\vec{f}(\vec{x}_i)$. Consequently, $\Delta\vec{x}$ can be calculated applying the inverse of $J(\vec{x}_i)$

$$\Delta\vec{x} = -J(\vec{x}_i)^{-1} \cdot \vec{f}(\vec{x}_i) \tag{5}$$

so that finally the assumption for the next iteration can be calculated as

$$\vec{x}_{i+1} = \vec{x}_i + \Delta\vec{x}. \tag{6}$$

In every step the error e of the current approximation is given by the Euclidean norm

$$e = \left\|\vec{f}(\vec{x}_i)\right\| \tag{7}$$

of $\vec{f}(\vec{x}_i)$ at the current position $\vec{x}_i$. Moreover, e can be used as a fitness function. The iteration process can be stopped once the difference of e between two iteration steps falls under a certain limit. However, dependent on the initial guess it may happen that only a local minimum is found or the approximation does not converge at all. In this case, another calculation with a different initial guess has to be started. It is obvious that the initial values of the iteration process are crucial for the convergence of the Newton-Raphson method. However, due to the high-dimensional solution space, finding an adequate initial guess is very complex. Therefore, in the current implementation, we decided to generate it randomly.

Enhancements due to Practical Requirements:: In practice, measurements are not ideal due to sensor noise and systematic errors. Thus, averaging is applied to reduce the impact of noise during calibration, which means that several measurements are carried out while a human is standing still at one position. Averaging over these measurements finally results in a less noisy value.

The impact of faulty measurements, on the other hand, can be reduced by over-determination and calculation of a best fit solution (least squares). That is, the NLSE is composed of more than the required number of equations, for which reason additional measurements and hence additional source locations are required.

In case of having an over-determined NLSE, instead of the inverse of $J(\vec{x})$ the pseudo-inverse

$$J^{+} = (J^{T}J)^{-1}J^{T} \tag{8}$$

has to be applied. Since with every new equation the complexity of the problem is increased, an optimal grade of over-determination has to be found. To check this, we did some evaluations. The results are presented in Sect. V. It should be noted that the complexity of the approach is O(n^3) for every iteration step, where n denotes the number of equations. The computation is dominated by the LU-decomposition for matrix inversion. However, since the calibration has to be processed only once and off-line, the runtime of the algorithm is not crucial as long as it is significantly faster than the manual calibration.

D. Preselection of Measurements

Due to the limited range of the used sensors and the applied algorithm that exploits that the radiation of a human is typically received by more than one pixel, the accuracy of the measured angle decreases with increasing distance between object and sensor.

Additionally, an erroneous AoA is calculated if a human is only partially covered by the FoV of a sensor. This is caused by the fact that the AoA is typically assumed to be in the middle of the covered part. Consequently, this error increases with decreasing distance between radiation source and sensor. Table II illustrates this error for a human located at the edge of the FoV of one sensor (24°) and different distances. It can be seen, that at a distance of 0.5 m the measurement error is almost 13°, which is very problematic for calibration, as high errors may lead to bad calibration results or to a non-solveable NLSE, either.

Distance	3 m	2 m	1 m	0,5 m
AoA	24°	22.9°	18.8°	11.1°

Table II
ERROR DUE TO PARTIAL COVERAGE

In order to avoid the negative impact of measurements that are erroneous due to the described reasons their non-consideration is convenient. For the identification of these measurements some kind of quality measure is required. For this purpose the already mentioned *score* can be used. It is calculated along with the AoA and decreases with increasing distance between sensor and source. However, evaluations have shown that this score is very noisy and highly non-linear with respect to distance as Fig. 4 illustrates.

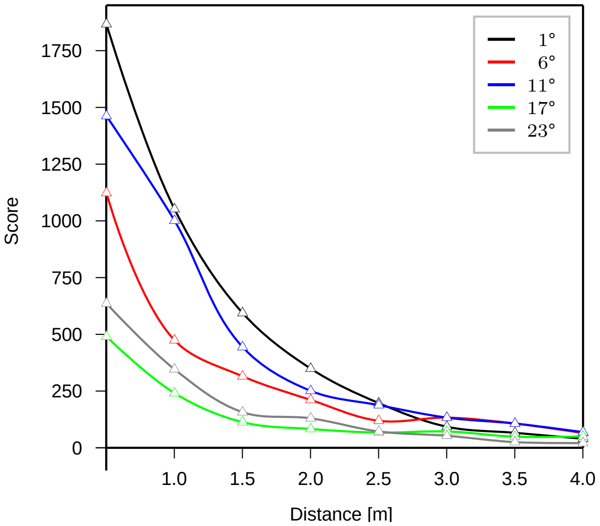


Figure 4. Score values with respect to distance and measured angle

Furthermore, it depends on the object size, its temperature and the AoA itself. Thus it is not very reliable and therefore only used as a threshold criterion. That means that a measurement is valid if its *score* is within a certain interval and rejected if not. In case of the later described real-world tests the limits were determined empirically and set to

$$50 \leq Score_{valid} \leq 1500. \quad (9)$$

IV. Process of Calibration

In addition to an adequate algorithm to locate the sensors, certain requirements must be met in order to enable a comfortable and successful calibration.

A. Software Requirements

As described in Sect. III-C, due to sensor noise, it is necessary that the user, who calibrates the system, stands still at a random position. The position must be chosen in a way that enough information is gathered to calibrate all sensors. Hence, a convenient calibration software has to guide the user with respect to the following aspects. The system should inform the user,

- when to move and when to stand still.
- when enough information is gathered.

In order to meet these requirements, we developed a calibration software that realizes this signalling by sound. After starting the calibration process and entering the room, the software notifies by a deep beep that the user should stop moving. After a short period of time a high beep indicates that this measurement is finished and that the user should continue moving around until another deep beep instructs him to stop. This procedure is repeated until enough information is gathered, which is reported by a high double beep. To realize this kind of calibration two further conditions must be fulfilled. The software must be able to detect, whether the user is standing still or moving, and whether he has moved far enough.

The detection of movement is realized by considering the changes in the measured AoAs. If all of them fall steadily under a certain threshold, it is likely that the user has stopped moving. On the other hand, the decision whether the user has moved far enough to process a new measurement is based on the AoAs. That is, if the change in at least one of the measured AoAs is steadily greater than a certain threshold, e. g. 2°, a new measurement can be processed.

Concurrently, the former described preselection process takes place. So measurements with a score outside the limits are directly rejected. If furthermore less than two measurements remain for one source location, it is rejected completely.

B. Convergence and Local Minima

After gathering the measurement information, the sensor poses have to be calculated. However, as already described, two problems may occur: First, due to the initial guess only a local minimum might be found and second, the calculation could not converge at all. The latter case can easily be detected by the software itself, which stops the iteration after a while, if no convergence is noticeable. The calculation is then restarted with another initial guess.

Due to measurement errors and the use of an overdetermined NLSE the calculated solution is not a *root* anymore but a best fit of the NLSE. Consequently, an indicator is required to identify whether the found solution is only a local or a global minimum. We realized this indication by a plausibility check, as illustrated in Algorithm 1. This check is based on the assumption that a reasonable solution is found when the average of the difference between the measured AoAs and the ones computed with the calculated poses and measurement positions falls under a certain limit. In other words, the solution fits the measured angles. This average is computed separately for every sensor over all measurements. A sensor pose is considered valid if the

Algorithm 1 Plausibility Check

```
Require: $angle_{meas}[1..N, 1..M]$,  $angle_{calc}[1..N, 1..M]$ and
  threshold
  for $i = 1$ to N do
    $avg_i$ = 0;
    for all $k = 1$ to M do
      if $angle_{meas}[i, k]$ is valid then
        $avg_i += |angle_{meas}[i, k] - angle_{calc}[i, k]|$
        $p = p + 1$
      end if
      $k = k + 1$
    end for
    $avg_i$ = $\frac{avg_i}{p}$
    if $avg_i$ > $threshold$ then
      return $false$
    end if
    $i \leftarrow i + 1$
  end for
  return $true$
```

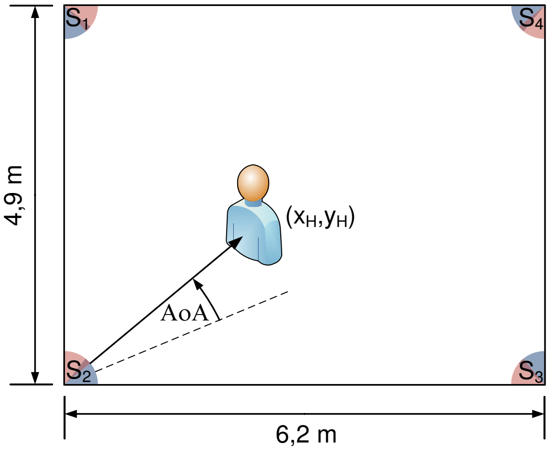


Figure 5. Testbed

computed average is smaller than a given threshold. The overall solution is accepted if all sensor poses are valid.

V. Experimental Results & Simulation

In order to test the developed algorithm, several simulations were conducted. Furthermore, our approach was tested in a real-world scenario.

A. Experimental Results

In order to make some real-world tests, we set up a location system consisting of four double-sensor modules placed at chest height in the corners of a room of $6.2 \times 4.9\,\text{m}^2$, as Fig. 5 depicts.

Since every source location is in the FoV of every double-sensor six different location measurements would have been sufficient (four variables per sensor pair; cf. Sect. III-C). However, due to measurement noise and in order to evaluate the effect of an over-determined calibration, measurements with 15, 18 and 21 different source locations were carried out, each with five passes. Table III illustrates the absolute mean position errors (MPE) in centimeter and the absolute mean orientation errors (MOE) in degree of the different passes. These errors are calculated with respect to the real sensor poses, which were determined by manual measurements. However, it has to be noted that also these measurements are not exact, with regards to orientation. Consequently, the calibration errors are only given relatively with respect to this measurement. Although these calibrations only draw samples, it becomes obvious that due to the limited accuracy of the sensors more than the minimum number of source locations is necessary to obtain a sufficient calibration result. Furthermore, it is shown that redundant source locations improve the mean accuracy, whereas the improvements above 21 source locations are little.

	15 locations		18 locations		21 locations	
	MPF	MOF	MPF	MOF	MPF	MOF
1	24,7	4,0	34,2	2,8	28,6	2,3
2	37,8	3,3	53,6	5,0	28,7	2,9
3	45,3	5,2	31,2	3,5	36,1	2,4
4	46,6	6,3	21,7	2,3	46,4	3,6
5	81,2	8,9	45,6	4,5	24,6	2,8
Ø	47,1	5,5	37,3	3,6	32,9	2,8

Table III
Calibration results with 15, 18 and 21 source locations

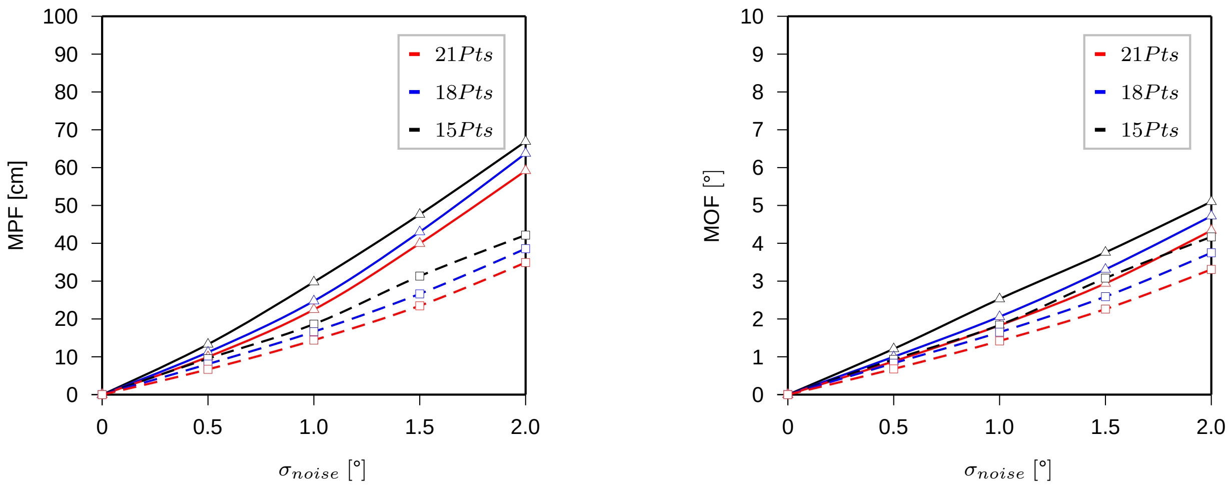
The actual benefit of the automated calibration are the tremendous time savings. In comparison to the manual calibration that required two people and lasted nearly two hours, the automated calibration could be carried out by one person in less than five minutes with 21 different source locations.

Finally, it should be noted that the calibration errors mainly depends on the accuracy of the used sensors and not on the proposed approach itself, as the simulation results show.

B. Simulation Results

As based on the real-world tests no general statement about the convergence, the accuracy and the behaviour under the influence of different noise levels can be given. We carried out several simulations, the results of which are presented in the following. For the simulations we applied the same setup as before. All simulations were done for 15, 18 and 21 different source locations, each with 1000 runs. The standard deviation of the measurement noise was adjusted in steps of 0.5° from 0.0° to 2.0°. According to former experiments, the AoA noise was modelled white and Gaussian. The source locations were generated randomly.

To evaluate the accuracy, only runs with a successful plausibility check were taken into account. In order to



(a) Mean position error

(b) Mean angle error

Figure 6. Calibration error with respect to noise

guarantee high convergence probability for these simulations (in contrast to the following ones), the initial guess was not randomly chosen but set to the exact sensor poses and source locations, whereas the latter were chosen non-deterministic. The resulting position and orientation errors are illustrated by the solid lines in Fig. 6(a) and 6(b). They confirm that an increased number of source locations improves the accuracy. Furthermore, the dependence between measurement noise and the mean position and orientation errors is shown to be almost proportional. As already mentioned, it is also clarified that the calibration accuracy only depends on the measurement error. If there is none, the calibration is exact. At this point, it should be clear that the position errors scale with room size whereas angular errors do not. This can be confirmed by comparing the results given in this paper with the formerly published one [1].

On the other hand, the results represented in dashed lines follow from simulations testing the convergence probability that will be described later on. The interesting aspect with respect to the former described simulations are the lower errors. The reason is easily explained: In the second simulation series random start values were used, which led to a lower overall convergence probability, especially for calibrations with more erroneous measurements. In conclusion, fewer bad results influence the mean values, which leads to lower mean errors.

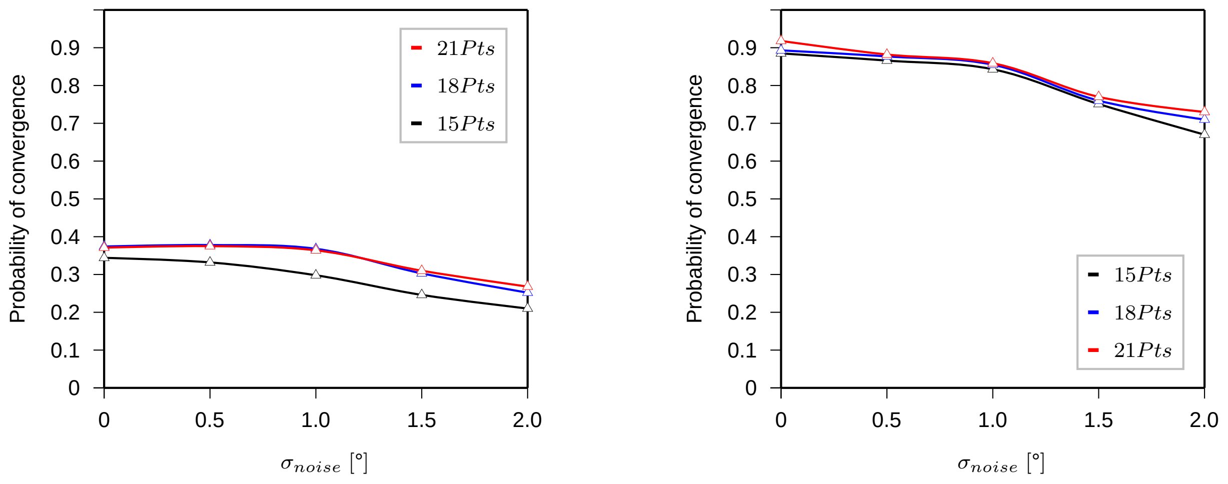
As already mentioned, in the simulations used to determine the convergence probability, the initial guess was chosen randomly. The according outcomes are shown in Fig. 7. Dependent on the measurement noise the convergence probability lies between 0.38 and 0.21 for only one run per measurement set, whereas higher noise worsens the results as well as a higher number of source locations improves them. On the contrary, by increasing the number of tries per set up to ten, the probability rises to values between 0.92 and 0.67.

Finally, Fig. 8 illustrates the average number of runs for successful calibrations with a maximum of ten and the corresponding averaged computing time on a dual core PC with 3 GHz. It can be seen that typically three runs are required for a successful calibration, which takes between 2.5 and 6.6 seconds dependent on the number of source locations.

So finally it can be stated that the simulations confirm the measurement results. When comparing these results with the real world experiments, it can be assumed that the overall measurement errors lie between 1.5° and 2°, which equals the accuracy of the sensors. Additionally, it has to be noted that the convergence probability of the real measurements for 18 and 21 source locations is approximately 0.5 and thus lower than in the simulations. However, that may be founded due to the fact that the sensor model assumed in the simulations was quite simple as e. g. measurement errors due to partially seen objects were not considered.

VI. Conclusion

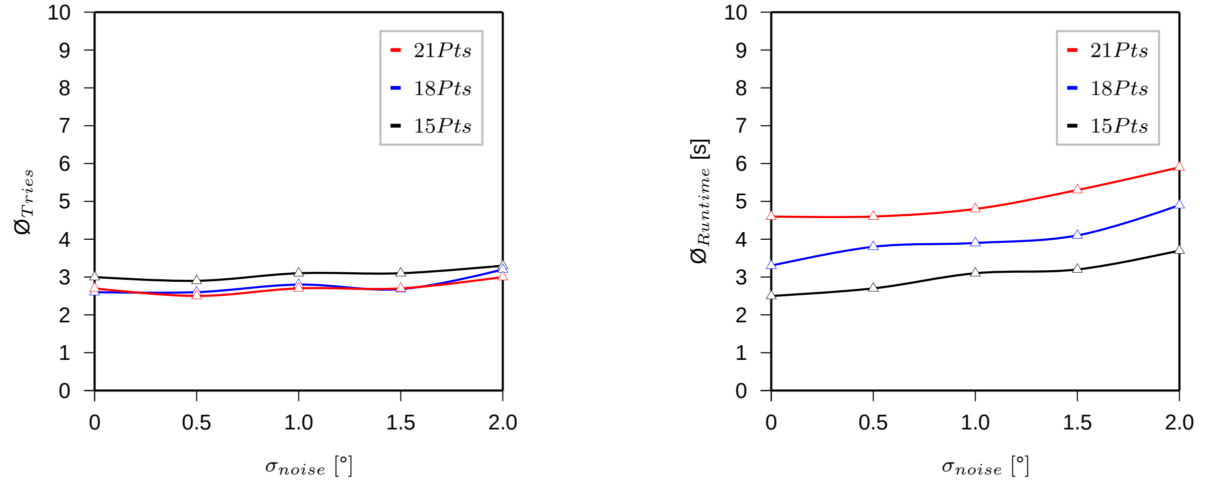
In this paper we presented a software-aided calibration approach for a triangulation-based indoor location system. This approach only requires that the user walks through the room and stops at random locations during calibration. Thereby, the localization of the nodes can be realized without any prior knowledge of sensor positions and orientations and the location of the moving person. Due to the limited computational power of the sensor nodes a centralized approach was chosen whereas the calibration problem is described by a non-linear system of equations. In order to solve this system, an enhanced Newton-Raphson method is applied, whereas unreliable measurements are identified and rejected



(a) One run

(b) Max. of 10 runs

Figure 7. Probability of convergence with respect to the number of locations and measurement noise



(a) Average number of tries

(b) Average runtime

Figure 8. Average number of runs / runtime with respect to noise

based on a certain quality measure during a preselection process before the calculation is started. Real-world tests and simulations show that the algorithm works fine under the influence of noise and that an increased number of source locations improves the node localization accuracy. Furthermore, it could be illustrated that by the developed algorithm and software a calibration can typically be carried out in less than five minutes, whereas this time is dominated by the measurements. In comparison to a calibration based on manual measurement, which takes almost two hours, a significant improvement could be realized.

Future work will concentrate on improving the calibration results by developing movement strategies. Furthermore, strategies to improve the selection of the initial guess, like genetic algorithms, are examined. Additionally, the calibration software could be enhanced in a way that prior information, such as individual sensor positions can be manually fed into the algorithm in order to achieve better results.

References

[1] J. Kemper, H. Linde, and M. Walter, “Human-Assisted Calibration of an Angulation based Indoor Location System,” in *2nd International Conference on Sensor Technologies and Applications (SENSORCOMM 2008)*, August 2008.

[2] J. Hightower and G. Borriello, “Location Systems for Ubiquitous Computing,” *Computer*, vol. 34, no. 8, pp. 57–66, August 2001.

[3] J. A. Tauber, “Indoor Location Systems for Pervasive Computing,” Massachusetts Institute of Technology, Cambridge, MA, USA, Tech. Rep., 2002.

[4] J. Kemper and H. Linde, "Challenges of Passive Infrared Indoor Localization," in *5th Workshop on Positioning, Navigation and Communication (WPNC)*, March 2008.

[5] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The Active Badge Location System," Olivetti Research Ltd. (ORL), Tech. Rep. 92.1, 1992, last accessed on 2010-06-08. [Online]. Available: http://www.cs.cmu.edu/~biorobotics/papers/sbp_papers/integrated2/want_ir_badge.pdf

[6] N. Yoshiike, K. Morinaka, K. Hashimoto, M. Kawaguri, and S. Tanaka, "360^0 direction type human information sensor," *Sensors and Actuators A: Physical*, vol. 77, pp. 199–208(10), 2 November 1999, last accessed on 2010-06-08. [Online]. Available: http://www.ingentaconnect.com/content/els/09244247/1999/00000077/00000003/art00367

[7] Ubisense, "System Overview," http://www.ubisense.net, May 2007, last accessed on 2010-06-08.

[8] R. J. Fontana, E. Richley, and J. Barney, "Commercialization of an Ultra Wideband Precision Asset Location System," in *IEEE Conference on Ultra Wideband Systems and Technologies*. IEEE, November 2003, pp. 369–373.

[9] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," in *19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 2, Tel Aviv, Israel, March 2000, pp. 775–784, last accessed on 2010-06-08. [Online]. Available: http://research.microsoft.com/en-us/people/padmanab/infocom2000.pdf

[10] A. Ward, A. Jones, and A. Hopper, "A New Location Technique for the Active Office," *IEEE Personal Communications*, vol. 4, no. 5, pp. 42–47, October 1997. [Online]. Available: citeseer.nj.nec.com/ward97new.html

[11] H. Balakrishnan, R. Baliga, D. Curtis, M. Goraczko, A. Miu, N. B. Priyantha, A. Smith, K. Steele, S. Teller, and K. Wang, "Lessons from developing and deploying the cricket indoor location system," MIT Computer Science and Artificial Intelligence Laboratory, Tech. Rep., November 2003.

[12] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Mobile-Assisted Localization in Wireless Sensor Networks," in *IEEE INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, FL, March 2005.

[13] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex Position Estimation in Wireless Sensor Networks." in *IEEE Infocom*, April 2001.

[14] D. Niculescu and B. Nath, "DV Based Positioning in Ad Hoc Networks," *Telecommunication Systems*, vol. 22, no. 1-4, pp. 267–280, 2003.

[15] ——, "Ad hoc Positioning System (APS) using AOA," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 3, 30 March-3 April 2003, pp. 1734–1743vol.3.

[16] M. L. Sichitiu and V. Ramadurai, "Localization of Wireless Sensor Networks with a Mobile Beacon," in *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, 25-27 Oct. 2004, pp. 174–183.

[17] P. N. Pathirana, "Node localization using mobile robots in delay-tolerant sensor networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 285–296, 2005, member-Nirupama Bulusu and Senior Member-Andrey V. Savkin and Member-Sanjay Jha.

[18] J. Scott and M. Hazas, "User-Friendly Surveying Techniques for Location-Aware Systems." in *Fifth International Conference on Ubiquitous Computing (UbiComp), Lecture Notes in Computer Science*, vol. 2864, Seattle, USA, October 2003, pp. 45– 54.

[19] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic Fine-Grained Localization in AD-Hoc Networks of Sensors," in *7th Annual ACM International Converence on Mobile Computing and Networking*, July 2001, pp. 166–179.

[20] A. GmbH, "Ambiplex IR.Loc," http://www.Ambiplex.com, last accessed on 2010-06-08.

[21] A. Hoffmann, B. Marx, and W. Vogt, *Mathematik für Ingenieure 1*, 1st ed. Pearson Studium, 2005.

Deployment of Wireless Sensor Network to Study Oceanography of Coral Reefs

Olga Bondarenko, Michael Kingsford
School of Marine and Tropical Biology,
James Cook University, JCU
Townsville, QLD 4811, Australia
e-mail: olechka216@gmail.com,
michael.kingsford@jcu.edu.au

Stuart Kininmonth
Australian Institute of Marine Science,
AIMS
Townsville, QLD4810, Australia
e-mail: s.kininmonth@aims.gov.au

Abstract — **Great Barrier Reef Australia (GBR) is affected by cold water intrusions originating in the Coral Sea and upwelled on the reef. Therefore biological interest in GBR upwelling has been driven by the view that upwelled waters rich in nutrients boost plankton production and overall productivity of the GBR system. Upwelling can be a high frequency short-duration event and therefore it may be challenging to quantify synchrony between physical and biological change impacting the reef. We deployed a Wireless Sensor Network (WSN) for in situ monitoring of upwelling. Temperature is a good proxy for upwelling however 3D dense spatial data required to correctly describe upwelling and their impact on plankton abundance. The array of underwater sensors was deployed at various depths on the coral reef in Nelly Bay, Magnetic Island, GBR. The temperature data are communicated in real time via the ad hoc network using RF signal to the on-shore base station. This permits us to collect the plankton data in real time synchronized to the temperature changes. To explore the utility of WSN we also deployed dataloggers to collect temperature data from the same location. This paper outlines the methods of the WSN deployment for ecological research. It also describes preliminary results. Our preliminary findings did not produce sufficient evidence for upwelling however we did find that the water temperature can vary by as much as 1 °C even on a small spatial scale due to stratification of the water column. Stratification can influence depth-related abundance of plankton and the supply of food to reef associated organisms however we could not confirm this with statistical confidence due to the limited plankton data collected while water stratification was observed. The use of robust real time WSN to trigger plankton collection at the events of upwelling or stratification would have assisted with this investigation.**

Keywords – oceanography; plankton; wireless sensor network; stratification; tidal upwelling; temperature; coral reef

I. INTRODUCTION

Understanding the relationships between physical and biological oceanography is a challenging task due to the very dynamic nature of the oceans. This calls for the deployment of new methods and technology in oceanographic studies that allows the detection and communication of changes in real time. It was proposed to deploy a Wireless Sensor Network (WSN) to quantify the synchrony between physical and biological changes impacting the reef due to oceanic upwelling [1].

Coral reefs have incredible diversity and density of organisms and could not survive without input of additional nutrients from outside the reef [2]. Upwelling has the potential to facilitate such input and also has a great influence on the supply of planktonic food to reefs [2, 3]. Upwelling can be a high frequency short-duration event and therefore it may be challenging to quantify synchrony between physical and biological change impacting the reef. The objective of our study was to determine the influence that physical oceanography has on distribution of plankton in a coral reef environment.

In Great Barrier Reef (GBR), Australia, cold water intrusions come from the Coral Sea bringing nutrients into GBR waters [4]. Biological interest in GBR upwelling has been driven by the view that upwelled waters are rich in nutrients and contribute significantly to the overall productivity of the GBR system [4]. Moreover, Furnas and Mitchell [4] found strong correlation between temperature and concentrations of phosphate, nitrate and silicate in upwelled regions of GBR. The ocean fluctuations in nutrients result in variations in the growth of marine organisms such as phytoplankton [5-7]. Understanding the plankton abundance and composition is essential to understanding of the GBR food chain.

Plankton is considered to be one of the most important organisms on Earth since it is a primary food producer for most aquatic life. Based on the trophic level, plankton could be divided into three broad groups: phytoplankton (producer), zooplankton (consumer) and bacterioplankton (recycler) [8]. Understanding the phytoplankton productivity in the world ocean has recently become a major concern because of its role in CO2 recycling and therefore the effect on global climate change [9]. In addition to making a significant contribution in removing carbon dioxide from the atmosphere, phytoplankton creates the foundation of the ocean food chain.

The phytoplankton generally increases in biomass at the junction where frontally convergent circulation has either supplied limiting nutrients or resulted in the aggregation of plankton particles [7]. Thomson and Wolanski [10] established that strong tidal currents can pump nutrient-rich water from below the mixing layer through the reef passages onto the shelf. Such inputs of inorganic nutrients are responsible for the large fluctuations in phytoplankton biomass and overall primary production [5]. Phytoplankton blooms, defined as rapid growth in population, take place when upwelled waters bring nitrate, phosphate and silicate nutrients into euphotic zones [5]. Plankton species can move up and down the water column [11], thus plankton abundance should be estimated with respect to depth and appropriate 3D sampling design is important.

The upwelling can be caused by various dynamic processes in the ocean including wind, topography and tidal movements. Large scale coastal upwellings are generally driven by wind force. This type of upwelling occurs when alongshore winds generate Erkman transport causing the surface waters to move offshore and be replaced by deeper nutrient-rich water that upwells close to shore [12]. High frequency coastal upwelling can also be associated with tidal jets, internal tides, internal waves and internal tidal bores [13]. The temporal and spatial variability in upwelling near coral reefs may contribute to temperature variability, the balance between locally and remotely derived nutrients, and the overall dynamics of coral reef system [14].

The GBR upwelling allows the cross-shelf intrusions of Coral Sea water through the reef matrix [15]. Andrews [16] used temperature to trace cross-shelf transport which in open stratified water produces a marked bottom-temperature signal. The temperature was found to mark the upwelling intrusions adequately [16]. Furnas and Mitchell [4] found that nitrate, phosphate and silicate concentrations are strongly correlated with water temperatures.

We hypothesized that changes of sea water temperature impact the abundance of plankton and propose to set up real time monitoring of the effect of high frequency temperature changes on plankton abundance. We also hypothesize that daily tides have similar effect on plankton abundance as previously documented upwelling but with smaller magnitude.

High frequency upwelling can be created by tidal movements. Tidal currents interacting with complex reef topographies are common in coastal environments [17]. Tidal jets, tidal waves and tidal bores have predominantly cross-shore direction [18]. In coral reef environments they facilitate water exchange between the near-shore shelf, the outer shelf, the slope and the open ocean. Pineda [19] found that internal tidal bores produce upwelling by transporting subsurface water onshore and facilitating the transport of larvae. It was recorded that such upwelling was caused by tidal bores and had an effect on surface temperature that lasted days [19]. Along with the water exchange, cross-shore circulations promote exchange of nutrients, pollutants and biological material [18, 20]. The intensity of tidal upwelling and the distance they can travel shoreward is hard to predict because it is influenced by bathymetry, tidal amplitude and the passage shape thus it is unique for each area. There are very few comparable data available on shallow water upwelling close to the shore. In this study we wanted to verify if tidal upwelling in the GBR penetrates as far as 80 km shoreward and thus deliver nutrients to inshore coastal reefs.

Stratification of the water column can also create variation in biomass distribution of plankton [21]. Stratification refers to layers of different physical properties. A density barrier between the layers reduces mixing of the water. However density differences can also promote dynamic processes in the ocean. For example, even small horizontal density differences caused by differences in surface heating can create strong currents. Water density is a function of water temperature and salinity. Density increases with an increase in salinity and a decrease in temperature. Therefore waters of high temperature and low salinity generally stay at, or near, the surface and the waters of low temperature and high salinity are generally located at depth. The salinity barrier is called the halocline and the temperature barrier is called the thermocline. The thermocline and halocline can have a strong influence on the distribution and dispersal of plankton species [21, 22]. Stratification can last for hours, days and in some cases, when the water is calm, it can last for months. Stratification can be destroyed by dynamic processes promoting water mixing such as tides, storms, hurricanes, upwelling, strong winds and currents.

To be able to trace the effect of high frequency temperature changes due to daily tides, stratification events and upwelling with sufficient tolerance we propose to monitor on a relatively small spatial scale compared to previous studies. Thus the aim of this study was to understand the effect of high frequency changes in the sea water temperature due to tidal fluctuations, upwelling and water stratification on plankton distribution and abundance at Nelly Bay, Magnetic Island, Australia.

We deployed a Wireless Sensor Network (WSN) for in situ monitoring of temperature on 3D scale to be able to collect high quality spatial data required to fully understand the impact of temperature on the distribution and composition of plankton species. Data loggers have been deployed by Australian Institute of Marine Science (AIMS) for in-situ monitoring of sea temperature along various reefs of GBR. Data loggers instantaneously record sea temperatures every 30 minutes and are downloaded every 6 to 12 months, depending on the site. The data loggers store the information which can be downloaded at the end of each experiment, generally every few months, thus immediate collection of plankton samples in the event of high temperature variation was not possible. We therefore deployed WSN to allow biological data collection (plankton) at the same time as physical change (temperature) was detected.

Utilization of WSN technology is quite appropriate when dealing with very dynamic organisms such as plankton. One of the main challenges faced in plankton field studies is the fact that plankton communities are very dynamic and under favorable conditions the cells can divide quite rapidly [23]. Large short-term fluctuations in phytoplankton biomass as well as transport of matter and energy through plankton community [23] calls for a special sampling technique where sampling can be performed shortly after the potentially favorable conditions have been detected. We deployed WSN to collect 3D temperature data and communicate information about changes in the water column in real time.

II. METHODS

A. *Sensor Array*

The array of sensors was deployed on a 3D spatial scale with horizontal coordinates spaced out along the reef crest and reef flat and at various depths. Sensor network is a term used to describe the latest trend in electronic monitoring where each sensor contains a small computer able to manage and collect environmental data and transmit in real time [24]. Ambient Systems is a supplier of wireless mesh networking solutions that consist of chips embedded with the Ambient's networking software and radio transceiver technology [25]. In this study we used Ambient Systems smart temperature sensor solution based on 1-Wire devices DS18B20 from Dallas Semiconductor. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1–Wire bus; thus, it is possible to use one microprocessor to control many DS18B20s. The resolution of the temperature sensor is user-configurable to 9, 10, 11, or 12 bits, corresponding to increments of 0.5°C, 0.25°C, 0.125°C, and 0.0625°C, respectively. In this study the sensors programmed to the maximum resolution of +/-0.0625 °C.

Multiple DS18B20 sensors were connected on 1-Wire to a processing unit called Unode supplied by Ambient Systems and positioned inside the surface buoy (Fig. 1). The Unode has integrated RF networking capabilities to communicate with other Unodes and the base station thus allowing us to create a sensor network with real time data transmission capabilities. Each string of temperature sensors was connected to a Unode and positioned underwater inside of an hydraulic cable at various depths (2 meters apart) (Fig. 2). We initially deployed a network consisting of 8 moorings with one Unode each, however due to technical problems the network was downsized to 4 moorings (Fig.3).

In a sensor network, each node is able to manage the collection of environmental data. This management includes interacting with other sensors to determine the data collection rates and electronic system status. The environmental data are then packaged up using standard networking protocols to broadcast into the network. This means that if the node is unable to directly contact the target base station the data can be rerouted to the target via other sensors (ad hoc network establishment). This is very important in choppy sea conditions. There is no hierarchy between the nodes and they can be spaced out randomly to form multi-hop mesh as long as the distances are within the signal reach. The sensors communicate a unique identification number and thus the data can be tagged with three dimensional attributes (x, y and depth). The transmitting frequency band of 900 MHz was selected as the most suitable compromise between baud rate, humid environment and commercial availability of transmitters [1]. Due to corrosion of underwater cables the temperature data from WSN was only received for 16 hours on 21st and 22nd of September thus limited our analysis.

B. *Dataloggers*

To ground-truth the data we planned to receive from WSN we also used TG3100 temperature dataloggers (Gemini Data loggers UK Ltd), that were calibrated to measure temperature with ± 0.2 °C accuracy. These dataloggers are manufactured in water proof packages thus we could place them on the outside of hydraulic cables containing DS18B20 sensors. We employed TG3100 temperature dataloggers to record temperature data over time at two depth levels and different spatial position on reef profile. We placed dataloggers on four moorings out of total eight moorings used by WSN. The inner moorings were positioned on reef flat and the outer moorings on the outer edge of reef slope. Dataloggers were attached to mooring lines at two depth levels, 1m from the sea surface and 1m from the sea floor (Fig. 2). The loggers at two depth levels were expected to detect the presence of stratification or upwelling. In total, eight dataloggers were synchronized and programmed to record temperature every 10 minutes for the period from 05/09/2007 until 25/09/07.

C. *Study site*

The data on temperature and plankton were collected at Nelly Bay (146 51’ 9” E 19 9’ 52”S), Magnetic Island, Australia. Magnetic Island is situated about 7 km off Townsville; it is bordered by a number of sheltered bays with fringing reefs. Magnetic Island is classified as inner-shelf Island and is situated 7 km offshore. Nelly Bay's 1800m-wide sand and rubble intertidal reef crest and slope area was used for data collection (Fig. 3). The temperature collecting nodes were spaced out along Nelly Bay reef crest and reef flat at various depths.

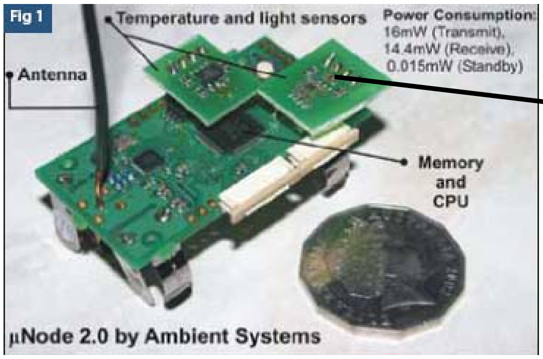


Figure 1(a): The electronic module called Unode supplied by Ambient Systems [25]. The Unode incorporates radio transmitter, receiver and a memory card.

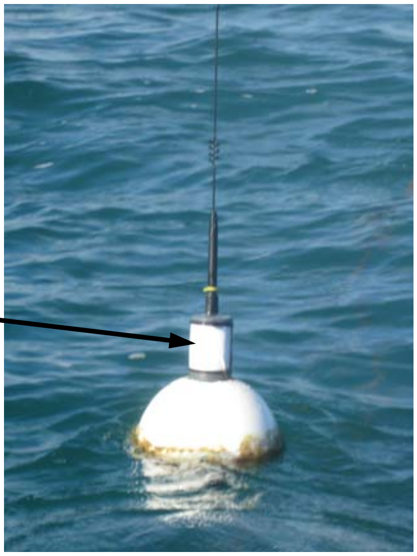


Figure 1(b): Photograph of the surface buoy with Unode placed inside the plastic canister with external antennae. The black arrow points Unode position inside the surface buoy.

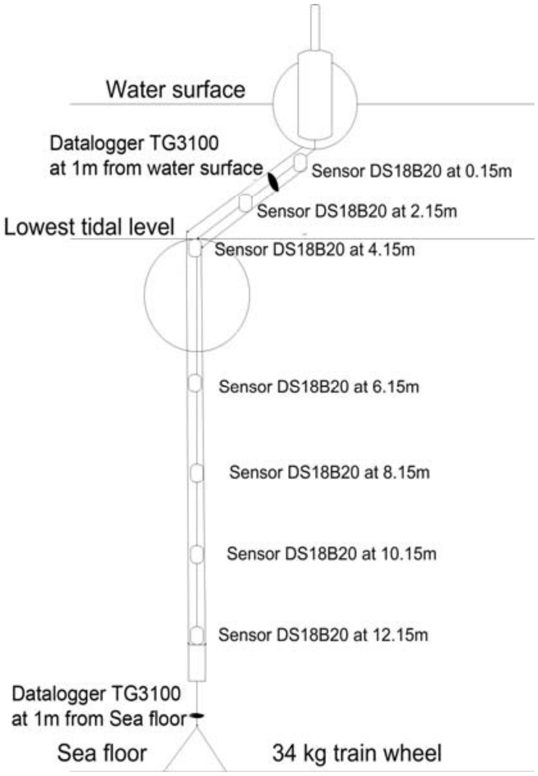


Figure 2 (a): Diagram of the sensor mooring with seven DS18B20 digital thermistors and two data loggers TG3100.

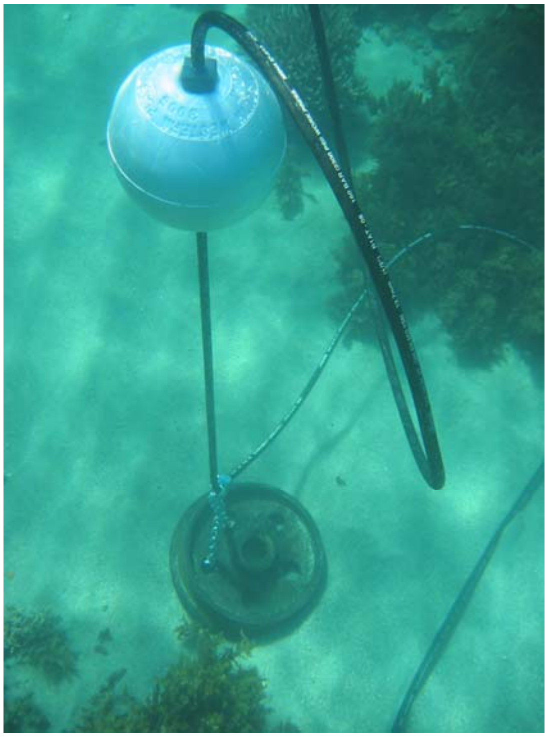


Figure 2 (b): Underwater photograph of the inner mooring with temperature sensors inside the hydraulic cable. Visible in this photograph is the sub-surface float and the wagon wheel anchor. The cable connecting sub-surface float to the surface buoy is floating loose to allow tidal variations.

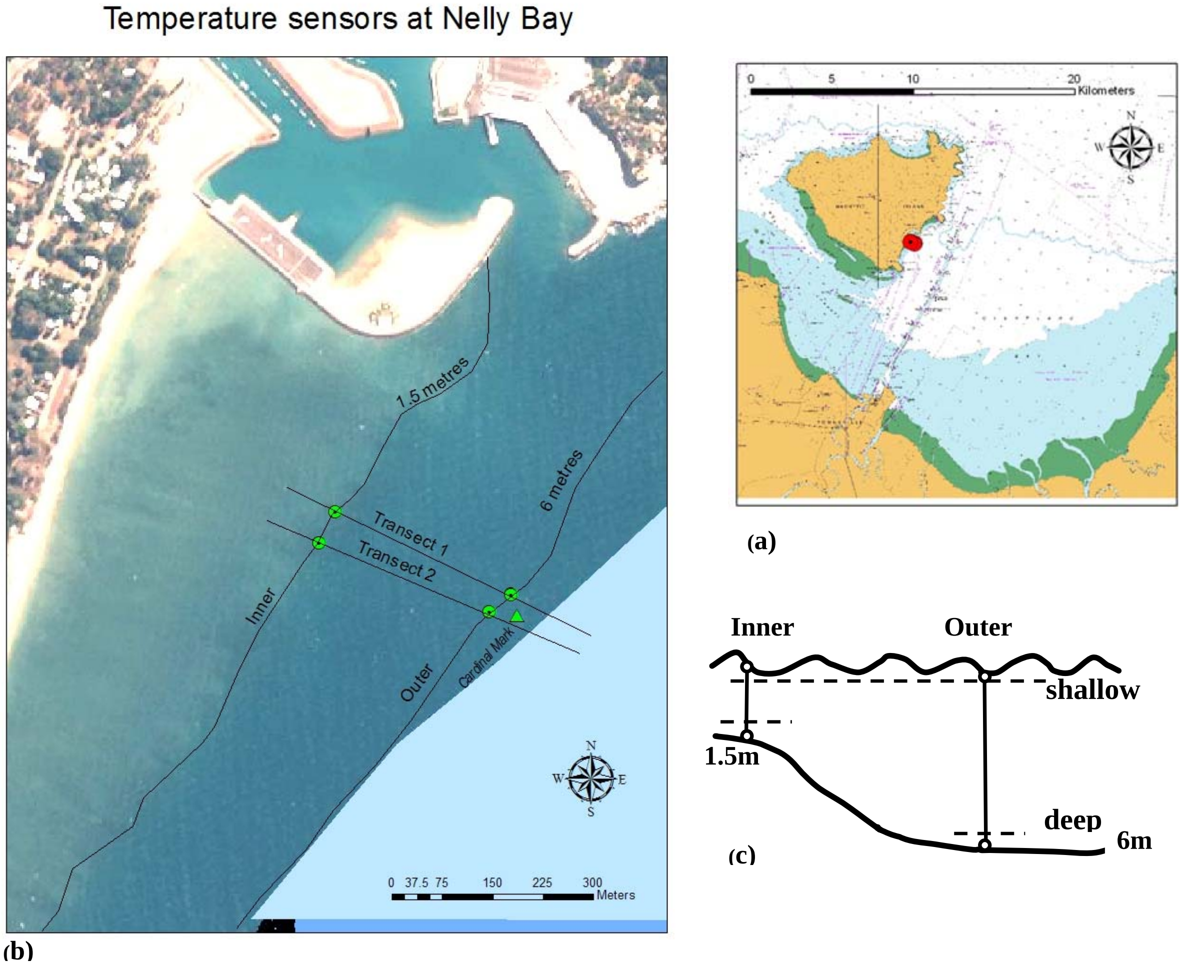


Figure 3: a) Magnetic Island and Nelly Bay b) Study site at Nelly Bay, Magnetic Island, Australia. The sensor network consisted of two transects with two moorings each. Each moorings had seven temperature sensors deployed on vertical scale 2 meters apart (Fig. 2). The inner moorings of each transect were placed on the reef flat at 1.5 meters depth (lowest astrological tide) and the outer moorings were placed on the outer edge of the reef slope at 6 meters depth; c) schematic representation of transects with inner and outer moorings in respect to reef bathymetry on vertical scale, dotted line indicates depth strata (shallow and deep).

D. Temperature stratification and upwelling

We expected to detect both water stratification and upwelling during observation period. The intrusion of cool water to coastal areas often varies with lunar tides [13]. It was predicted therefore that stratification and upwelling would be greatest during spring tides. To separate these events from the temperature offset caused by calibration and resolution of DS18B20 sensors and TG3100 dataloggers a threshold had to be established. When comparing data from 2 dataloggers (shallow and deep) the maximum error rate would be 0.4 °C net based on the accuracy of TG3100. This level of error would be very unusual. A criterion of 0.5°C net vertical gradient from the surface temperature was recommended as a threshold to distinguish weakly stratified regions from unstratified [26]. We therefore adopted this criterion (0.5°C as maximum difference between shallow and deep temperatures, ΔT) for the water column to be considered mixed. Thus stratification or upwelling would be detected if ΔT≥0.5 ºC.

We expected to be able to detect upwelling as cold water intrusions moving vertically up and horizontally towards the shore and finally reaching the surface. It was predicted that the outer mooring sensors would detect temperature drops first and then cold water intrusions would reach inner moorings but with some time delays.

We initially intended to spatially analyze the stratification and upwelling events based on the temperature data from 8 mooring stations. However due to technical problems the data were downloaded from 4 stations only thus limiting our capacity to do spatial analysis with statistical confidence.

E. Plankton data collection

To measure the influence of water temperature on plankton abundance, depth stratified plankton samples were collected next to each temperature station. We expected that depth related patterns would be greatest at spring tides and therefore plankton samples were collected on a high tide during spring tides using Niskin bottles. Niskin bottles are in common use for small size plankton sampling; they are cylinders that can remove columns of water of known diameter and depth [27]. The Niskin bottle method was preferred for this study over collection with plankton net because it minimizes trauma and enhances the survival of phytoplankton taxa that are easily damaged or killed when they come into contact with the mesh of plankton net [28]. This method also allowed us to take samples precisely next to each temperature station which would not be possible with long tows of plankton nets. In this study we used 5-litre Niskin bottles. The water samples were collected next to each temperature station at two depth levels that correspond with the depth of temperature loggers (1 meter from the sea surface and 1m from the sea floor). We collected plankton samples at the time of high tide during spring tides (high tide of ≥ 3.4 meters). In total we collected 40 samples over 5 days from 6th to 11th of September 2007.

The water bottle adequately sampled mesoplankton (0.2 - 2 mm) and microplankton (20-200 µm). A 50 µm mesh sieve was used to concentrate 5-litre field samples into 250 ml jars. All samples were concentrated and preserved in 3% formalin within three hours from collection to avoid predation or decomposition.

F. Laboratory techniques

In the laboratory facilities samples were further filtered using 50 µm mesh sieve to a 20 ml concentrate out of which 1ml subsamples were taken. We used a modified 10 ml calibrated pipette with wide mouth (5mm) to provide wider entrance for the small organisms [28]. The 20 ml concentrate was stirred prior to taking a subsample. Manual mixing was not sufficient to mix up and break up colonial phytoplankton. Phytoplankton species which aggregated in colonies or chains were excluded from counts due to higher effect of patchiness on subsample variance. Thus only unicellular unchained species were counted and analyzed. We used Sedgewick grid which contains 1ml of subsample volume for plankton counting.

III. RESULTS

A. Temperature variations and physical oceanography

The water column was generally homogeneous in September 2007 however a vertical stratification was found on some occasions at both transects (Fig. 4). The average difference between shallow and deep dataloggers was 0.1 ºC which indicated that the entire study area was a mixed layer. Stratification events occurred during spring and neap tides and were most obvious at the outer sites (Fig. 4). In contrast, the waters were well mixed at the inner sites. The maximum temperature difference between surface and near the substratum loggers was 1.2 ºC (1pm on 6th of September, Fig. 4b). A 1 ºC difference on 17th of September ranked second and occurred around 4pm (Fig. 4b). The first stratification event lasted for 2 hours (12pm-2pm) and the second event on 17th of September lasted for 8 hours (11am-7pm) calculation based on critical criterion of $\Delta T \geq 0.5$ ºC.

The first stratification event occurred during spring tides (maximum amplitude of 2.7 meters) and the second event occurred during extreme neap tides (amplitude of 0.8 meters, Fig. 4). Tide amplitude therefore was not the main factor driving temperature stratification as originally predicted. This was confirmed by regression analysis that showed no relationship between the tide amplitude and stratification level represented as the difference between temperatures of shallow and deep layers of the water column ($r^2 = 0.0035$; ANOVA F=0.267; df=1,78; ns).

The data also showed that extreme spring tides (amplitude of around 3 meters) did not promote mixing of the water column more or less than tides of lower amplitude. Greatest stratification did not occur at any particular phase of the tide. On some occasions stratification was greatest at low tide and others at high tide but only during neap tides (Fig. 4). Peak periods of stratification were during tides of low amplitude and near low water at other times (Fig. 4).

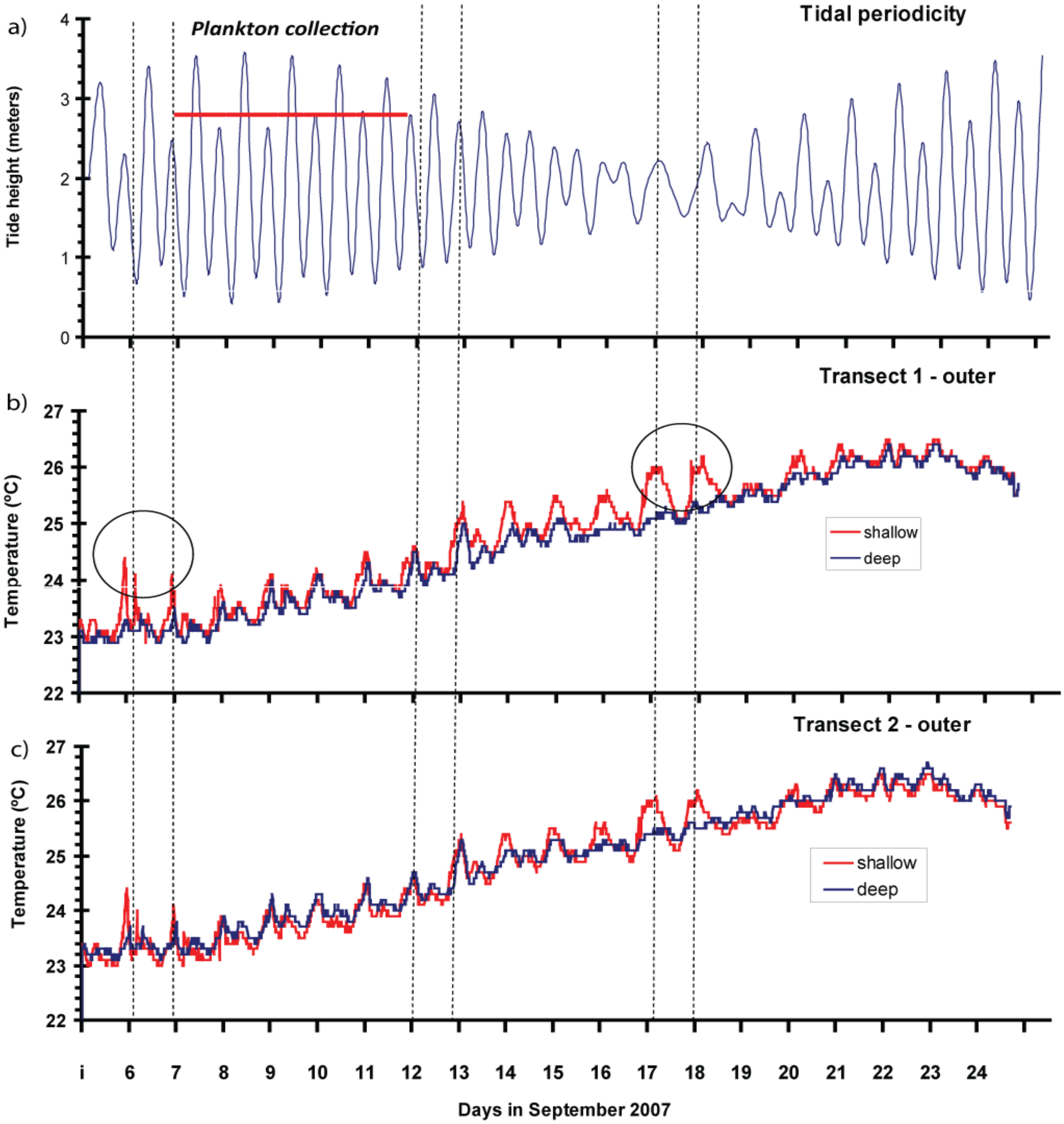


Figure 4: a) Tidal variation during 20 day period in September 2007, the red line highlights the spring tides and when plankton was collected. b) and c) Temperatures at shallow (1m from the surface) and deep (1m above the sea floor) dataloggers from transect 1(b) and transect 2(c) recorded at the outer moorings (Fig. 3). Circles in figure (b) highlight the events of water stratification. The dotted lines show the tidal phases during daily temperature peaks and falls on 6, 12 and 17 of September.

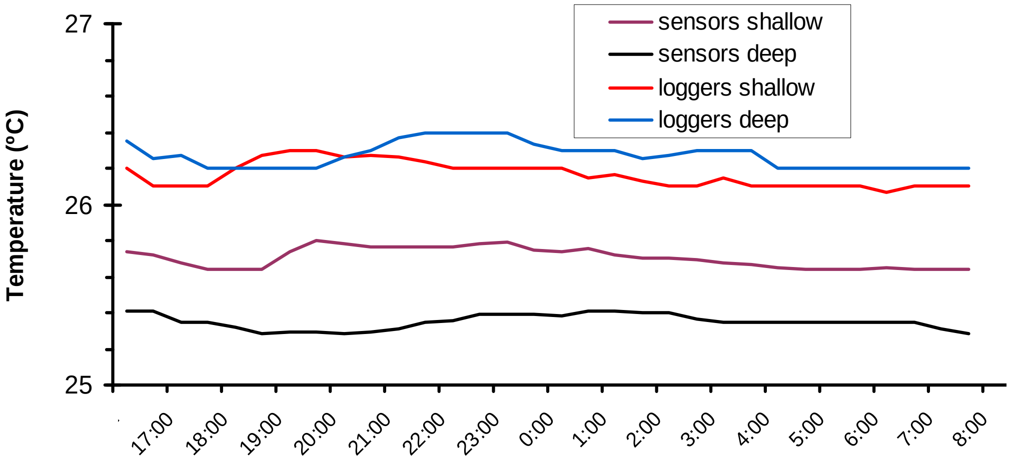


Figure 5: Temperature data collected by the data loggers (TG3100) and real time sensors (DS18B20) at the outer mooring station of transect 2 on 21st and 22nd of September 2007. The data loggers and real time sensors were fixed on the same mooring line at two depth levels: shallow (within 1 meter from the surface) and deep (1 meter from the sea floor).

No upwelling was detected during the observation period. Despite thermal stratification the cold water did not persist at the surface. The daily temperature drops of 0.8 ºC on average did not always correlate with phase of the tide (Fig. 4). Moreover, high temperature variations between daily maximum and minimum occurred during spring tides (maximum difference of 1.3 ºC, Fig. 4) and during neap tides (maximum difference of 1.1 ºC, Fig. 4). Tidal amplitudes did not correlate to daily temperature variations ($r^2 = 0.0049$; df=1, 20; ns). Thus bigger tides did not produce greater differences between daily temperature maximum (peaks) and daily temperature minimum (drops).

B. *Real time sensor array*

The utility of a real time sensor array based on WSN technology was tested and ground- truthed with dataloggers. We compared the temperature data measured by dataloggers and real time sensors on the same spatial and temporal scale (Fig. 5). Both datasets showed the same trends, however there was a temperature offset between real time sensors and dataloggers of approximately 0.5 ºC (Fig. 5). This was due to the accuracy of temperature data received from DS18B20 sensors.

C. *Plankton abundance*

The relationships between plankton abundance and depth varied daily for most taxa. When differences were found it was generally at day one of the plankton collection (6th of Sept) when the water column was stratified by 1.2 ºC difference between shallow and deep loggers. Differences in abundance between depths were found at day one for nauplius larvae, diatom *Coscinodiscus* spp. and the dinoflagellate *Ceratium* spp. Abundance of nauplius was high in shallow waters during day one and changes in rank abundance among days resulted in a significant interaction between day and depth. Abundance of copepods varied among days with highest numbers of copepods collected at the last two days of collection period. Although there was a trend for difference of copepod abundance between depths on day two and five this was not detected by ANOVA. Main effects and interactions were not detected for other taxa probably due to high residual variants.

IV. DISCUSSION

A. *Temperature stratification*

Although we did not find sufficient evidence for upwelling this study confirmed initial prediction that the water temperature can vary significantly even on a very small spatial scale; thus adequate spatial resolution is required when collecting temperature data to analyse oceanographic events on coral reefs.

This study challenges the existing view that the inner shelf waters of the GBR Lagoon are generally unstratified [16, 29-31]. Wolanski et al. [30] measured the temperature across the GBR Lagoon offshore from Townsville and found that there were no vertical temperature gradients for inshore waters and some were found offshore during calm weather conditions. Wolanski et. al's [30] measurements were taken weekly from Jan 1979 to Jan 1980 over a large spatial scale. Similar observations were documented by Orr [32] in September-October 1933 in the GBR Lagoon close to Low Isles (station at 16.35ºS, 145.6ºE) in 32 meters of water where they found less than 0.1ºC gradient between surface and deep waters. Similar to Wolanski et al. (1981) Orr recorded temperature on a weekly basis [32]. Our study was different to previous studies of the GBR Lagoon both spatially and temporally. Stratification events that we found in Nelly Bay were short duration events lasting less than one day. For most of the observation period the waters in Nelly Bay were well mixed (less than 0.1ºC vertical gradient) and thus if the temperature was recorded weekly the stratification events most probably would not be reflected in the collected temperature data. The important implication of this study is that sea surface temperature (SST) data alone may not be reflecting the full complexity of oceanographic processes within lagoon.

B. Influence of stratification on plankton distribution

Thermal stratification of the water column influenced the distribution of some planktonic taxa. Stratification often affects the distribution of phytoplankton [33]. In this study there were strong trends for nauplius larvae, *Coscinodiscus* spp. and *Ceratium* spp. to be more abundant in shallow waters when the water column was stratified. Vertical differences in abundance were most obvious in nauplius larvae and *Coscinodiscus* spp, both of which were more abundant in shallow waters during stratification.

C. Utilization of real time WSN for oceanographic studies

The ability to explore the relationship between stratification and distribution of plankton is often hampered by a lack of real time data. We demonstrated that utilization of WSN can provide real time communication of temperature data about oceanographic events and consequently biological sampling can be planned to address specific hypotheses. For example, timely information about stratification events would facilitate exploration of the biological effect that physical oceanography has on plankton. Unfortunately due to technical problems with WSN we did not receive real time data about the second stratification event on 17th of September and thus plankton samples were not collected during this event. This limited our ability to do comparative analysis of plankton abundance during thermal stratification but at the same time highlighted the advantages that WSN technology can offer for plankton studies.

WSN offers several advantages over historical monitoring techniques by streamlining the data collection process, potentially minimizing human errors and time delays, reducing overall cost of data collection, and significantly increasing the quantity and quality of data both temporally and spatially [34]. Wireless sensor networks allow fine grained interface between the virtual and physical worlds and thus represent the future for environmental monitoring [35]. Future studies would be able to utilize wireless sensor network to trigger plankton collection once the water temperature anomalies were detected. The design of the system has to be more robust to be able to survive in the aggressive sea water environment [36] and temperature sensors would have to be calibrated to industry standards prior to deployment.

V. CONCLUSION

This study demonstrated that short term stratification can occur in shallow tropical waters and influence the distribution of plankton. This challenges the traditional view that waters of the GBR Lagoon are always well mixed and that surface values of temperature and salinity are representative of the whole water column. Stratification was caused by cooling in shallow water at night. Greatest warming in shallow water happened during low tides and at low tide amplitude phases of the lunar cycle. The ability to observe changes in phytoplankton production at the time or shortly after physical changes in the water column is crucial to furthering understanding of the trophic dynamics of marine ecosystems. The current study highlights the utility of real time WSN as a means of achieving this goal.

ACKNOWLEDGMENT

The authors gratefully acknowledge the assistance of:

- The Ambient Systems electronic engineers: Supriyo Chatterjea, Wouter van Kleunen and Johan Kuperus
- Dr Paul Havinga, Ambient Systems
- A/Prof M. Palaniswami, the ARC Research Network on Intelligent Sensors, Sensor Networks and Information Processing, The University of Melbourne
- Financial support from the Telstra Foundation and the ARC Research Network on Intelligent

Sensors, Sensor Networks and Information Processing

- Lorna Hempstead, Magnetic Island, Australia

REFERENCES

1. Bondarenko, O., S. Kininmonth, and M. Kingsford, *Coral Reef Sensor Network Deployment for Collecting Real Time 3-D Temperature Data with Correlation to Plankton Assemblages.* In Proceedings for SENSORCOMM 2007, Valencia, Spain 2007.
2. Hamner, W.M., M.S. Jones, J.H. Carleton, I.R. Hauri, and D. Williams, *Zooplankton, planktivorous fish, and water currents on a windward reefr face: Great Barrier Reef, Australia.* Bulletin of Marine Science, 1998. **42**(3): p. 459-479.
3. Kingsford, M.J. and A.B. MacDiarmid, *Interrelations between planktivorous reef fish and zooplankton in temperate waters.* Marine Ecology Progress Series, 1988. **48**: p. 103-117.
4. Furnas, M.J. and A.W. Mitchell, *Nutrient inputs into the central Great Barrier Reef (Australia) from subsurface intrusions of Coral Sea waters: a two-dimensional displacement model.* Continental Shelf Research, 1996. **16**(9): p. 1127-1148.
5. Furnas, M.J. and A.W. Mitchell, *Phytoplankton dynamics in the central Great Barrier Reef : Seasonal changes in biomass and community structure and their relation to intrusive activity.* Continental Shelf Research, 1986. **6**: p. 363-384.
6. Steel, J.H., *Spatial pattern in plankton communities*. Vol. 1. 1978: New York : Plenum Press.
7. Dustan, P. and J.L. Pinckney, Jr., *Tidally Induced Estuarine Phytoplankton Patchiness.* Limnology and Oceanography, 1989. **34**(2): p. 410-419.
8. Thurman, H.V., *Introductory Oceanography.* Macmillan Pub. Co. NY, 1994.
9. Kolber, Z.S., *ECOLOGY: Getting a Better Picture of the Ocean's Nitrogen Budget.* Science, 2006. **312**(5779): p. 1479-1480.
10. Thomson, R.E. and E. Wolanski, *Tidal period upwellings within Raine Island Entrance, Great Barrier Reef.* Journal of Marine Research, 1984.
11. Capone, G.D., G.P. Zehr, H.W. Paerl, B. Bergman, and E.J. Carpenter, *Trichodesmium, a globally significant marine cyanobacterium.* Science, 1997. **276**(May).
12. Mann, K.H. and J.R.N. Lazier, *Dynamics of Marine Ecosystems: Biological-Physical Interactions in the Oceans.* Blackwell, Cambridge, MA., 1991: p. 11-15.
13. Wolanski, E. and G.L. Pickard, *Upwelling by internal tides and Kelvin waves at the continental shelf break on the Great Barrier Reef.* Austtralian Journal of Marine Research, 1983. **34**: p. 65-80.
14. Leichter, J.J. and S.L. Miller, *Predicting high-frequency upwelling: Spatial and temporal patterns of temperature anomalies on a Florida coral reef.* Continental Shelf Research, 1999. **19**: p. 911-928.
15. Andrews, J.C. and M.J. Furnas, *Subsurface intrusions of Coral Sea water into the central Great Barrier Reef I. Structures and shelf-scale dynamics.* Continental Shelf Research, 1985. **6**: p. 491-514.
16. Andrews, J.C., *Thermal waves on the Queensland shelf.* Marine and Freshwater Research, 1983. **34**(1): p. 81-96.
17. Zamon, J.E., *Tidal changes in copepod abundance and maintenance of a summer Coscinodiscus bloom in the southern San Juan Channel, San Juan Islands, USA.* Marine Ecology Progress Series, 2002. **226**: p. 193-210.
18. Church, J.A., J.C. Andrews, and F.M. Boland, *Tidal currents in the central Great Barrier Reef.* Continental Shelf Research, 1985. **4**(5): p. 515-531.
19. Pineda, J.S., *Predictable Upwelling and the Shoreward Transport of Planktonic Larvae by Internal Tidal Bores.* Science, 1991. **253**(5019): p. 548-551.
20. Pineda, J., *Internal tidal bores in the nearshore: Warm-water fronts, seaward gravity currents and the onshore transport of neustonic larvae.* Journal of Marine Research, 1994. **52**(3).
21. Schulz, J., C. Mollmann, and H.J. Hirche, *Vertical zonation of the zooplankton community in the Central Baltic Sea in relation to hydrographic stratification as revealed by multivariate discriminant function and canonical analysis.* Journal of Marine Systems, 2007. **67**(1-2): p. 47-58.

22. Gallager, S.M., H. Yamazaki, and C.S. Davis, *Contribution of fine scale vertical structure and swimming behaviour to formation of plankton layers on Georges Bank.* MARINE ECOLOGY PROGRESS SERIES, 2004. **267**: p. 27-43.

23. Morris, I., *The Physiological Ecology of Phytoplankton.* Blackwell Scientific Publications, 1980: p. 58-420.

24. Kininmonth, S., et al., *Sensor Networking the Great Barrier Reef.* Spatial Sciences Qld, 2004(spring): p. 34-38.

25. Chatterjea, S., S. Kininmonth, and P. Havinga, *Sensor Networks.* GEO connexion International magazine, 2006(October): p. 20-22.

26. Sprintall, J. and M.F. Cronin, *Upper ocean vertical structure.* In: J. Steele, S. Thorpe, and K. Turekian (eds.), Encyclopedia of Ocean Sciences, First Online Update, Academic Press, London, UK, 2008.

27. Sutherland, W.J., *Ecological Census Techniques.* 1996: p. pp 169-170.

28. Kingsford, M.J. and C. Battershill, *Studying temperate marine environments, a handbook for ecologists.* Canterbury University press, 1998: p. 92-172.

29. Andrews, J.C. and P. Gentien, *Upwelling as a Source of Nutrients for the Great Barrier Reef Ecosystems: A Solution to Darwin's Question?* Marine Ecology Progress Series, 1982. **8**: p. 257-269.

30. Wolanski, E., M. Jones, and W.T. Williams, *Physical Properties of thr Great Barrier Reef Lagoon Waters near Townsville. II Seasonal Variations.* Australian Journal of Freshwater Research, 1981. **32**: p. 321-3334.

31. Wolanski, E. and P. Ridd, *Mixing and trapping in Australian tropical coastal waters.* Coastal Estuarine Studies, 1990. **38**: p. 165-183.

32. Pickard, G.L., J.R. Donguy, C. Henin, and F. Rougerie, *A review of the physical oceanography of the Great Barrier Reef and Western Coral Sea.* Australian Institute of Marine Science monograph series, 1977. **2**: p. 20-52.

33. Sournia, A., *Phytoplankton manual.* UNESCO press, 1978: p. 75-87.

34. Glasgow, H.B., J.M. Burkholder, R.E. Reed, A.J. Lewitus, and J.E. Kleinman, *Real-time remote monitoring of water quality: a review of current applications, and advancements in sensor, telemetry, and computing technologies.* Experimental Marine Biology and Ecology, 2004. **300**: p. 409-448.

35. Krishnamachari, B., *Networking Wireless Sensors.* Cambridge University Press, 2005.

36. Bondarenko, O., S. Kininmonth, and M. Kingsford, *Underwater Sensor Networks, Oceanography and Plankton Assemblages.* In Proceedings for ISSNIP 2007, Melbourne, Australia 2007: p. 657-662.

Target Tracking in Marine Wireless Sensor Networks

Ahmed M. Mahdy and Jonathan M. Groenke
Department of Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX USA
ahmed.mahdy@tamucc.edu and jgroenke@islander.tamucc.edu

Abstract— **It is evident that wireless sensor networking has a secure place in the future of communication systems. The potential and promise of such networks cannot be underestimated. A major advantage is the wide array of configurations and working environments that allow a wealth of applications. In the last few years, research on marine wireless sensor networks has been growing steadily. Nonetheless, there is an immediate need for further investigation into basic and applied research. In this paper, major challenges and applications are discussed. A perspective on target tracking in marine wireless sensor networks is thoroughly presented. In addition, the basic concept for underwater tracking algorithms is presented. From inhabitant monitoring to homeland security, the underwater environment is a major user of target tracking. The military field has greatly benefited from terrestrial wireless sensor networks. We strongly believe that the case with marine wireless sensor networks is no different. This paper discusses how underwater target tracking may enhance digital battlefields of the future. We also present a two-layer broadband wireless infrastructure for marine/terrestrial sensor networks with military and homeland security applications. The development of such infrastructure enhances the survivability of ad hoc networks, widens the domain of applicability, and emphasizes the role of marine wireless sensor networks in future battlefields.**

Keywords-marine wireless sensor networks; underwater communication; target tracking; digital battlefields

I. INTRODUCTION

Modern communication systems have been enjoying the unique capabilities of the wireless technology for the last few decades. Wireless systems have introduced new possibilities and opened the door for novel technologies that made wireless communications a first choice to a wide range of applications in a variety of domains. The versatility and flexibility of the wireless technology have enabled the design of revolutionary systems including satellite, cellular, and ad hoc networks through the years.

The communications discipline has substantially progressed over the last century. From the early telegraphs and telephony, to mobile systems, and to recent optical wireless systems, the field has witnessed numerous advances on all levels. Nonetheless, wireless communications represent a major factor in the way communications have evolved. The use of wireless transmission added new dimensions to the human perception of communications especially with the evolution of cellular and satellite systems. Cellular systems are infrastructure based networks divided into a group of contagious cells. Every cell is serviced by an access point (i.e. base station) that controls data traffic. Ideally, a cell is circular with a radius equal to the transmission range of a centered base station. However, hexagonal cells make the actual case. The ability to communicate on the move has enabled tremendous applications and fuelled related research especially within the areas of network protocols and data multiplexing. Satellite systems with their line of sight communications have triggered extensive research leading to novel transceiver designs. The application of satellite systems affects our daily life whether it is a TV broadcast, weather forecast, or a Global Positioning System (GPS) satellite. The impact of wireless systems on all aspects of our modern life is clear.

Among the many flavors of wireless systems, ad hoc and sensor networks represent a remarkable and unique kind of wireless communications. An ad hoc network is a multihop wireless system; an autonomous system that is composed of wireless nodes communicating in a peer to peer fashion where every node serves as a host and router at the same time [2][3]. A node may be mobile or stationary. Nodes exchange information packets that usually travel in a store and forward manner. A wireless sensor network is a collection of tiny sensor devices with wireless capabilities. A sensor device includes sensing, processing, communication, and battery modules. A sensor node can locally process data and communicate with other nodes to form a network of tiny sensors. The placement of such nodes usually follow a random approach, which eases the setup process and facilitates the deployment of sensor networks in temporary locations such as disaster relief areas. Nevertheless, this complicates the design process of sensor networks' algorithms and protocols, which have to be self-organized.

Despite the fact that wireless sensor networks have a wide range of applications that have ensured a secure place in the future for this technology, research and development of underwater wireless sensor networks have been minimal [4]. Basic and applied research on the development of this unique kind of wireless sensor networks is greatly needed. In this paper, we present major challenges and applications focusing on the role of marine wireless sensor networks in future digital battlefields. The rest of this paper is organized as follows. In Section II, major research challenges are presented. In Section III, a discussion on target tracking in marine wireless sensor networks is presented. Sections IV and V emphasize the use of these networks in military applications. The paper is concluded in Section VI.

II. Marine Wireless Sensor Networks

The current protocols and design specifications of terrestrial wireless sensor networks cannot accommodate the requirements of the marine environment [5]. This makes the deployment of current wireless sensor networks in a marine environment challenging [6]. New techniques and algorithms are needed to achieve this goal. The following presents major design issues that need immediate attention.

A. Types of Sensors

Sensor nodes are the building components of wireless sensor networks. A sensor node is a tiny device that has the ability to sense, process, and communicate data. These nodes are usually deployed in harsh environments such as battlefields and disaster relief areas. Therefore, the manufacturing of sensor nodes handles this problem by providing reliable sensors that can stand unfriendly conditions. Nonetheless, most of these sensors cannot survive marine environments. Water can have severe impact on the operation of sensors. The effect of water on sensor devices and the different characteristics of waterproof sensors need to be closely studied. In addition, we believe that the marine environment opens the door for the introduction of new capabilities of the sensor devices.

Many applications of marine sensor networks will require sensors to monitor the underwater conditions. This necessitates the need for sensor nodes that can either reside under the water surface permanently or temporarily. Accordingly, new features of sensor devices will be introduced, which may lead to a redesign of existing sensor nodes. A typical marine sensor network will be composed of floating and diving sensor nodes. The former will be nodes that can float on the water surface and collect relevant data such as surface temperature, wave frequency…etc. On the other hand, diving sensors are those nodes residing under the surface of the water collecting data like pressure, depth, visual imaging…etc. It is imperative to understand the different factors that affect the operation of such nodes. Each type has its own and unique characteristics that should be reflected on the design of these sensors. A careful study on these factors should be conducted and a comparative analysis of the different characteristics should be carried out.

B. Network Architecture

Ad hoc networks can be classified according to the existence of an infrastructure support. In fact, ad hoc networks do not necessarily require an infrastructure, which leads to two major classes; infrastructureless and infrastructured ad hoc networks. Infrastructureless ad hoc networks are pure decentralized systems where all participating nodes equally share network management responsibilities. This organization has been known for its simplicity and has dominated the design of ad hoc networks. Yet, its performance has never been to the expectations due to the inefficiency of such decentralized systems [7][8][9][10]. This triggered the development of more efficient infrastructured organizations, centralizing some functionalities of the network, has been inspired by the classic base station cellular scheme. The hierarchical clusterhead model has proved to be the best practical solution for infrastructured ad hoc networks. The network is virtually divided into clusters managed by designated nodes called clusterheads. Despite the fact that many clustering techniques have been proposed for the clusterhead based scheme, the selection of clusterheads has been mostly based on a single quality measure. This severely limits the efficiency of the selection process and can degrade the network performance. For example, consider a connectivity based selection that assigns the node with the highest connectivity degree to be the clusterhead. In this case, a highly connected node with low energy level can be selected as a clusterhead. Such a node, with the extra load a clusterhead is exposed to, may become out of energy in a short period of time triggering a new selection round, which increases the overhead and affects the stability of the network. We believe that clusterhead selection should not be based on only one measure but rather a set of quality measures. Moreover, the selection criterion should be scalable in order to accommodate new measures and/or disable existing measures. The effect of the clustering techniques on the network performance is commonly evaluated in terms of network stability and fairness (i.e. load balance). Network stability is adversely proportional to the number of clusterhead replacement; the less the number of selected clusterheads is, the more stable the network is. On the other hand, an ideally fair clustering technique uniformly distributes the management load over the network nodes. The more the number of nodes involved in the management of the network is, the fairer the technique is. Clearly, there is a tradeoff between network stability and fairness. One of the two merits is sacrificed on the account of the other in many cases for the sake of simpler designs. This can significantly deteriorate the network performance. Therefore, the clustering technique should be able to strike a tradeoff between stability and fairness in order to achieve better overall performance. We believe that clustering techniques should be adaptable and can be easily configured to seek specific network merits. To illustrate, the same technique can be configured for maximum stability, maximum fairness, or optimized overall performance.

C. *Network Protocols*

The impact of the marine environment on the operation of sensor networks cannot be underestimated. Basic communication protocols and techniques will need to be redesigned to reflect the new requirements of such an environment [11][12]. For example, existing routing algorithms cannot serve marine sensor networks in their current format [13][14]. A distinction between the floating part of the network and its diving counterpart will be essential for a proper operation of the network. Therefore, the algorithm should reflect the associated cost with every communication link by providing a relative weighing factor distinguishing all-surface, surface-underwater, and all-underwater links' costs.

It may be necessary for a wireless sensor network to update its topology to alleviate the effect of recent changes to the environment that degrade or even halt the operation of the system. In such cases, the network seeks the best possible topology that can furnish an optimized performance given the current conditions. A best possible topology for conventional sensor networks may not be the same for a marine sensor network due to the heterogeneity of communication links and the impact of water on signal transmission. This raises the need for new algorithms that can handle the topology reconfigurability in the water. The network efficiency highly depends on the performance of clusterheads. A clusterhead failure might result in a cluster failure triggering the isolation of its nodes. Accordingly, clusterheads must be qualified enough and adequately selected to claim such responsibility. Ideally, the outcome of the selection process should improve the performance of the cluster as well as the whole network. Currently available selection algorithms need to be revisited to reflect the changes in the environment and the characteristics of the sensor devices. The selection of a diving sensor as a clusterhead may not be appropriate in a network dominated by floating nodes.

A vital aspect of sensor networks is the energy consumption due to the size and working lifetime of sensing nodes. Therefore, low-energy protocols are essential for the operation of marine sensor networks. New internetworking schemes are also needed to allow better communication between sensor networks and other non sensor networks. In addition, error control protocols for mobile sensor networks are highly appreciated. Dedicated coding schemes to improve communication quality between nodes are needed.

Another major issue is network security. The rapid growth rate of communication networks and the open nature of nowadays information resources have turned network security into a mandatory requirement for communication systems. Military, business, and personal information are major types of data exchange across networks. Such information is mostly intended to designated recipients. Vulnerability of such networks imposes a major design challenge for secure data transmission. With this concern in mind, security is on the top of design priorities in current networks especially wireless sensor networks. Several security measures are required including authorization and authentication. Relevant research has shown that secure protocols for the recently developed communication technologies, especially those involving mobility, are easily recognizable. Routing algorithms with security guarantee are needed. Related research focuses on the different types of attacks to help measure the vulnerability of a network. In this context, detection of anomaly in communication behaviors can help prevent possible attacks. More attention to this issue should be given. All of these areas are widely open for research especially for the unique marine environment. Further discussion can be found in [15]. In this paper, we focus on target tracking algorithms due to the wide range of applications that may benefit from such capabilities.

III. A Perspective on Target Tracking using Marine Wireless Sensor Networks

Target tracking is a skill currently used by individuals in occupations such as wildlife management to locate and track various animals, and military organization to detect and track the enemy on a battlefield. For example, in wildlife management, there are many cases where a specific animal needs to be located or tracked for various studies. Depending on the environment it may be very difficult for any person or group to deploy in order to locate a specific animal and gather required information. This is especially true if the animal's habit is not well suited for humans. For example, if marine biologists are trying to track and record what specific species of fish eat in a certain area of the ocean, it might be more prudent to deploy relatively low cost sensor nodes to record data in the area of interest instead of deploying a team of scientists to collect the same data. Deploying sensor nodes can be less dangerous and more cost effective than having a group of people live on a boat in the middle of the ocean for some indeterminate amount of time. This also holds true for other inhospitable environments such as a desert or tropical jungle. Military applications of WSN for target tracking may be more apparent. On the battlefield there are numerous scenarios where sensor nodes would be preferred over human deployment. For instance, if some military leader needs to know when or if the enemy is transiting some specific area of interest then it is much safer to deploy a network of sensors instead of sending a team of soldiers.

Issues to be considered with mobile target tracking include types of nodes, node distribution, initial target detection, target localization, target classification, and sustained target tracking. Node type discusses basic types of sensor nodes and how they can be configured. Node distribution covers how nodes are to be distributed given specific target characteristics. Target detection discusses how the network detects a target. Localization deals with the transition from target detection to sustained target tracking. Classification and sustained tracking cover how the network classifies the target and tracks its movement.

One of the first items to consider prior to determining the type of sensor node to use is the target's characteristics. Knowing the characteristics is essential to identify what target traits can be exploited. In order to better illustrate this point, consider an example of a marine biologist who needs to locate a specific species of whale off the coast of Alaska. The biologist wants to locate as many of the whales as possible so it can tag and track them for further study. The biologist knows the general area where the whale lives but the area covers hundreds of square miles and he does not have the manpower to effectively cover the territory. The biologist wishes to utilize sensor networks that will be deployed in strategic areas and notify him when a whale is located.

There must be some classifying trait that emanates from the whale such as sound. If the biologist heard this particular noise he could immediately identify the sound as a characteristic of the target whale and classify the target as the whale of interest. We can now use this identifying characteristic to choose or configure a specific sensor. In this example, the sensor must be able to detect sound in a particular frequency range with the use of a hydrophone. Sensors can be configured to sense one or many frequencies in the spectrum and they are not limited to sound.

Sensors can be categorized as passive or active. Passive sensors are more suitable for target tracking since they do not disrupt the environment in which they are placed and they do not alert the target to anything unnatural. These passive sensors can be a) omnidirectional which simply detects a signal within a particular range from any three dimensional directions, or b) directional which means they can provide a relative bearing to the signal of interest. Active sensors do not rely on the target to emit light, sound, or electromagnetic energy. Instead the active sensor transmits some form of energy and detects any alteration of that energy caused by the target. For example, an underwater sensor can produce sound and sense any echo that bounces off an object in the water. The advantage of using this type of sensors over a passive sensor is the higher probability of accurate tracking data. In some cases, this active signal will illuminate the target more than the targets natural emissions. However, the disadvantage is the target will be alerted to the tracking devices and it may alter its course away from the sensor field. Also, the energy needed to produce an active signal will deplete the battery life much faster than that of a passive sensor.

Sensors can detect many different phenomena and it is paramount the right sensor is used when trying to locate a target. Using the correct type of sensor is the foundation for the rest of the network. The sensor must be able to detect the exploitable characteristics of the target or the entire system may fail.

Prior to searching, the sensor's storage device will have to be preloaded with the relevant target information. In order for the sensor to detect the target, each sensor will have to discern between the target of interest and any other frequency it may detect. When a node is first deployed one of the first tasks the node must execute is an ambient noise reading. Ambient noise is background noise which is always present and the sensor will use this reading as a baseline for all other signals received. Having the ability to compare a received signal with ambient levels and prerecorded target data is an important feature. The end user is only interested in target information and he/she is not concerned with all received signals. Therefore, we do not want every sensor node to pass everything it detects to the base station or end user. This will expend much of the network's power by transmitting data that could be of no use. If each node can determine if the frequency received matches that of the desired target then each node will only pass relevant data. This will allow for efficient message passing which helps maximize quality of service and increase energy efficiency.

The sensor node distribution is important because sensors must be placed in such a manner that the network will have the highest probability of target detection. One factor to consider when designing a distribution pattern is the detection range of the sensor. This will require some prior research of the target to determine a source level for each detectable characteristic. If the user wishes to exploit a specific frequency then he will need to know the average distance from the source that the frequency will be detectable. For example, if the sensor uses a hydrophone to detect a target then the source of the sound must be close enough to the sensor for the sensor to detect the sound. Using our whale scenario, an average range of the emanated sound from the whale must be predetermined in order to know how far apart we want to place our sensors. For example, if we want to use the song of the whale as a target identifier. We will assume the biologist has done some research and knows the average distance from which a whale's song can be heard is 5 kilometers direct path. Therefore, for a high probability of detection the sensor must be within 5 kilometers of the whale to detect the song. This means the distance between any two nodes should be no greater than 10 kilometers to ensure no holes exist inside the network. Nodes can be positioned closer together to increase probability of detection but then area covered will be reduced. Conversely, the nodes can be positioned farther apart to cover more area but then probability of detection will decrease. Node placement all depends on probability of detection and the desired area coverage.

Once the detection range is resolved it must be compared to the effective communication range between sensor nodes. If the sensor nodes are too far apart to communicate then spacing will have to shrink. A cost analysis will have to determine the best course of action. One solution would be

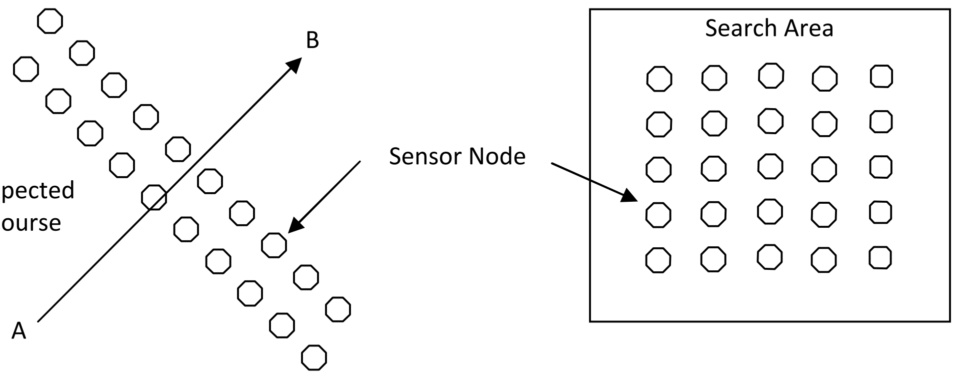


Figure 1. An Example of a Network Distribution Pattern; 2x10 and 5x5 patterns

to use sensor nodes with larger batteries and powerful transmitters but this raises the price of each node. If the nodes are placed closer together then the target may be alerted and it may alter its natural course or speed of advance. This is a case where an Autonomous Underwater Vehicle (AUV) [16] would be a good option to cover the voids within the network grid.

After the node spacing is decided, it is necessary to determine the overall physical shape of the distributed sensor node network. This shape, of course, depends on the target. If the target is going to transit from some point ‘A’ direct to another point ‘B’ then an elongated pattern configured perpendicular to the line from ‘A’ to ‘B’ would be favored; shown in Figure 1.a. A good example would be a 2 by 10 pattern with 10 kilometer spacing between nodes and the center node positioned on the expected route of travel. A top down view of the nodes would look like a 100 kilometer long pattern. This will provide a good probability of detecting our transiting whale. If we determine that our target is not transiting and is, instead, feeding or mating in a certain area then our pattern will have to adapt. The shape of the distribution pattern should match the search area while maintaining our predetermine node spacing as shown in Figure 1.b. A typical example would be n nodes laid out in a √n by √n pattern.

Since the sensor nodes are placed in specific locations relative to each other it is easier to manage the topology of the network. Topology control is an important issue that determines which nodes are allowed to, or able to, communicate with surrounding nodes. When there is a predetermined plan where each node will be placed then it is easier to control and maintain the topology. Depending on the network structure and the nodes task designation the node will become self aware of its relative position when the node is alive. If we are using a flat network structure were all nodes play the same role then each node will find its location in the network by finding its neighbors and the topology control protocol will establish links. If it is a hierarchical network structure then, after each node discovers it relative position within the network, the topology protocol will establish master nodes or clusterheads. These superior nodes are either chosen through some election protocol or they are preselected because they possess some characteristics the surrounding nodes do not. The nodes will self organize in their respective clusters and communicate with the clusterhead/master node. Homogeneous nodes may alternate roles between slave and master status depending on the network protocol [17].

When the search area is large, the nodes will most likely be deployed from a moving airplane, boat, or truck so node placement is estimated and not exact. This differs from random node distribution where nodes are deployed in some arbitrary pattern leading to unpredictable node density. This type of deployment is faster but the topology control is more difficult to establish and maintain. The nodes will still form a predetermined flat or hierarchical network but the initial establishment of communication links will take more processing power. When nodes are distributed randomly, there is no way to determine if every node will be connected to the network. The possibility of multiple networks within the entire set of nodes is possible. On the other hand, if nodes are placed in a prearranged manner then this possibility is greatly reduced.

Routing techniques are vital for such systems because the information must go from the sensor to some clusterhead or base station when the target is detected. In the case of the clusterhead, it must receive all relevant information from the sensor nodes in order to locate, track, and classify the target. The information must transfer from node to node with minimal latency. The routing technique mentioned in [17] discusses virtual grid architecture routing.

The network is constructed in "clusters that are fixed, equal, adjacent, and no overlapping with symmetrical shapes." Each cluster has a clusterhead that performs data aggregation at a local level (i.e. within the cluster) while a subset of clusterheads perform data aggregation at a global level. The clusterheads are responsible for tracking targets moving throughout their respective clusters. The data is routed from the sensors to the clusterhead. In [18], a different approach with location-aware routing and processing is discussed. In this case, the nodes are self organized in cells upon the occurrence of an event of interest. The event in this case is target detection. When a target is detected, the nodes detecting the target are organized in a cell and elect a manager node which creates new cells. As the target moves through the sensor field, the nodes route data to the current manager node which processes the data to track the target. There are numerous routing techniques that can be used. The application takes a toll on determining which technique to use.

Once the network is configured, deployed, and activated the network works autonomously to detect the target. Target detection begins when one or more of the sensor nodes detect a frequency that matches a target frequency in its database. If the target is moving, which is likely the case; the detected frequency may not be the exact frequency in the data base due to the Doppler effect. The node will have to process this information in order to determine a probability or confidence level that the frequency is the target of interest. If the node determines that the frequency does belong to the target then it will pass this information to the manager node (or clusterhead). The manager node will process this information and match it with any other event data received. If the manager node has not received any messages from other nodes it might send a request message or alert message to other nodes in the vicinity of the sensor which received the target frequency. Once other nodes detect the target and pass the respective information to the manager node, the manager node will attempt to localize the target and determine an estimated course and speed of the target. The manager node will send a message to other manager nodes in the direction of the target. These other manager nodes can put their respective clusters on alert if they have been asleep to conserve energy. One of the manager nodes will have to eventually collect enough data points to positively classify the target as the target of interest and then send a message to the base station which will alert the end user.

A. Target Tracking Algorithms

Following the aforementioned discussion, developing a tracking algorithm for such underwater applications is greatly based on the classical Doppler equation:

$$\boldsymbol{f_o} = \frac{(\boldsymbol{c} + \boldsymbol{v_o})}{(\boldsymbol{c} + \boldsymbol{v_s})} \boldsymbol{f_s} \qquad (1)$$

Where: $\boldsymbol{f_o}$= the frequency observed by the listener. $\boldsymbol{f_S}$ = the frequency emitted by the source. $\boldsymbol{c}$ = the speed of the wave through the medium. $\boldsymbol{v_o}$ = the velocity of the listener through the medium. $\boldsymbol{v_S}$ = the velocity of the source through the medium

The basic algorithm concept is for a given sensor to detect an acoustic frequency $\boldsymbol{f_o}$ of a target source $\boldsymbol{f_S}$ and calculate the velocity of the target $\boldsymbol{v_s}$. If the observed frequency is higher than that of the source frequency then the target is moving toward the sensor. Conversely, if the observed frequency is less than that of the source frequency then the target is moving away from the sensor. The location of the target is automatically known to be in the vicinity of the sensor due to the fact that the sensor detected the target. The direction of the target can be approximated if other sensor nodes detect the same target. It should be noted that this Doppler equation is only accurate when the source is moving directly to or from the listener. However, the level of inaccuracy for targets that do not move directly to or from a sensor is acceptable considering the large coverage areas in an ocean environment. It is also assumed that the acoustical path from the source to the sensor is a direct path.

For this concept to work properly, the node placement and the acoustical characteristics of the target must be predetermined. Sensor nodes must be spaced at a distance so that only one or two sensors can detect the target at any one point in time. As previously discussed, in order to ensure appropriate sensor spacing, we must take into account the predicted range or our target. Assuming the target is in the vicinity of our sensor network, if the detection range of our target is smaller than expected then, at most, only one sensor will detect the target at a given point in time. If the detection range is larger than expected then no fewer than two sensor nodes will detect the target at any given point in time. The tracking algorithm can still work in these two cases but it will require more processing and more intra-network communication which will result in more power consumption. For the purpose of this research, the assumption is made that the predicted detection ranges of the target are accurate and spacing between any two sensors is twice that of the predicted range. Once the sensor spacing has been determined the next step is to construct a sensor pattern. We believe that a honeycomb pattern, illustrated in Figure 2, fits the majority of underwater tracking applications.

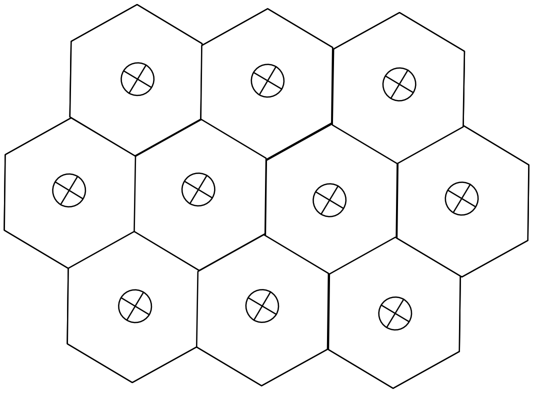


Figure 2. A Honeycomb Pattern

This orientation will provide minimal overlap in sensor coverage and negate any coverage gaps. Each hexagon is divided into six sections with the sensor node positioned at the center and each of the six sections is given a number and relative bearing as shown in Figure 3. Neighboring nodes are predetermined and programmed into each sensor. Furthermore, each sensor node is aware of its neighbors' relative location. This configuration will aid in allowing the sensor nodes to pass relative positions of the target. When a sensor detects a target the only immediate conclusion that can be made is that the target is within the detection radius of a sphere surrounding the sensor. Using the Doppler equation and the stored source frequency the sensor can determine an approximate velocity.

The stored source frequency is loaded into the sensor and based on historical target observations. Multiple source frequencies can be loaded in the sensor for the same target. Using a threshold for acceptable velocities, the sensor can ignore observed frequencies if they do not match target characteristics. For example, suppose a sensor observes a frequency of 80 Hz and has two stored frequencies of 30 Hz and 150Hz.

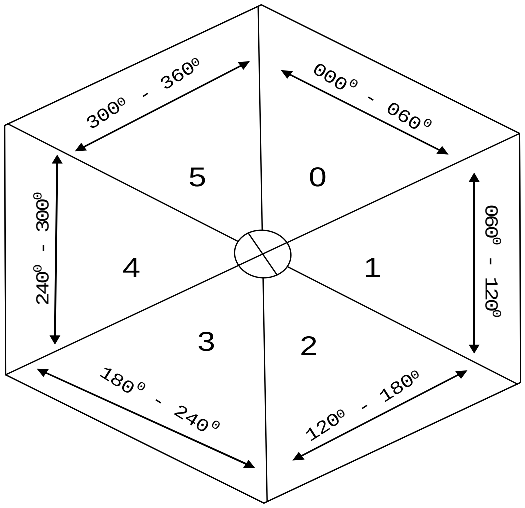


Figure 3. A Honeycomb Cell

The stored frequencies are used as frequency emitted by the source in the Doppler equation. The stored historical target data gives an acceptable velocity of two to seven knots. Computing the velocity of the observed frequency would yield 1820 knots and 2548 knots respectively. This observed frequency would be discarded by the sensor. If the sensor observed a frequency of 149.8Hz then the computed velocity would be 2329 knots and 4 knots respectively. This observed frequency would be accepted and the sensor would continue to monitor this frequency.

In the following, we present a tracking algorithm that can be used in marine wireless sensor networks (i.e. Figure 4 and Figure 5):

1) Sensor receives frequency and compares it with preset frequencies in its tiny database through the use of the Doppler equation.
2) If velocity is out of predetermined tolerance then discard. Otherwise, send 'contact' message to neighboring nodes.
3) Receive 'contact' replies from neighbors. If no positive contact replies from neighbors then send contact report with time stamp to clusterhead. Wait for another signal and/or continue monitoring frequency
4) If positive contact from a neighbor is received, then match neighbor's ID with sector and determine approximate relative direction. Send contact report with time stamp to clusterhead.
5) If a contact message is received by neighbor and no received frequencies match target data, reply to neighbors with 'no contact' message. If monitoring a frequency that matches target data then reply to neighbor with 'contact' message and then calculate approximate direction of target.

With respect to communication between nodes, all stationary nodes will be connected in a 2-D mesh with no wrap around. That is, each node will only be able to communicate with its immediate neighbor north, south, east, and west. When the sensor nodes are given their respective grid coordinates, they are also given the coordinates and identification of their immediate neighbors. The AUV will be able to communicate only with its closest neighbor. To find its closest neighbor, the AUV will have to broadcast its position and wait for a return message from the closest stationary sensor node. Different algorithms may be developed based on the concepts discussed in this Section.

IV. APPLICATIONS OF MARINE WIRELESS SENSOR NETWORKS WITH TARGET TRACKING

Marine wireless sensor networks offer an unmatched option to a wide range of different domains. The significance of the aforementioned research lies in the fact that it opens the door for a variety of applications as well as new areas of relevant research in wireless networks. In the following, we present candidate areas that highly benefit from a marine wireless network with target tracking capabilities.

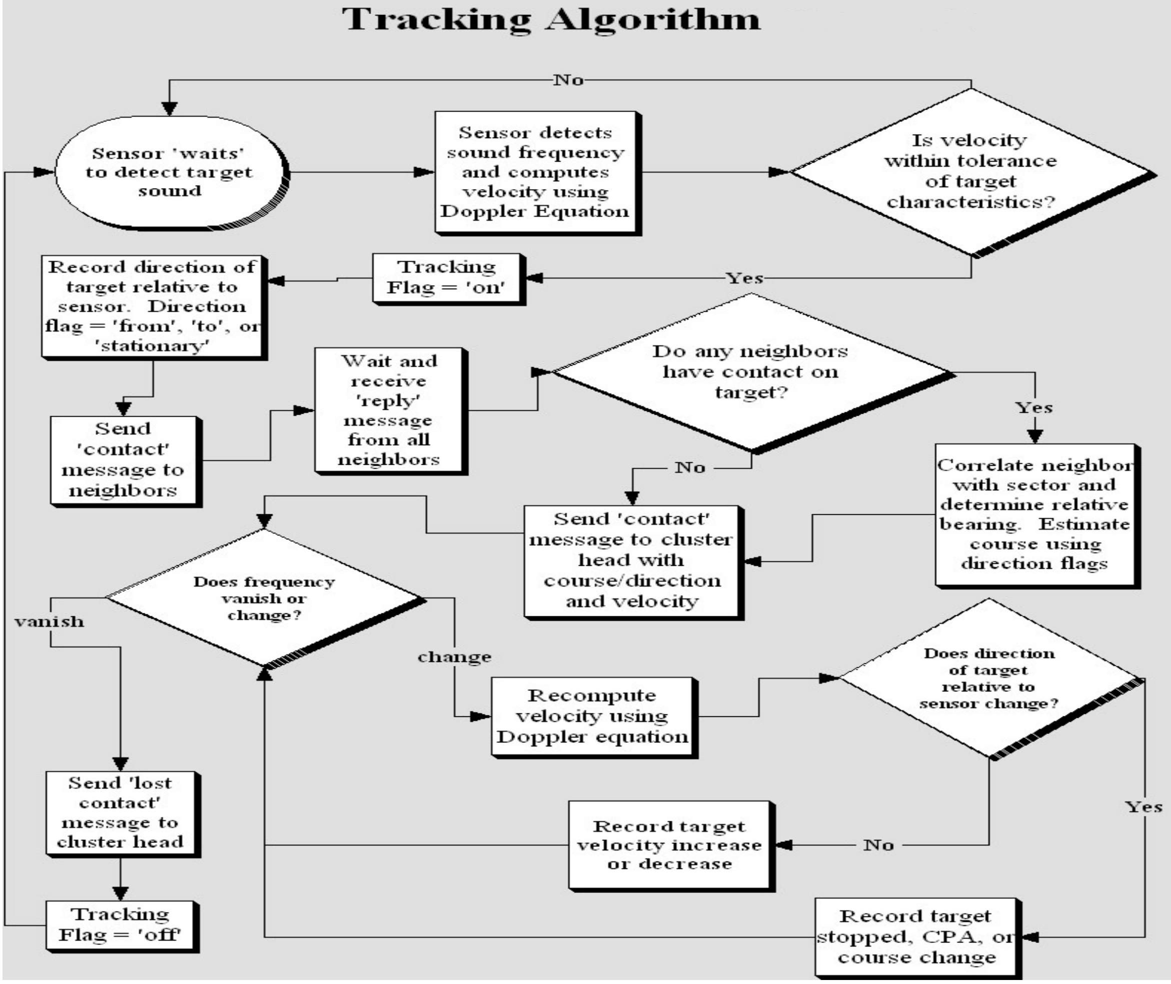

Tracking Algorithm
No
Sensor 'waits' to detect target sound
Sensor detects sound frequency and computes velocity using Doppler Equation
Is velocity within tolerance of target characteristics?
Record direction of target relative to sensor. Direction flag = 'from', 'to', or 'stationary'
Tracking Flag = 'on'
Yes
Wait and receive 'reply' message from all neighbors
Do any neighbors have contact on target?
Send 'contact' message to neighbors
Yes
No
Correlate neighbor with sector and determine relative bearing. Estimate course using direction flags
Send 'contact' message to cluster head with course/direction and velocity
Does frequency vanish or change?
vanish
change
Recompute velocity using Doppler equation
Does direction of target relative to sensor change?
Send 'lost contact' message to cluster head
Yes
Record target velocity increase or decrease
No
Tracking Flag = 'off'
Record target stopped, CPA, or course change

Figure 4. Flow Chart for the Tracking Algorithm

Military and Homeland Security: The land-based applications of sensor networks in the military and homeland security domains show how significant wireless sensor networks can be. Smart uniforms equipped with sensor nodes can automatically report data on the status of troops and their locations. Tanks and military vehicles can also be equipped with sensor nodes forming a wireless network connecting the different units of the army. Real-time border sensor networks represent a great asset in policing the borders and reporting any suspicious movements. All of these applications can be mapped to their corresponding marine applications only if a wireless sensor network can be deployed in the water.

Ocean Inhabitant Research: The possibility of having sensor nodes diving in the ocean collecting data about the different inhabitants offers a unique opportunity for ocean life can play a major role in bringing ocean research to new levels.

V. Marine Wireless Sensor Networks in Future Digital Battlefields

According to the “Army Research Office In Review document” [1], the need for *on-the-move mobile wireless networks* that can be deployed in the battlefields of the future cannot be underestimated. Sensor networking presents an essential component of future digital battlefields. The main idea is to develop a secure two-layer broadband wireless infrastructure for mobile ad hoc and sensor networks with military and homeland security applications that take advantage of underwater sensors that can track targets of interest. The development of such infrastructure enhances the survivability of ad hoc networks and widens the domain of applicability by ensuring secure communications with broadband capabilities that can meet

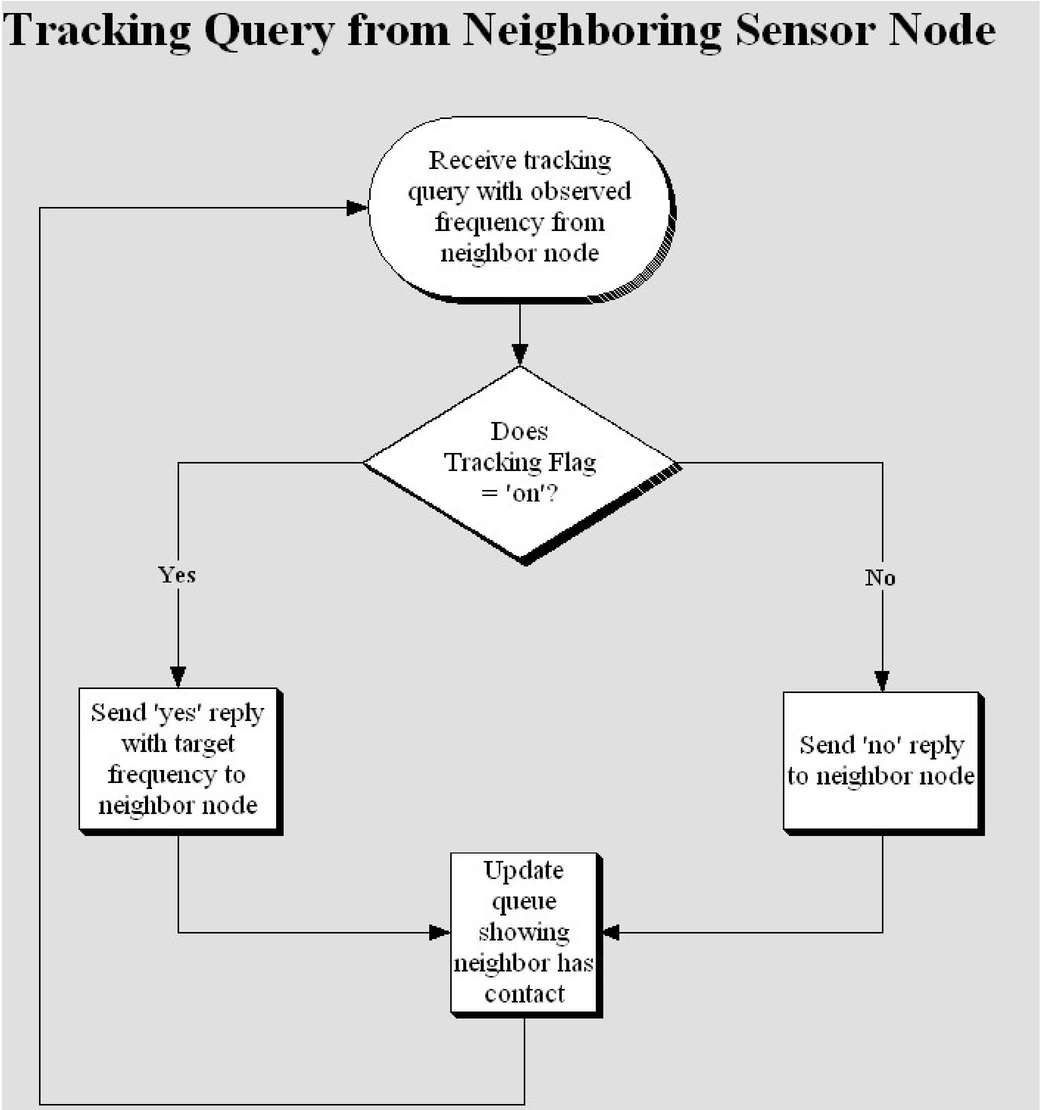

Tracking Query from Neighboring Sensor Node
Receive tracking query with observed frequency from neighbor node
Does Tracking Flag = 'on'?
Yes
No
Send 'yes' reply with target frequency to neighbor node
Send 'no' reply to neighbor node
Update queue showing neighbor has contact

Figure 5. Flow Chart for the Tracking Query

the requirements of expected future digital battlefield applications. More specifically, improving the efficiency of the virtual organization of ad hoc networks through scalable clustering techniques and the support of a secure wireless broadband backbone are the scope of this work. We present a two layer networking infrastructure with virtual as well as physical support. The bottom layer is a virtual layer that addresses the efficiency issues of infrastructureless ad hoc networks by providing a hierarchical organization. The network is divided into clusters where every cluster is supervised by a clusterhead. Clustering methods are designed to seek specific network merits such as security, stability, and load balance. The proposed infrastructure includes further support through a second layer of broadband backbone. This top layer furnishes the network with broadband-enabled nodes with communication capacity that meets the growing demands on high transmission rates unlike existing ad hoc networks. This layer provides an alternate communication path for isolated nodes ensuring a connected network and robust military operations. This will alleviate the problem of poor reliability of existing ad hoc networks leading to better scalability. In addition, these nodes will function as gateways allowing broadband communications with other networking platforms independent of the underlying technology, thus conferring to a major requirement of future wireless systems. Figure 6 depicts a hypothetical battlefield sensor network based on the proposed architecture. This network can be rapidly deployed allowing a wide range of entities to securely communicate for fast information sharing and better decision making. The bottom layer forms a mobile ad hoc network of army robots, underwater sensor nodes, troopers, tanks, vehicles, sensors…etc. The top layer is a backbone that provides a broadband wireless cloud using high-speed wireless technologies such as optical wireless, ultra-wideband, WiMAX and/or WiFi. Helicopters, robots, and ships can be equipped to operate as backbone communication nodes. The following explains how this research satisfies the Army Future Force operational goals and how it is highly relevant to the army research and needs.

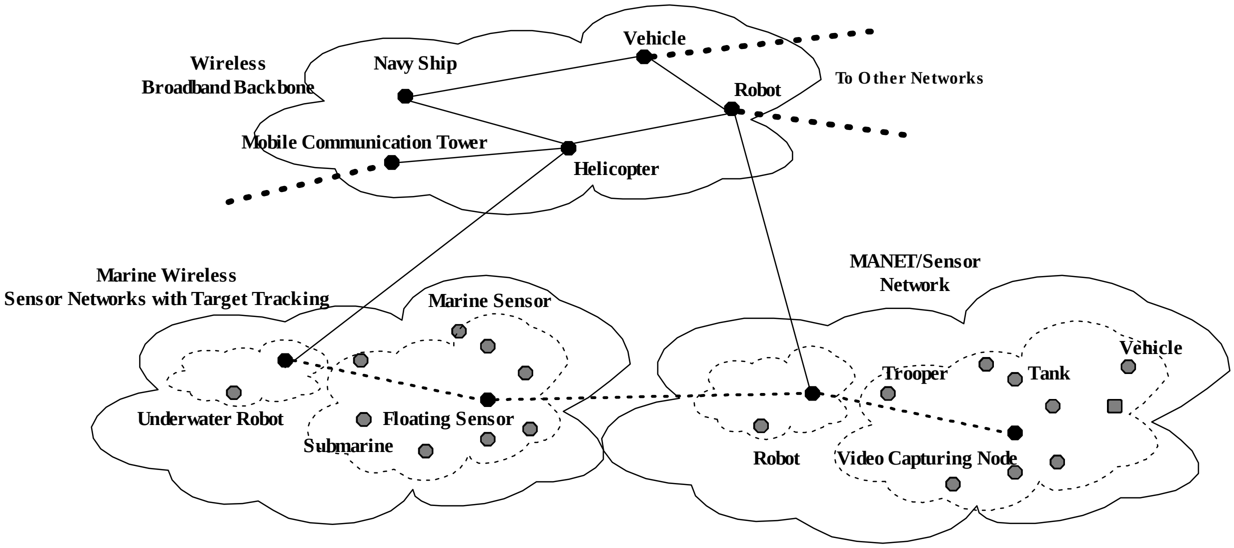


Figure 6. An Example of How Marine Wireless Sensor Networks can be used in Future Digital Battlefields

Communications and Networks Needs:

The Army Research Office In Review explicitly states its major communications and networks needs as follows [19]

- *“Mobile wireless communications networks will be required that are both adaptive and can operate on-the-move.”*
- *“New sensor, communication, and weapon systems based on unmanned robotic and teleoperated aerial and ground vehicles must be developed.”*
- *“The concepts of light, agile forces and the digital battlefields require a seamless…and highly mobile wireless communication system with a highly dynamic topology.”*

The proposed architecture directly addresses the above army research goals by providing a self-organizing mobile ad hoc network that can be rapidly deployed. The bottom layer of the proposed architecture furnishes the network with a flexible topology that can timely respond to environment changes. Moreover, the nature of mobile ad hoc networks allows the use of marine sensor networks, robots, and vehicles to serve as communication nodes (see Figure 1). The ad hoc layer of the proposed infrastructure will address the performance and efficiency issues. This layer is organized in the form of clusters and is dynamically established in two steps. Clusters are, first, formed following a cluster formation technique, and then clusterheads are selected. The top layer of the proposed architecture provides a backbone through the use of broadband wireless technologies that mostly operate in a point-to-point fashion (ex, optical wireless or ultra-wideband) achieving highly secure communication links with high immunity to jamming and interference.

VI. CONCLUSIONS

Current research on wireless sensor networks is based on the assumption that these networks are deployed in a terrestrial environment. Relevant protocols and design specifications are developed under this condition. This makes the deployment of currently existing wireless sensor networks in marine environments extremely challenging. In this paper, we present major challenges and applications of underwater wireless sensor networks. Target tracking in such environments is emphasized. A basic target tracking algorithm for cluster-based marine wireless sensor networks is presented. We discuss application areas that highly benefit from a marine network with target tracking such as military and inhabitant monitoring. We highlight the army’s immediate needs for secure agile broadband communications for future digital battlefields emphasizing the role of marine sensors with target tracking capabilities. A two-layer architecture for broadband terrestrial/marine ad hoc and sensor networks that can provide warfighters with a secure on-the-move means for high-capacity communications is discussed.

ACKNOWLEDGMENT

This work was supported, in part, by an NSF CRI grant and a Texas Research Development Fund grant.

REFERENCES

[1] A.M. Mahdy, “Marine Wireless Sensor Networks: Challenges and Applications," Proceedings of the 7th International Conference on Networking (ICN), 2008, pp 530-535.

[2] D.P. Agrawal and Q-A. Zeng, “Introduction to Wireless and Mobile Systems,” Thomson Brooks Cole, USA, 2003.

[3] D.B. Johnson and D.A. Maltz, “The Dynamic Source Routing Protocol in Ad Hoc Networks,” Mobile Computing, T. Imielinski and H. Korth eds., Culwer, 1996, pp. 152-181.

[4] I.F. Akyildiz, D. Pompili, and T. Melodia, “Underwater Acoustic Sensor Networks: Research Challenges”, Ad Hoc Networks, Volume 3, Issue 3, 2005, pp. 257-279.

[5] J.H. Cui, J. Kong, M. Gerla, and S. Zhou, “The Challenges of Building Scalable Mobile Underwater Wireless Sensor Networks for Aquatic Applications”, IEEE Network, 2006, pp.12-17.

[6] I. Vasilescu, K. Kotay, and D. Rus, “Krill: An Exploration in Underwater Sensor Networks”, Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors, 2005.

[7] J.N. Al-Karaki, A. Kamal, and R. Ul-Mostafa, “On the Optimal Clustering in Mobile Ad Hoc Networks,” Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC), 2004, pp. 71-76.

[8] W. Lou and J. Wu, “A Cluster-Based Backbone Infrastructure for Broadcasting in MANETs,” Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS), 2003.

[9] J. Waldo, “Constructing Ad Hoc Networks,” Proceedings of the IEEE International Symposium on Network Computing and Applications (NCA), 2001, pp. 9-20.

[10] Y. Song and D. Park, “Infrastructure Support for Ad Hoc Network: When and Where?,” Proceedings of the 2004 International Conference on Parallel Processing Workshops (ICPPW), 2004, pp. 141-147.

[11] P.C. Etter, “Underwater Acoustic Modeling, Principles, Techniques and Applications,” 2nd edition, E&FN Spon, 1996.

[12] D.B. Kifoyle, and A.B. Baggeroer, “The State of the Art in Underwater Acoustic Telemetry”, IEEE Journal of Oceanic Engineering, (OE-25, no. 5), 2000, pp. 4-27.

[13] D. Makhija, P. Kumaraswamy, and R. Roy, “Challenges And Design of Mac Protocol For Underwater Acoustic Sensor Networks”, 4th IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2006, pp.1-6.

[14] J. Proakis, J. Rice, E. Sozer, and M. Stojanovic, “Shallow water acoustic networks,” Encyclopedia of Telecommunications, John Wiley and Sons, 2003.

[15] J. Partan, J. Kurose, and B.N. Levine. “A Survey of Practical Issues in Underwater Networks,” Proceedings of the 1st ACM International Workshop on Underwater Networks, 2006, pp 17-24.

[16] S. Pai, P. Guerrini, and J. Potter, “Autonomous initial capture systems for AUV recovery,” Proceedings of the 3rd International Conference and Exibition for Underwater Acoustic Measurements: Technologies and Results, 2009.

[17] J.N. Al-Karaki, and A.E. Kamal, “Routing techniques in wireless sensor networks,” IEEE Wireless Communications, Volume 11, Issue 6, 2004, pp 6-28.

[18] R.R. Brooks, P. Ramanathan, and A.M. Sayeed, “Distributed target classification and tracking in sensor networks.” Proceedings of the IEEE. Volume 91, Issue 8, 2003, pp. 116 –1171.

[19] Army Research Office In Review, http://www.arl.army.mil/main/main/default.cfm?Action=29&Page=172, (last visited June 2010).

An Integrating Platform for Environmental Monitoring in Museums Based on Wireless Sensor Networks

Laura M. Rodríguez Peralta, Lina M. Pestana Leão de Brito
Centro de Ciências Matemáticas (CCM)
Exact Sciences and Engineering Competence Centre (CCCEE)
University of Madeira (UMa),
Campus da Penteada, 9000-390 Funchal, Madeira, Portugal
{lmrodrig, lina}@uma.pt

Abstract – **Monitoring the museum's environment for preventive conservation of art purposes is one major concern to all museums. In order to properly conserve the artwork it is critical to continuously measure some parameters, such as temperature, relative humidity, light and, also, pollutants, either in storage or exhibition rooms. The deployment of a Wireless Sensor Network in a museum can help implementing these measurements in real-time, continuously, and in a much easier and cheap way. In this paper, we present the first testbed deployed in an Contemporary Art Museum, located in Madeira Island, Portugal, and the preliminary results of these experiments. On the other hand, we propose a new wireless sensor node that offers some advantages when compared with several commercially available solutions. Furthermore, we present a system that automatically controls the dehumidifying devices, maintaining the humidity at more constant levels.**

Keywords – Art preventive conservation, Awareness tool, Environmental monitoring, Wireless Sensor Networks.

I. INTRODUCTION

Today's museum managers are faced with the constant demanding of gaining greater control of the indoor environment, under increasing budgetary constraints. Those in charge of historic buildings have the added complexity of preserving not only the existing artwork but also the building's historic structure. In this type of environments, it is very important to minimize the visual impact caused by monitoring systems for esthetical reasons. So, both kinds of protections must be accomplished with minimal intrusion from the new system being installed.

The conservation of artwork in museums is a very well known problem, either in exhibition rooms or in archival collections. Monitoring the museum's environment is one of the most important tasks and concerns of all museums. In order to properly conserve the artwork it is critical to continuously measure and control some parameters, such as temperature, relative humidity, light and, also, pollutants (such as: carbon dioxide, several types of acids, dust particulates, etc.).

It is also crucial to consider that the desired values of these parameters depend on the type of material or on the group of materials (typical in contemporary art) that constitute the artwork. So, depending on the type of works that are in exhibition or in storage rooms, different rooms may have different requirements regarding environmental conditions. The main goal of preventive conservation is to maintain the artworks under basically constant levels of, above all, humidity and temperature. However, in the case of rare objects and artefacts, it is required an extremely precise control of temperature and humidity levels.

The deployment of a Wireless Sensor Network (WSN), which is composed by wireless sensor nodes, in a museum can help implementing these measurements continuously, and in a much easier and cheap way. It causes almost no visual impact due to the small size of sensor nodes and to the absence of cables, what is extremely important in a museum. Also, it eliminates the problems inherent to traditional measuring equipments, such as mechanical hygrothermographs, psychometers and hygrometers; there are no moving parts to break and it stays in calibration.

A WSN typically consists of a large number of tiny wireless sensor nodes (often referred to as nodes or motes) that are densely deployed [2]. Nodes measure some ambient conditions in the environment surrounding them. These measurements are, then, transformed into signals that can be processed to reveal some characteristics about the phenomenon. The data collected is routed to special nodes, called sink nodes (or Base Station, BS), typically in a multi-hop basis. Then, the sink node sends data to the user. Depending on the distance between the user and the network, a gateway may be needed in order to bridge both, either through the Internet or satellite.

A sensor node typically consists in five components: sensing, memory, processor, transceiver (transmitter and receiver) and battery. Nowadays, nodes are intended to be small and cheap. Consequently, their resources are limited (typically, limited battery, reduced memory and processing capabilities). Moreover, due to short transmission range (caused by restrained transmission power), nodes can only communicate locally, with a certain number of local neighbours [2]. So, nodes have to collaborate in order to accomplish their tasks: sensing, signal processing, computing, routing, localization, security, etc. Consequently, WSNs are, by their nature, collaborative networks [3].

Taking advantage of wireless communications, WSNs allow for a wide range of applications: environmental monitoring, catastrophe monitoring, health, surveillance, traffic monitoring, structural monitoring, security, military, industry, agriculture, home, etc.

In this paper, which is an extension version of paper [1], we describe the experimental deployment of a small WSN carried out in a contemporary art museum called Fortaleza São Tiago, in Madeira Island. This WSN aims at monitoring the environment of the museum, for artwork and for building conservation purposes. We highlight all the problems identified at the initial phase, which influenced the final deployment of the complete WSN. We also present a new wireless sensor node that we have developed specifically to environmental monitoring applications, but considering the specific requirements of the museum, for example, reduced size and cost. This is one of the main contributions of our work.

This work was developed in the context of the WISE-MUSE project (Environmental Monitoring based on Wireless Sensor Networks for conservation of artwork and historical archives project).

This paper is organized as follows. In section II, we describe the related work in the area of WSNs applied to the monitoring of museums or historical buildings. Section III presents the problem of environmental monitoring of museums and the use of WSNs as a cheap and suitable solution. Still in the same section, a practical testbed deployed in a museum is described, as well as all the problems identified. The results of these experiments are, also, described. Section IV presents the proposed WISE-MUSE sensor node and Section V presents the humidity control device. Finally, Section VI provides some conclusions and some perspectives of future work.

II. Related Work

There are some works that are related to the deployment of WSNs in museums. However, the most common applications are, usually, to use WSNs for security reasons [4, 6]; or to monitor the number and distribution of visitors in the museum [5]; or for the creation of interactive museums [5, 7, 8].

There are also some wireless equipments that measure humidity and temperature commercially available (Omega, 2007 [9], 2DI 2007 [10]), however they have bigger dimensions and they are more expensive than wireless sensor nodes.

Spinwave Systems offer a solution for preserving a building's architecture using WSNs [11]. Spinwave provide precise control and monitoring of environmental variables, such as temperature and humidity, in buildings where wired sensors are not feasible or are prohibitively expensive. However, they do not monitor light or pollutants. Besides, the nodes are expensive. They claim to ensure minimal disruption to building occupants and improved indoor climate. But, nodes are quite big for being applied in an environment where the visual impact is of extreme importance.

Lee et *al.* [12] present a scenario of applying WSNs to monitor the environment of art galleries, but focusing in measuring only humidity and temperature. However, the paper focus on a different problem, i.e., on evaluating the efficiency of the ALOHA protocol without retransmission, when transmitting from sensor nodes to a base station. Del Curto & Raimondi [13] present a work where WSNs have been used for preserving historic buildings. Crossbow manufacturers [14] present two systems to be applied in museums and archives: a system that monitors humidity and temperature, the CLIMUS, and a system that controls the air conditioning unit, the REAQUIS. However, they do not use WSNs; sensors used are wired are much bigger than wireless sensor nodes commercially available.

Our goal is to create a WSN for monitoring not only humidity and temperature, but also light and pollutants, in a museum. We are also using smaller and cheaper wireless sensor nodes than the ones used by [11]. These factors are the basis that makes our solution more suitable to the environmental monitoring of museums or historical building.

III. Applying WSNs in Museums Environmental Monitoring

Today's museum managers are faced with the constant demanding of gaining greater control of the indoor environment for preventive conservation of art purposes, under increasing budgetary constraints. In the particular case of Fortaleza São Tiago, in Madeira Island, Portugal, environmental measurements are performed in a very rudimental way, using traditional and very expensive measuring equipments, such as mechanical hygrothermographs, psychometers and hygrometers. These measurements take too long to be performed in the all museum and require a specialized person for this purpose. These equipments require calibration and caused visual impact on visitors and on exhibition rooms. And, so far, these measurements haven't been performed as often as it should. The administration choices regarding a more flexible and practical solution are limited by severe budgetary constraints.

The deployment of a WSN in a museum can help implementing these critical measurements automatically, continuously, and in a much easier and cheap way. It causes almost no visual impact due to the small size of sensor nodes and to the absence of cables, what is extremely important in a museum. Also, it eliminates the problems inherent to traditional measuring equipments; there are no moving parts to break and it stays in calibration.

Our goal is to create a WSN for monitoring not only humidity and temperature, but also light and some pollutants, in a museum, for art conservation purposes. This work was developed in the context of an ongoing project named WISE-MUSE, which aims to applying WSNs to museum environmental and structural monitoring and automatic control.

Our initial deployment at the museum of contemporary art, Fortaleza São Tiago, has served as a proof-of-concept and consisted of a small group of wireless nodes capturing environmental data continuously. Data collected by sensor nodes is sent wirelessly to a database, through the sink node that is connected to a PC. Data can be visualized, in real time, through a web page, in different ways: tables, graphics, colour gradients, etc.

As Fig. 1 illustrates, we have defined, essentially, five phases involved in the environmental monitoring of a museum. The first phase regards the monitoring of temperature, humidity, light and pollutants; then, collected data is sent to a data repository; after that, this data can be visualized in different formats (graphics, tables, colour gradients, etc.); afterwards, data is analyzed to verify its compliance with the art conservation rules; finally, as a last phase, the environment conditions are automatically optimized accordingly to the analysis results.

A. *Field Deployment*

With this first experimental testbed, we aimed at testing the behaviour of the wireless sensor nodes and at identifying some problems regarding both the nodes and the application scenario. This way, we were able to choose the sensor nodes that best suit to this specific application.

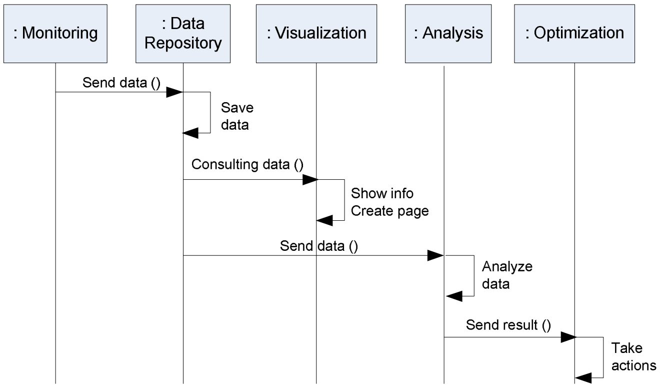


Fig. 1. Phases involved in a museum's environmental monitoring.

As this was an experimental testbed, we have only deployed three wireless sensor nodes and one sink node (or base station), as Fig. 2 illustrates. So, we decided to install three Crossbow [14] mica2 motes, equipped with an MTS400CA data acquisition board and a mote equipped with a mib520 board, functioning as the base station. The radio transceiver of these nodes operates at the 868,919 MHz ISI band and they communicate using the ZigBee [19] communication protocol.

The MTS400CA data acquisition board measures: ambient light, relative humidity, temperature, 2-axis accelerometer, and barometric pressure. This board has a common humidity and temperature sensor, the Sensirion SHT11, which can measure a temperature range from -10 to +60°C and a humidity range from 0 to 90% RH. The ambient light sensor is a TAOS TSL2550D, which has a spectral response similar to human eye [14]. In our future experiments, we intend to replace the humidity and temperature sensor by the Sensirion SHT15, because it has a higher accuracy than the Sensirion SHT11.

Mica2 nodes are smaller than Spinwave [11] nodes, with 58 x 32 x 7mm (excluding the battery pack) opposed to 121 x 70 x 25 mm, respectively. Nevertheless, our WISE-MUSE sensor node implemented in this project is smaller and cheaper than Mica2 node, which will be described next. Fig. 3 shows mica2 motes equipped with the battery pack.

Currently, this WSN measures the most important parameters, which are temperature, humidity and light; however, internal voltage is also monitored so that the user is aware of the state of the nodes' batteries. The sensor nodes are programmed to measure and send data each 60 seconds.

B. *Preliminary Results*

As explained before, all collected data is centralized in a PC, connected to the Internet. Fig. 4 shows some screenshots of the web interface created for the specific application of museums' environmental monitoring. In this figure we present the 24 hours' graphics of temperature, humidity, light and internal voltage.

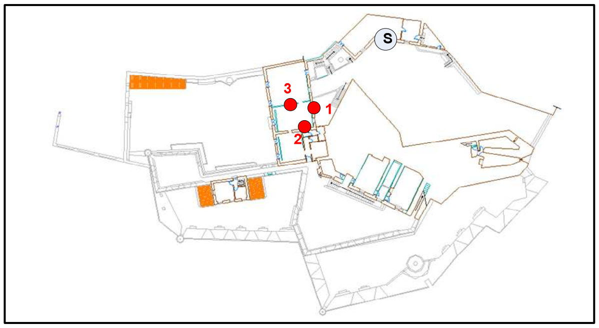


Fig. 2. Experimental WSN deployment in Museum Fortaleza São Tiago.

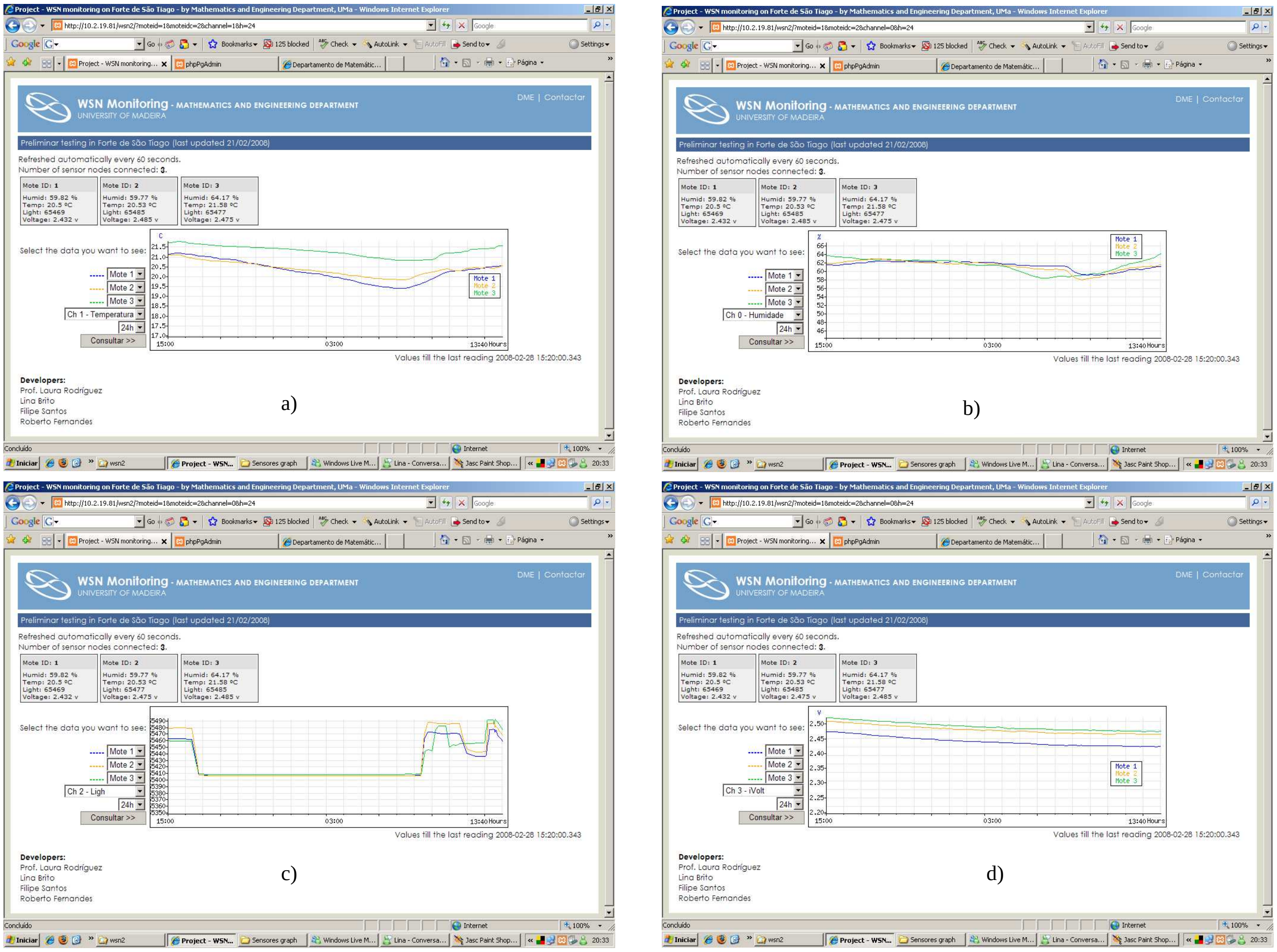
Fig. 3. Crossbow Mica2 mote.

After analysing this data, it was possible to achieve some conclusions about the behaviour and the type of variations of temperature and humidity throughout the day and night, in a single room of the museum. Considering the graphics of each node, we verified that temperature varies about 2ºC whereas humidity varies about 6% RH (see Fig. 4 a) and b)). According to the museums' managers, these variations are typical during the winter, in this museum; however, these parameters usually vary a lot more in summer. Fig. 4 c) shows that light varies, essentially, accordingly to the day and night periods, or when the lights are turned on or off (corresponds to the variations at the end of the graphic).

So, we verified that these parameters are not as constant as they ideally should be. Therefore, it is very important that the WSN is connected to the air conditioning and dehumidifying systems in order to automatically control, above all, the temperature and the humidity of these rooms.

As mentioned before, the sensor nodes are programmed to measure and send data each 60 seconds. Fig. 4 d) illustrates the decrease of batteries level during a 24 hours' period.

To evaluate the duration of batteries, we changed this value by programming the nodes to perform one measurement and transmission each 10 seconds. Comparing the internal voltage parameter obtained in both cases, we concluded that the duration of the batteries is proportional to the energy consumption. In this specific experiment, the batteries lasted about 6 times less than in the first case.



Project - WSN monitoring on Forte de São Tiago - by Mathematics and Engineering Department, UMa - Windows Internet Explorer

http://10.2.19.81/wsn2/?moteid=1&moteidc=2&channel=1&h=24

WSN Monitoring - MATHEMATICS AND ENGINEERING DEPARTMENT
UNIVERSITY OF MADEIRA

Preliminar testing in Forte de São Tiago (last updated 21/02/2008)

Refreshed automatically every 60 seconds.
Number of sensor nodes connected: **3**.

Mote ID: 1
Humid: 59.82 %
Temp: 20.5 ºC
Light: 65469
Voltage: 2.432 v

Mote ID: 2
Humid: 59.77 %
Temp: 20.53 ºC
Light: 65485
Voltage: 2.485 v

Mote ID: 3
Humid: 64.17 %
Temp: 21.58 ºC
Light: 65477
Voltage: 2.475 v

Select the data you want to see: Mote 1, Mote 2, Mote 3, Ch 1 - Temperatura, 24h

Values till the last reading 2008-02-28 15:20:00.343

Developers:
Prof. Laura Rodríguez
Lina Brito
Filipe Santos
Roberto Fernandes

a)

Project - WSN monitoring on Forte de São Tiago - by Mathematics and Engineering Department, UMa - Windows Internet Explorer

http://10.2.19.81/wsn2/?moteid=1&moteidc=2&channel=0&h=24

WSN Monitoring - MATHEMATICS AND ENGINEERING DEPARTMENT
UNIVERSITY OF MADEIRA

Preliminar testing in Forte de São Tiago (last updated 21/02/2008)

Refreshed automatically every 60 seconds.
Number of sensor nodes connected: **3**.

Mote ID: 1
Humid: 59.82 %
Temp: 20.5 ºC
Light: 65469
Voltage: 2.432 v

Mote ID: 2
Humid: 59.77 %
Temp: 20.53 ºC
Light: 65485
Voltage: 2.485 v

Mote ID: 3
Humid: 64.17 %
Temp: 21.58 ºC
Light: 65477
Voltage: 2.475 v

Select the data you want to see: Mote 1, Mote 2, Mote 3, Ch 0 - Humidade, 24h

Values till the last reading 2008-02-28 15:20:00.343

Developers:
Prof. Laura Rodríguez
Lina Brito
Filipe Santos
Roberto Fernandes

b)

Project - WSN monitoring on Forte de São Tiago - by Mathematics and Engineering Department, UMa - Windows Internet Explorer

http://10.2.19.81/wsn2/?moteid=1&moteidc=2&channel=0&h=24

WSN Monitoring - MATHEMATICS AND ENGINEERING DEPARTMENT
UNIVERSITY OF MADEIRA

Preliminar testing in Forte de São Tiago (last updated 21/02/2008)

Refreshed automatically every 60 seconds.
Number of sensor nodes connected: **3**.

Mote ID: 1
Humid: 59.82 %
Temp: 20.5 ºC
Light: 65469
Voltage: 2.432 v

Mote ID: 2
Humid: 59.77 %
Temp: 20.53 ºC
Light: 65477
Voltage: 2.475 v

Mote ID: 3
Humid: 64.17 %
Temp: 21.58 ºC
Light: 65485
Voltage: 2.485 v

Select the data you want to see: Mote 1, Mote 2, Mote 3, Ch 2 - Ligh, 24h

Values till the last reading 2008-02-28 15:20:00.343

Developers:
Prof. Laura Rodríguez
Lina Brito
Filipe Santos
Roberto Fernandes

c)

Project - WSN monitoring on Forte de São Tiago - by Mathematics and Engineering Department, UMa - Windows Internet Explorer

http://10.2.19.81/wsn2/?moteid=1&moteidc=2&channel=0&h=24

WSN Monitoring - MATHEMATICS AND ENGINEERING DEPARTMENT
UNIVERSITY OF MADEIRA

Preliminar testing in Forte de São Tiago (last updated 21/02/2008)

Refreshed automatically every 60 seconds.
Number of sensor nodes connected: **3**.

Mote ID: 1
Humid: 59.82 %
Temp: 20.5 ºC
Light: 65469
Voltage: 2.432 v

Mote ID: 2
Humid: 59.77 %
Temp: 20.53 ºC
Light: 65485
Voltage: 2.475 v

Mote ID: 3
Humid: 64.17 %
Temp: 21.58 ºC
Light: 65477
Voltage: 2.485 v

Select the data you want to see: Mote 1, Mote 2, Mote 3, Ch 3 - iVolt, 24h

Values till the last reading 2008-02-28 15:20:00.343

Developers:
Prof. Laura Rodríguez
Lina Brito
Filipe Santos
Roberto Fernandes

d)

Fig. 4. Initial Web interface created to the museum' environmental monitoring application, showing the 24 hours' graphics of temperature (a), humidity (b), light (c) and internal voltage (d) measurements.

C. *Problems Identified*

Several problems were identified during this experimental phase. These problems are related, essentially, with the type of building where the WSN was deployed and, also, with the hardware characteristics and resource limitations of the wireless sensor nodes.

Type of building - Fortaleza São Tiago was built in the 17th century as a military fortress. Therefore, the width of the walls is very large (from 0,5 to 1m, with 1m being the predominant width). The building has double doors and windows. It has an irregular shape and its rooms and terraces are distributed among three floors, as shown in Fig. 5. All these factors influence and difficult the signal propagation and, consequently, the transmission range of sensor nodes.

Location of the sensor nodes - Sensor nodes must not be placed near to the visitors' passageways, in order to avoid the risk of being stolen or damaged. This will also help to minimize the visual impact caused by the nodes. As can be seen in Fig. 6, nodes were located in rather discrete places. (Fig. 6 shows nodes 1 and 2; node 3 is located on the opposite side of the exposition room, also above an electricity socket, like node 2).

Batteries - The batteries of sensor nodes do not last more than some days and their transmission range is not as good as described in the manufacturer's datasheets, even in line-of-sight conditions. In order to increase the transmission range of sensor nodes, we had to set their transmitting power to its maximum value (5dBm). This obviously affects the energy consumption, which is a typical problem of WSNs. To minimize this problem we have decreased the number of measurements and transmissions per time period (one each 60 seconds). The program can still be changed so that the nodes perform even less measurements and transmissions in order to save more energy. Decreasing the transmission power, by increasing the total number of nodes or using a more efficient antenna, or equipping the nodes with a better radio device will also decrease the energy consumption.

Type of antenna – Nodes are equipped with omnidirectional antennas. Thus, the location of the nodes near the walls can cause signal reflections. Changing the type of antenna to a more efficient one (directional) will improve the signal propagation characteristics and, consequently increase the duration of batteries.

Offset adjustment – Even though the sensors used on the nodes do not have to be calibrated, an offset adjustment has to be made. This way, more reliable measurements are ensured.

Temperature and humidity sensor – This sensor should be placed away from the main circuit board, to avoid that the chip's temperature affects the readings.

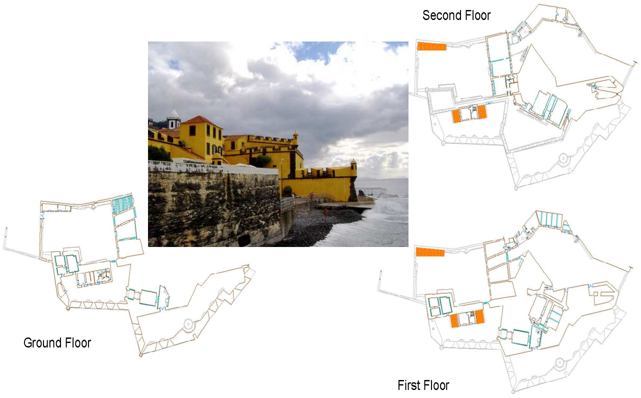


Fig. 5. Museum Fortaleza São Tiago and correspondent floor plants.

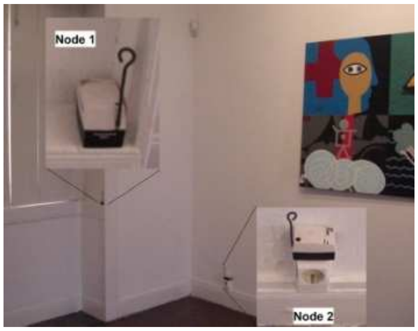


Fig. 6. Wireless Sensor Network deployment in an exposition room of Fortaleza São Tiago

D. *Awareness tool for WSNs visualization*

Opposing to traditional networks, WSNs are only useful if sensor nodes are aware of the environment surrounding them. This means that the great potential of WSNs lies in its ability to correlate collected data in time and in space [17, 18].

This is one of the reasons why we are developing a 3D web-based awareness tool for WSNs visualization, which will be applied to the specific case of a museum's environmental monitoring. This tool is based in the CWSN (Collaborative Wireless Sensor Networks) model, published in [16].

CWSN is a formal model of collaborative work created specifically to the case of WSNs. It is, essentially, a graph-based model; but, it also includes other objects in order to make the modelling of all the entities of a WSN (presented in Table I) possible, which is fundamental to completely represent a WSN. The network hierarchy (from the collected data to the user) can be visualized, as well. Moreover, it is a generic model because it can be applied to heterogeneous networks (any type of nodes, any size, any hardware characteristics, any types of signals, etc.). According to the WSNSCW model definitions, the specific case of this small WSN deployed in the museum will be represented as deployed in Fig. 7. Due to the reduced number of sensor nodes available, data collected by nodes 1, 2 and 3 is sent to the sink node on an one-hop basis. Accordingly, nodes 2 and 3 send data to node 1, which, in turn, sends data to the sink node. The obstacle existent between nodes 2 and 3 is shown in Fig. 8.

Regarding the awareness tool, one of its most important properties is the 3D representation of the network. This is very important so the user can have a more realistic view of the network, becoming more aware of the surrounding environment (different types of terrains, different types of rooms, which obstacles might interfere with the collaboration established between nodes, etc.).

This tool will allow for the visualization of different granularities: fine-grain (sensor nodes), middle-grain (clusters) and coarser (sessions) modelling level. Also, it will allow for an interactive navigation in the map of the network.

We will apply this awareness tool to the specific case of a museum's environmental monitoring. So, data visualization will be enhanced by integrating it in a 3D representation of the museum, giving the user a much more realistic view of the network.

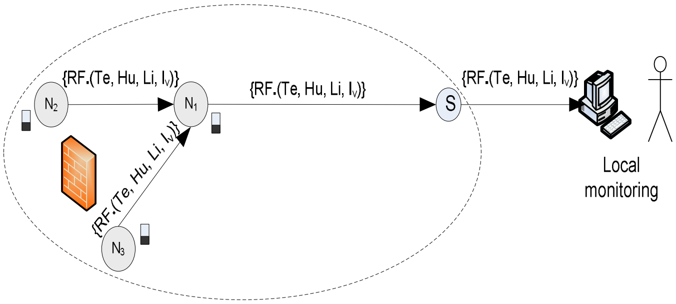


Fig. 7. Representation of the WSN created in Fortaleza São Tiago, using the entities and notations defined in the CWSN Model [16].

Fig. 8. Artwork that obstructs the line-of-sight between nodes 2 and 3, impeding communication.

IV. The WISE-MUSE Sensor Node

We have developed a new wireless sensor node, which is shown in Figure 9. It is designed specifically for environmental monitoring applications, but also considering the specific requirements of the museum, for example, reduced size and cost. This device emerges as the element that collects the environmental parameters, such as temperature, humidity and light. In addition to these three parameters, it is possible to send the battery status (internal voltage) and the RSSI signal. The sensor node transmits the captured data to the base station, via radio frequency (RF).

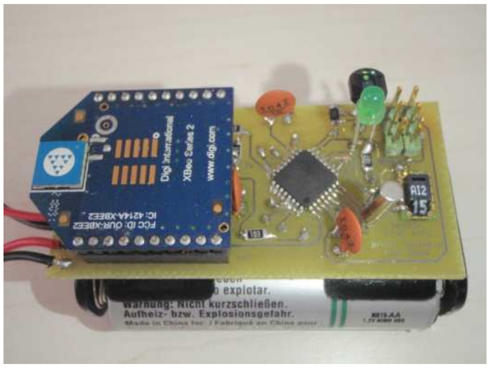


Fig.9. New Wise-Muse sensor node.

The radio module used is the XBee or the XBee PRO, from the Digi manufacturer [20], which operate according to the ZigBee protocol [19], i.e., it is designed according to the IEEE 802.15.4 standard and to support the specific requirements of WSNs (above all, low cost and low power). The ZigBee protocol allows the creation of Personal Area Networks (PAN) networks, supporting several network topologies, namely star, mesh and cluster-tree.

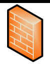
To meet the requirements of the museum in terms of the physical location of the rooms that needed to be covered by the WSN and, consequently, in terms of transmission range, we had to employ a cluster-tree topology. As a result, the type of nodes we needed to deploy were some end devices, some routers and one coordinator. The end devices and the routers were developed by us, whereas the coordinator was acquired to DIGI manufacturer. Nevertheless, the router created can also act as an end device whenever needed; in our experiments, it was used as both a router and as an end device. The differences between the end devices and the routers, at the hardware level, reside in the power supply module since it has to be connected to an electricity socket; the router can never be turned off or go into the sleep mode. At the software level, there are also some differences since it has to be programmed as a ZigBee router, i.e., it has to receive data from the end devices that are wirelessly connected to it (i.e., end devices that are not under the range of the coordinator), and forward this data to the coordinator (sink node).

The WISE-MUSE sensor node, which corresponds to the ZigBee end device, was designed and built from scratch, with a set of components to meet the proposed requirements. This is a node of small dimensions, which causes minimum impact in the Museum. Its low cost is one of the strengths of this sensor node (less than 70€). Another advantage is its low energy consumption.

To be more precise, the WISE-MUSE sensor node has four main blocks:

- The power unit, which is composed of two AA batteries and a step-up circuit that allows to guarantee the supply of a constant power (3.3V) to the microcontroller Xbee and to the sensors;
- The microcontroller that is the "brain" of the node. It receives data from multiple individual sensors, processes it and, then, sends it through an Xbee radio frequency card;
- Two specific sensors: a light sensor that measures the brightness in the rooms of the Museum, and a sensor that measures both temperature and humidity;
- The transceiver module that transmits the collected data.

TABLE I
ENTITIES OF THE CWSN MODEL

Symbol	Concept	Description
○	Sensor node	Wireless sensor nodes, typically with limited resources. These nodes can be either stationary or mobile. Also, they can be in one of two possible states: active or inactive (sleep mode) in order to save energy.
S	Sink node/ Base Station	Node to which data collected by ordinary nodes is sent; being responsible to send data to the gateway.
A	Anchor node	Node with known localization.
⬡	Cluster	Group of nodes, created according to: geographical area, type of sensor, type of phenomenon, task, etc.
CH	Cluster Head	Sensor node to whom all sensor nodes in the cluster send the collected data; it is responsible for sending the received data to the Sink node.
A → B	Relationship	The arrow represents a relationship between nodes A and B. It also represents and adjacency relation between nodes A and B (see section 3.2); nodes A and B are neighbours. A relationship can be established based on: localization, phenomenon, type of sensor node, etc.
{TypeOfSignal.Data} →	Data flow	This label identifies both the type of signal being used (radio frequency, ultrasound, acoustical or light) and the type of data being transmitted between nodes (temperature, humidity, light, sound, video, internal voltage, etc.).
	Gateway	Device responsible to send the data to the user, through the Internet.
	Obstacle	An object (building, tree, rock, etc.), which obstructs line-of-sight between two or more nodes, not allowing for direct communication between them.
◌	Session	In a certain moment, there may be several collaborative sessions in a WSN. A session can be established based on the objective (type of phenomenon to monitor, geographical area to monitor, etc.) of the WSN.
▄	Battery	It represents the percentage of the sensor node's remaining battery.
🧍	User	Person that interacts with the WSN, querying the network, visualizing data, etc. The user customizes the work of the sensor nodes; the data collected by sensor nodes is used by the users' application.

As just mentioned, the microcontroller is the sensor's core. To guarantee an easy programming of this component, it was designed to be easily connected to a programmer, using the AVR-ISP500 protocol (AVR-ISP500 is USB low cost in-system programmer for AVR microcontrollers; it implements the STK500v2 protocol as defined by Atmel. ISP stands for In-System Programmable). Using this connection, the code can be easily updated whenever necessary. The chosen microcontroller is the Atmega 168 (Atmel, 2009), since it presents a set of characteristics that fits almost perfectly all our purposes; its low cost, low consumption and high performance are the main reasons for this choice.

The sensor elements chosen are the SHT15 humidity and the temperature sensor, from the Sensirion Company [21], and the S1087 photocell, from the Hamamatsu Company [22]. The photocell captures the sunlight and returns a value of voltage to the microcontroller, which is then converted to the intensity of sunlight (LUX) unit values. The SHT15 sensor calculates the relative humidity and the temperature values. This is a CMOS industrial device, totally calibrated, that allow for good stability at low cost. Its accuracy is much appropriated considering the requirements of the project (+/-2% for humidity and +/-0.3Cº for temperature).

In order to meet the power autonomy requirements of a sensor node, each node is powered by two AA batteries (1.5V each) that feed the microcontroller, the Xbee module and the sensor STH15. These batteries have a capacity of 2450mAh and an output voltage of 1.5V. Their technology is the Nickel Metal hydride and they weight only 28g.

Finally, we describe the modules for radio transmission frequency (Xbee or Xbee-PRO from Digi manufacturer). These modules were chosen because they require minimal power and because they provide a real and consistent delivery of information between devices, operating at the 2.4GHz ISM frequency band. Although there are other brands for radio transmission modules with lower power consumption, the popularity and the characteristics of Xbee modules determined our choice.

Table II presents a comparison of sensor nodes that could be used in the environmental monitoring of museums. We have considered the nodes' characteristics, as described in the manufacturers' datasheets. It is important to highlight that, in the current market, there are several RF modules separated from the sensing modules. Therefore, we attempt to look beyond the prototype created in the project, by analysing other solutions that perform the collection and transmission of data in these indoor environments.

Analysing Table II, there are several advantages and disadvantages of each solution proposed for monitoring environmental parameters. At the sensing module level, the sensors presented are quite similar. Most of the modules use the SHT15 or STH11 sensors, by Sensirion, and their accuracy does not vary much, about +/-0.3ºC for temperature and about +/-3% for relative humidity.

Almost all sensor nodes collect light, temperature and humidity, with the exception of the Mica2 Sensor Board MTS101CA that can't measure the humidity. The Mica Z reads other kind of data, but this is not necessary in the specific case of the Museum. Besides, Mica Z has the disadvantage of a higher cost.

Regarding the microcontroller there are some considerable differences. Most microcontrollers are manufactured by Atmel, with the exception to the Tmote Sky sensor, which is manufactured by Sentilla [23]. Atmega devices have many features in common with our prototype; The WISE-MUSE prototype presents a smaller flash memory; however, this factor should not be seen as a disadvantage since it conducts to a reduction on the amount of code programmed into the microcontroller. Consequently, the microcontroller has to process less amount of code what leads to a lower power consumption of the node. Therefore, it is not necessary a bigger flash memory. We verified that the energy consumption of our prototype can be reduced when operating at a frequency of 1MHz. In these conditions, its consumption can be decreased to 0.3mA, a value smaller than for the other devices.

The Mica motes [14] must be programmed through a base station, which involves an additional cost. The Tmote Sky and the WISE-MUSE nodes offer an advantage over the others; they are more easily programmed. The Tmote is programmed using USB, while the Wise-Muse prototype is programmed using the Olimex programmer that uses the AVR-ISP500 protocol.

In relation to the transceivers of each sensor node, all of them use the IEEE.802.15.4 protocol, which is the most appropriate protocol for WSNs. The Mica 2 operates in the 868/916Mhhz, 433MHz or 315MHz ranges, while the others use the 2.4GHz range.

We believe that the Xbee module used in the WISE-MUSE prototype is well ranked due to its higher power transmission. Using the Xbee-PRO, which allows an even higher transmission range what is one advantage when comparing to the other nodes. Its disadvantage regards the energy consumption; however, the Xbee is designed to enter in the sleep mode, waking up only in pre-defined intervals to send data; in this way, the problem of energy consumption is minimized.

Figure 10 shows the battery status of the two AA batteries that are used on the WISE-MUSE prototype. We have collected samples during several days to get an estimated battery lifetime. Analysing the graph, we can observe that the batteries' level follows a typical behaviour: in the first 10 days, the battery level dropped from 2.8 to 2.6 volts; however, from that point on, the battery level only dropped 0.1V during the next 30 days, which represents lower battery consumption. Therefore, after 40 days, the sensor node remains operational.

TABLE II
COMPARISON BETWEEN THE WISE-MUSE MOTE AND OTHER COMMERCIALLY AVAILABLE MOTES.

SENSOR NODES					
Characteristics		***WISE-MUSE***	***Mica 2 (MPR400CB) + Sensor Board (MTS101CA) [14]***	***Mica Z (MPR2400CA) + Sensor Board (MTS400CB) [14]***	***Tmote Sky [19]***
Produced by:		The authors at the UMa	Crossbow	Crossbow	Moteiv Corporation
Sensors	*Ambient parameters*	Light, relative humidity, temperature and battery level.	Light, temperature, and prototyping area	Light, relative humidity, temperature, 2-axis accelerometer, and barometric pressure	Humidity, temperature, and light
	Temperature and humidity sensors	SHT15	Termistor	SHT11	SHT11 or SHT15
	Accuracy	Temp: +/- 0.3Cº Hum: +/- 2%	Temp: 0.2 Cº	Temp: +/- 0.2ºC Hum: +/- 3.5%	Temp: +/- 0.2ºC or +/- 0.3Cº Hum: +/- 3.5% or +/- 2%
	Light Sensors	S1087 by Hamamatsu	CdSe photocell	TLS2550, by TAOS	S1087, by Hamamatsu
Transceiver	*Module*	Xbee or Xbee PRO, both by Digi	Chipcon Wireless Transceiver	TI CC2420 802.15.4/ZigBee compliant radio	Chipcon Wireless Transceiver
	Frequency	2.4GH	868/916 MHz, 433 MHz or 315MHz	2.4Gh	2.4GHz
	Standard	IEEE.802.15.4/ Zigbee	IEEE.802.15.4/ ZigBee	IEEE.802.15.4/ ZigBee	IEEE.802.15.4/ ZigBee
	RF Power	0dBm to 18dBm (PRO)	-20 to 5dBm	-24 to 0dBm	0dBm
	Outdoor range[1]	100mt to 1200m (PRO)	150mt	75 to 100mt	50 to 125mt
	Current Draw (Tx)	35mA @ 0dBm[2]	27mA @5dBm	17,4mA @ 0dBm	17,4mA @ 0dBm
Processor	*MCU*	Atmel, Atmega 168	Atmel, ATmega128L.	Atmel, ATmega128	Texas Instruments MSP430 microcontroller
	Current Draw (in Active mode)	0.3mA @1MHz 1.9mA @4MHz 6.8mA @8MHz	5mA @4MHz 17mA @8MHz	5mA @4MHz 17mA @8MHz	2.4mA @ 8MHz
	Flash Memory	16Kb	128kb	128kb	48kb
	Programming access	ISP	Base station	Base station	USB
	Serial Communication	UART	UART	UART	UART
	Physical dimensions (mm) (Excluding battery pack)	(58x28x12)	(58x32x10)	(58x28x10)	(65x32x22)
Sensor Node Price		70€	352€	[3]	82€

[1] Considering an outdoor environment without obstacles.

[2] The consumption of the Xbee´s module regarding the RF transmission is certainly a little below 35mA, because this value is not totally spent with the RF transmission, but it also includes some processing activities.

[3] No information has been obtained so far on this feature, however as the sensing module of the Mica Z is superior than the Mica 2, the total cost of the module will be higher.

Even though the decrease of the battery level does not follow a linear function, we think that the battery level will remain stable for some more days. Further tests will continue to be carried out, nevertheless, the WISE-MUSE prototype is supposed to have at least 1 month of autonomy, although its batteries can last 2 months. These tests were performed on one node only, but in the future all the nodes should be tested.

Looking to the sensor nodes as a whole, they all have similar dimensions, and the WISE-MUSE mote presents a very low cost when compared to the Mica motes.

In conclusion, it is important to note that the module created was designed and built for the specific indoor environment monitoring at the Museum, presenting a number of advantages that may be attractive for this monitoring application, where its skills are within the requirements of the final client. Moreover, this new sensor node brings some advantages when compared to other commercially available solutions, such as low cost, small size, low power consumption, and higher transmission range.

V. Humidity control device

The humidity control device has been developed in the WISE-MUSE project to carry out the automatic control of dehumidifiers in order to regulate relative humidity levels inside the rooms where the WSN is implemented.

Figure 11 shows the developed device. It is essentially composed of one Xbee module, one relay, and one AC-DC converter, integrated in a box that allows the device to be plugged into an electrical outlet and at the same time to the dehumidifier.

When the humidity values reach an unacceptable limit (this limit were specified by the Museum), the base station sends a control message by radio frequency to the control device (more precisely to the XBee module, activating one of its pins). It receives the control message and analyzes it. As a consequence, it activates the relay. If humidity level is above the maximum limit then it turns on the dehumidifier. In opposite, if humidity level is below the minimum limit then it turns off the dehumidifier.

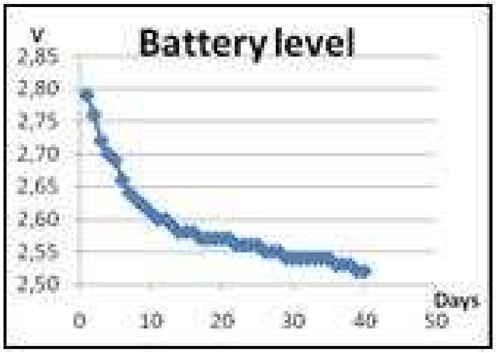


Figure 10. Average battery level

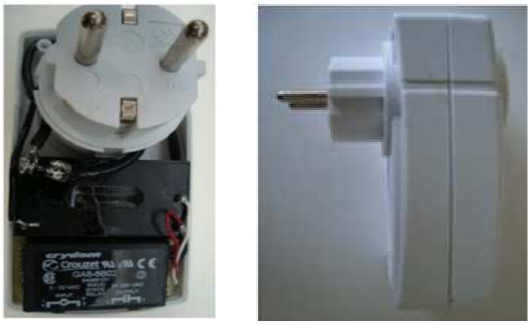


Figure 11. Humidity control device.

VI. Conclusion and Future Work

In this paper, the first experimental WSN for automatically and continuously monitoring the environment of a museum was presented. This solution was compared with related solutions, being outlined its advantages.

Besides being a simpler solution, the use of WSNs for environmental monitoring of a museum is, indeed, a more reliable solution. It is also less expensive than manual data collection or than a wired central monitoring system. Several problems were identified during the experiments, but they can be classified in two types. The problems related with the type of building where the WSN is being deployed, because it affects the signal propagation; and the problems related with the hardware characteristics and resource limitations of the sensor nodes, with the battery being the most limited resource. Equipping the nodes with a more efficient antenna and using a higher number of nodes will lead to the need of a lower transmission power, increasing the duration of the batteries. Also, changing the node’s program so that they perform less frequent measurements and transmissions will allow for high energy savings.

We also analyzed the graphics of collected data, what allowed us to understand the behaviour and the type of variations of temperature, humidity and light, throughout the day and night.

One of the main contributions of our work is the development of a new sensor node created for environmental monitoring applications, which is still a prototype since it is currently being tested. Nevertheless, even at this stage, we have already demonstrated that it brings some advantages when comparing to other commercially available solutions.

Furthermore, in order to increase the efficiency of this environmental monitoring system, we have implemented a system that automatically controls the dehumidifying devices, maintaining the humidity at more constant levels.

Regarding the visualization of data, data is available in different formats (tables, graphics, colour gradients, etc.), in a real-time basis. Additionally, an historic of collected data is kept for future consults in Microsoft excel and word format.

As for future work, we have several tasks to accomplish. After understanding the environment variations in Fortaleza São Tiago and after identifying all the existent problems, we have to carefully plan the deployment of a complete WSN that coverage all the exhibition rooms, archives and library. At the moment, only the Museum´s storage rooms are being monitored and controlled by the WISE-MUSE platform. We also have to extend the WSN to monitor some pollutants, in particular the carbon dioxide (CO_2).

We are developing a 3D awareness tool, based in the WSNSCW model, and that will be validated applying it to this specific application. So, data visualization will be enhanced by integrating it in a 3D representation of the museum, giving the user a much more realistic view of the network.

We also intend to carry out an analysis of the battery performance of other existing sensor nodes, in order to proof the effectiveness of the WISE-MUSE prototype.

Finally, we intend to connect the WSN to the air conditioning and dehumidifying systems in order to automatically control the temperature and the humidity of the exhibition and storage rooms.

ACKNOWLEDGMENT

I would like to express my gratitude to Bruno Gouveia, Filipe Santos and Roberto Fernandes who have collaborated in this project.

REFERENCES

[1] L. Brito, L. Rodríguez, F. Santos, and R. Fernandes, “Environmental Monitoring of Museums Based on Wireless Sensor Networks”, Proc. 4th International Conference on Wireless and Mobile Communications (ICWMC 2008), IEEE Computer Society Press, Jul/Aug 2008, Athens, Greece, pp. 364- 369.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A Survey on Sensor Networks”, IEEE Computer Networks, vol. 38, no. 4, pp. 393-422, March 2002.

[3] D. Gracanin, K. Adams, and M. Eltoweissy, “Data Replication in Collaborative Sensor Network Systems”, Proc. 25th IEEE International Performance, Computing, and Communications Conference (IPCCC 2006), April 2006, pp. 389-396.

[4] D. Wang, Q. Zhang, and J. Liu, The Self-Protection Problem in Wireless Sensor Networks, ACM Transactions on Sensor Networks (TOSN), vol. 3, no. 4, article no. 20, October 2007.

[5] S. Pai, P. Kurylosk, H. Yip, S. Yennamandra, S. Wicker, K. Boehner, and G. Gay, Networks of Sensors in Public Spaces: Combining Technology with Art, Proc. IEEE 21st International Conference on Advanced Information Networking and Applications (AINAW '07), vol. 2, May 2007, pp. 396-402.

[6] T. Onel, E. Onur, C. Ersoy, and H. Delic, Advances in Sensing with Security Applications: Wireless sensor networks for security: issues and challenges, Kluwer Academic Publishers Group, pp. 95-120, January 2006.

[7] J. Heidemman and N. Bulusu, Using Geospatial Information in Sensor Networks, Center for Embedded Network Sensing Papers. Paper 735, http://repositories.cdlib.org/cens/wps/735, September 2001.

[8] F. Oldewurtel and P. Mähönen, Neural Wireless Sensor Networks, International Conference on Systems and Networks Communications (ICSNC '06), October 2006, pp. 28.

[9] Omega, February 2008, http://www.omega.com

[10] 2DI, February 2008, http://e2di.com/catalog.html.

[11] Spinwave Systems, June 2009, http://www.spinwavesystems.com/

[12] A. Lee, C. Angeles, M. Talampas, L. Sison, and M. Soriano “MotesArt: Wireless Sensor Network for Monitoring Relative Humidity and Temperature in an Art Gallery”, IEEE International Conference on Networking, Sensing and Control (ICNSC 2008), April 2008, pp. 1263-1268.

[13] D. Del Curto and F. Raimondi, “WISPHER: cooperating WIreless Sensors for the Preservation of artistic HERitage, Case study: Arena di Verona, in Adjunct Proc. of Embedded WiSeNts (Project FP6-004400), Zurique, January 2006.

[14] Crossbow, June 2009, http://www.xbow.com/

[15] L. Liu, H. Ma, D. Tao, and D. Zhang, “A Hierarchical Cooperation Model for Sensor Networks Supported Cooperative Work”, Proc. 10th International Conference on Computer Supported Cooperative Work in Design (CSCWD’06), May 2006, pp. 1-6.

[16] L. Brito and L Peralta, “A model for Wireless Sensor Networks Supported Cooperative Work”, 3rd International Conference on Computer Graphics Theory and Applications (GRAPP 2008), Funchal, Madeira, Portugal, January 2008, pp. 505-511.

[17] M. Broxton, J. Lifton, and J. Paradiso, “Localizing a Sensor Network via Collaborative Processing of Global Stimuli”, Proc. 2nd European Workshop on Wireless Sensor Networks, Jan/Feb. 2005, pp. 321-332.

[18] A. Hu and S. Servetto, “Algorithmic Aspects of the Time Synchronization Problem in Large-Scale Sensor Networks", Mobile Networks and Applications, vol. 10, 2005 Springer Science + Business Media Inc., 2005, pp. 491-503.

[19] ZigBee Alliance, 2004, www.zigbee.org

[20] Digi, May 2009, http://www.digi.com/products/wireless/zigbee-mesh/

[21] Sensirion, May 2009, http://www.sensirion.com/

[22] Hamamatsu, April 2009, http://jp.hamamatsu.com/en/product_info/index.html

[23] Sentilla (formerly, Moteiv Corporation), April 2009, http://www.sentilla.com/

A Wireless IP Multisensor Deployment

Diana Bri, Hugo Coll, Miguel Garcia, Jaime Lloret
Integrated Management Coastal Research Institute, Universidad Politécnica de Valencia
Camino Vera, s/n, 46022, Valencia
diabrmo@posgrado.upv.es, hucolfer@posgrado.upv.es, migarpi@posgrado.upv.es, jlloret@dcom.upv.es

***Abstract*— Every sensor node in a Wireless Sensor Network has a microcontroller, a transmitter/receiver and a sensor. It is able to acquire data from specific point in a real environment and transmit it through the Wireless sensor Network. Sometimes it is useful to gather different types of data from the same place in order to obtain a final result. In the related literature, very few works are about sensing different parameters using a unique sensor. In this paper we present a Wireless IP multisensor that is able to gather several types of data from the environment and transmit the result of their combination. Our decision has being mainly based on its development costs, its expansion capacity, the possibilities provided by the operating system, and its flexibility to add more features to the sensor node. We will show all the characteristics of our proposal and the hardware needed for its expansion. Then we will discuss its main application areas. A comparison with many wireless IP sensors offered in the market is also provided.**

Keywords – Wireless multisensor, environment monitoring, Wireless Sensor Deployment.

I. Introduction

A sensor is any kind of transducer that transforms the magnitude we want to measure in another which is easier to measure. There can be direct indication sensors (e.g., a mercury thermometer) or there can be connected to an indicator (possibly by means of an A/D converter). The measured values have to be read by a human.

Nowadays, a sensor is considered as a basic electronic device for our lives. It covers a wide range of possibilities and applications and, furthermore, it can be used to sense and monitor different parameters according to our necessities. Several aspects can be related with the sensor nodes: a simple physical sensor or a multiple physical sensor, and a wired network of sensors or a wireless network of sensors. We should distinguish between multiple sensors, i.e. many sensor nodes, and a multisensor node, that is able to sense several magnitudes.

Sometimes, it is necessary to control several parameters simultaneously in the same place. The main aim is to have greater control on specific application in order to do these measures more efficient. Therefore, we can use the combination of several physical sensors, forming a multisensor [1]. An obvious example would be the combination of responses from different physical sensors, for determining the trajectory of a wildfire. This combination could be the wind direction, a large and quick increment of the temperature, increment of the CO_2 concentration and a very low value of relative humidity. The right process of this information will report that a fire is happening and what is its direction. This information will allow the firefighters to provide evacuation plans and think on efficient strategies to combat the fire, with the goal of minimizing the loss of forest mass and materials (houses and facilities). If only one sensor was used, the system would not be able to provide whether there is fire, or if it is just a very hot day, or a person who is smoking near the sensor. There are a lot of fields where the use of multisensor can be benefit, such as, home control, building automation, medical applications or robots, among others. It will be later discussed in this paper.

Before we get to combine sensors, we should know how they can be classified. On one hand, many different types of magnitudes can be measured and, on the other hand, there are many types of sensors to measure any physic magnitude. In order to do a study it is needed to classify them according to some criterion. The following classification of sensors has been done considering each sensor like an isolated device which function is to detect sign of qualities or physic phenomenon according to the kind of sensor, and they are converted to useful signals for a measurement or control system:

- According to the energy contribution, sensors can be divided into modulators and generators. In modulators or active sensors, the energy of the output signal almost always comes from an auxiliary energy. However, the output energy in generator or passive sensors is provided by the input. Generally, active sensor circuits require more connexions than passive sensors in order to supply energy. On the other hand, modulator or active sensors' sensibility can be modified with the supplying signal (it can not be modified in generator or passive sensors).

- According to the output signal. They can be analog or digital sensors. The output of the analog sensors is continuous. The information is in the amplitude of the signal. The output of the digital sensors is discrete. Discrete signals are more reliable.
- According to the operation mode. They can be defection or comparison sensors. In defection sensors, the measured magnitude produces a physical effect that causes a similar effect, but opposite, in another part of the instrument, and related with some useful variable. A comparison sensor tries to keep the defection null by means of the application of a well-known effect and it is opposite to the effect generated by the magnitude to measure. Usually, measurements given by comparison are more precise because the opposite known effect is calibrated with a quality's reference magnitude.
- According to the input/output relationship. They can be classified in zero-order, first-order, and second-order or superior order. The order is related with number of independent storage elements of energy. This classification is very important when a sensor is used in a loop control system [2].
- According to the input signal [3]:

  a) *Mechanical:* longitude, area, volume, mass, flux, force, pressure, speed, acceleration, wavelength, position or acoustic intensity.

  b) *Thermal:* temperature, heat, entropy or flux of heat.

  c) *Electric*: voltage, current, resistance, inductance, capacity, load, electric field, frequency, dielectric constant or bipolar moment.

  d) *Magnetic:* intensity of field, density of flux, magnetic moment or permeability.

  e) *Radiation*: intensity, wavelength, polarization, phase, reflectance, transmittance or refraction index.

  f) *Chemical:* concentration, potential redox or PH. Together with electronic sensors, chemical sensors are the most important sensors due to their application fields. These sensors are being used successfully in environments, medicine and industrial processes [4].

In table I, a classification according to three criterions and some examples for each one of them is shown.

However, there are some cases where it is needed to gather several types of measurements, by only one device, in order to obtain more reliability or to obtain specific results. To achieve this goal, it is needed a multisensor.

TABLE I. SENSORS CLASSIFICATION

Criterions	***Types***	***Examples***
Energy contribution	Modulators Generators	Delta-Sigma Thermocouple
Output Signal	Analog Digital	Potentiometer Position coder
Mode of operation	defection comparison	Defection accelerometer Servo-accelerometer

The remainder of this paper is as follows. Section 2 gives all the works we have found related with multisensors. A multisensor is presented in section 3. We will give its main hardware and software features, its implementation, how it can be extended and the operative system used by the device. Section 4 compares the wireless device of the selected hardware with other wireless devices in the market. Some performance test results are shown in Section 5. Section 6 discusses the main application environments where it can be applied. In section 7, we have compared our proposal with many other developed sensor nodes that can be found in the related literature. Section 8 gives the conclusion and future works.

II. RELATED WORKS

In the literature, we can find some works related with multisensors, but almost all of them have been deployed for robots [5] [6].

In 1989, Ren C. Luo and Michael G. Kay [7] introduced the interest and the importance of developing multisensors to increase the capabilities of intelligent systems. The paper describes the multisensor integration and fusion and gives some approaches to the problem. It shows some methods for integrating and fusing multisensory information, to existing multisensor systems used in different areas of application. In 2002, Ren C. Luo et al. extended this information in [8]. They provided an overview of the sensor technologies and described the paradigm of multisensor fusion and integration as well as fusion techniques at different fusion levels. Applications of multisensor fusion in robotics, biomedical system, equipment monitoring, remote sensing, and transportation system were also discussed.

David L. Hall and James Llinas presented a theoretical introduction to multisensor data fusion in [9]. They provided a tutorial on data fusion, introducing data fusion applications, process models, and identification of applicable techniques. Their aim was to show how to fussion sensor measures to obtain a result. They also show a flow chart to explain different manners to interconnect multiple sensors in a single device.

A multisensor example can be found in reference [10]. J. D. Cullen et al. presented a multisensor for online monitoring of the spot welding in automotive industry. The multisensor was able to measure the current, the voltage and the welding force.

Another multisensor example was presented in reference [11]. It was a novel design of the light-addressable potentiometric sensor (LAPS) for realisation of a portable multisensor device. Light sources and electronics including an oscillator, a multiplexer, a pre-amplifier and a high-pass filter were encapsulated in a pen-shaped case, on which the sensor plate was mounted. This sensor device is capable of measuring up to four different ion species by integrating different ion-selective materials on the sensing surface, each illuminated with an independent light source.

E. Kuljanic et al. [12] presented a multisensor approach for chatter detection in milling and compared a single-sensor systems and multisensor systems in terms of accuracy and robustness against malfunctions in order to demonstrate that machines with more sensors give better results for the system. The signal characteristics both in time and frequency domain were condensed into a set of chatter indicators, which were further elaborated by means of statistical basic concepts.

On the other hand, we can find other works where their authors developed a multisensor kernel system, in order to manage the multisensor such as the one presented by T. C. Henderson et al in [13].

No one of the works shown presents a multisensor node that is able to have configured an IP address and join an IP wireless sensor network.

III. Multisensor Proposal

In this section we are going to present a sensor node that is able to sense several parameters from the same place while it is able to form a Wireless IP network of multisensors. The IP protocol is one of the most widely used network interconnection protocol. IP provides a way to transport datagrams from any source to a destination, regardless of whether these machines are in the same network. Moreover, in case of being other networks between the sensors, the IP protocol allows to access them remotely.

In order to achieve our aim, we looked for a device with a control unit. This control unit will manage and control all sensors connected to the device. On the other hand, the electronic circuit must have several input interfaces in order to connect several physical sensors. One of the main aspects taken into account, in order to decide which the best election was, was the circuit costs, its expansion capacity, the possibilities provided by the operative system, and its flexibility to add more features to the sensor node.

Our proposal is based on the use of the Linksys WRT54GL router, from Cisco Systems inc., as the core controller [14]. It is an embedded system that has two serial interfaces in its board, so it meets our pre-requisites. First of all, let’s see the router’s hardware and software main features [15].

A. *Linksys WRT54GL hardware features*

This sub-section shows the main features of the Linksys WRT54G version 4.0.

The core of this router is a Broadcom BCM5352E processor, working at 200 Mhz. It is a Microprocessor without Interlocked Pipeline Stages (MIPS), which is a RISC microprocessor architecture. It has 256 Bytes prefetch cache, 4 MB INTEL TE28F320 flash memory and 16 MB RAM at 100 MHz clock rate. All the system has a 12 Volts DC power supply.

In order to communicate with other devices, and interconnect them, WRT54GL is equipped with 4 port full-duplex 10/100base TX switch to connect wired Ethernet devices, and a wireless access point, which lets connect both Wireless-G (IEEE 802.11g at 54Mbps) and Wireless-B (802.11b at 11Mbps) devices to the network. The transmitting power for wireless connection is up to 18 dBm. At last, 1 full-duplex 10/100base TX WAN port allows connecting the network to the WAN.

In addition to the mentioned ports, Linksys WRT54GL offers internally GPIO (General Purpose Input/Output), UART (Universal Asynchronous Receiver-Transmitter) and JTAG (Joint Test Action Group) ports. Some extensions can be made to the router by using these ports. Fig. 1 shows the embedded board and its hardware distribution. The UART is signalled as JP2 and the JTAG port is signalled as JP1.

B. *Linksys WRT54GL software features*

One of the main features that have caused the use of the Linksys WRT54GL as a sensor node was the possibility of installing a Linux Kernel 2.4. On one hand, it is a well known operative system, so we didn’t need to learn how to work with a new operating system and, on the other hand, we know all the possibilities that a Linux is able to provide us. So, at software level, this model is based on open source software, causing the development of different specific software applications for it and expanding the factory default capabilities. Several of its main software features that can be mounted over this platform are:

- Web server.
- DHCP Client and DHCP Server.

- Telnet server and client.
- SSH server and client.
- RADIUS server.
- WPA/TKIP and AES encryption.
- WPA2 encryption.
- Wireless MAC address filtering.
- Powerful SPI firewall.
- VPN client and server.
- VLAN.
- VoIP switchboard.
- QoS Bandwidth Management.
- Wake On LAN.
- MMC/SD Card Support.
- USB port support.

C. *Implementation*

In order to connect two sensors directly to the board, we made an extension using the GPIO of the Linksys WRT54GL router. It provided us two serial ports through the JP1 port.

Fig. 2 shows a serial port interface connected on board by welding pins on JP1. Then, we added two DB9 Female DCE ports because we wanted flexibility in order to change the type of sensor connected to our device.

Fig. 3 shows the integrated circuit used to provide two serial ports. A RS-232 line converter is needed to go from +3.3V to +5V. In our design we propose this implementation with a MAXIM MAX233 integrated circuit. It contains four sections: dual charge-pump DC-DC voltage converters, RS-232 drivers, RS-232 receivers, and receiver and transmitter enable control inputs

On the other hand, we connected a SD/MMC card reader to the GPIO pins of the CPU found inside the Linksys and with the help of a little driver we can use as a block device from Linux. Compiling the kernel for the Linksys with e.g. support for MSDOS partitions and VFAT we are able to mount, read, write, partition and so on in normal SD and MMC cards. The speed for reading and writing is about 200 KB/s.

The SD/MMC card reader has been tested for a 1GB SD and MMC cards. It allows us to install applications for signal processing and store and manage acquired data from both sensors. Now, the sensor node is able to store and process the data stored without the need of sending the measures from its sensors continuously. The system is now able to gather data, process it internally and send only alarms or statistical data spontaneously to the network, thus saving energy.

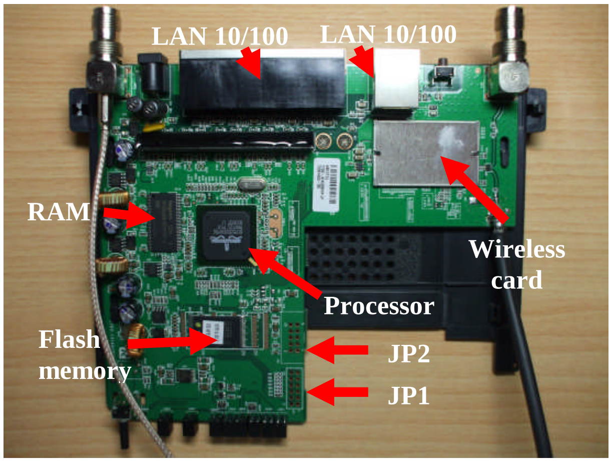


Figure 1. Hardware distribution on board

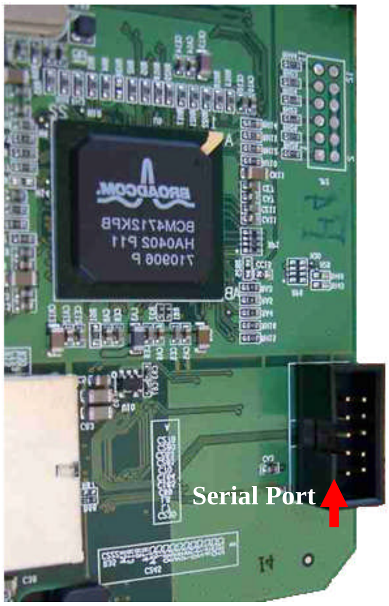


Figure 2. Serial ports connected on board

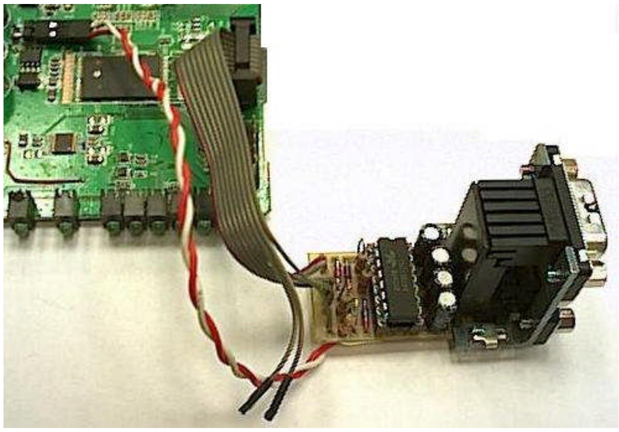


Figure 3. Integrated Circuit used to provide two serial ports.

D. *Proposal extension*

The main drawback of our proposal was that Linksys WRT54GL router only allows two physical sensors because it has only two available serial ports. This limitation would limit us the number of application environments. So, we tried to extend this feature by adding an external hardware. One of them is the use of a JTAG to parallel port interface, and the other is given by the Arduino open hardware platform [16].

There are several different kinds of interfaces for hooking up to the JTAG headers (signalled as JP1 in the Fig. 1). We are going to talk about JTAG to Parallel port interface. It adds a parallel port to the WRT54GL providing another interface to connect more sensors. In the design stage we have two different variants of interfaces, the unbuffered and the buffered one. The first way is the unbuffered one, this is the cheapest and the easiest way to add a parallel port to the WRT54GL, only with 100 ohm resistors connected directly to the specified JTAG pins. The other variant is the buffered one. It is more complex that the unbuffered interface but it is more immune to noise and static. It provides a higher data transfer rate than the unbuffered one. In our proposal we use a PHILLIPS 74HC244 buffer to implement it. The 74HC244 is an octal buffer/line driver, 3-state. The JTAG to parallel port interface provides slow data rates. This is due to the nature of the parallel port, but, these data rates are really enough to the multisensor of our proposal.

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for anyone interested in creating interactive objects or environments. The board is available to buy, but schematics are free distributed in order to let anyone mount its own board. Controlling software can be downloaded for free from the Arduino website. Fig. 4 shows the Arduino board. It is a cheap, robust input and output board based on the ATmega168. It has 13 digital pins (6 of which allow PWM output) and 6 analog inputs. There are many different versions that allow adding USB ports and/or serial ports, Bluetooth connexions and so on.

Arduino can sense the environment by receiving input from a great variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. Arduino hardware can be stand-alone or can communicate with software on running on a computer. So, according to it, we connected the Arduino board to the WRT54GL by using the serial port on the router board. Now, we have deployed a real multisensor system dotted with multiple in and out ports to connect sensors and any kind of devices. The core controller, placed in the Linksys WRT54GL router, is strong enough to compute all the data received from the Arduino board.

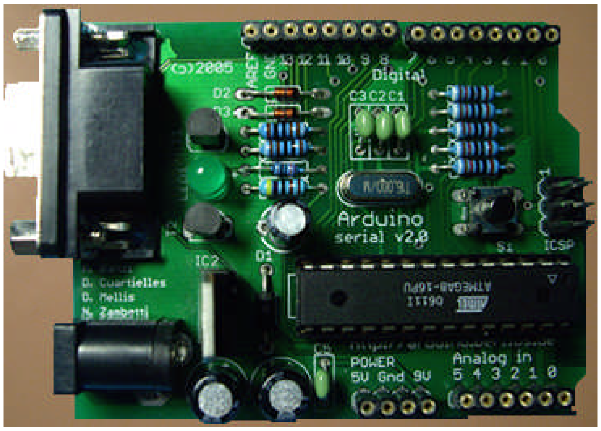


Figure 4. Arduino serial board

Due to the very small size of the Arduino board, it can be allocated inside the WRT54GL cage.

E. *Operating system and communications*

There are several Linux-based distributions built for the Linksys WRT54GL router, with a high variety of functionalities. The main distributions are Wrt54g-linux [17], BatBox [18], OpenWRT [19] and DD-WRT [20]. Wrt54g-linux and BatBox are loaded on RAM, so they have very light and have limited functionalities. Moreover, they are not stored in the router after it has been reset. Those are the reasons why both operative systems have not been considered to be used in our proposal.

We have chosen DD-WRT for our implementation due to the high quantity of features included and the open source community support. In order to change the operative system, first of all we have to erase the flash memory. Then, we can install DD-WRT operating system. This operation can be done by Web GUI, TFTP or just typing directly in command line. The final result is the same, so there is no matter the way chosen for it. After the operative system is changed, we have a Linux inside the device and we can work with it as with a regular Linux.

Once the operating system is installed, the following step (at software level) is to communicate the Arduino board with the sensors connected to its ports. The Arduino programming language is based on C/C++, so data can be sent to the sensors or received from them by using serialWrite() and serialRead() functions (when devices are connected to the serial ports). Also speed can be modified by typing begin Serial (speed) [21]. Depending on the device connected to the board and the kind of port used to communicate, the C/C++ code of the communication script could be a bit different. The procedure will be to change references to pins involved in the transmission and reception and the specific commands for each port connection.

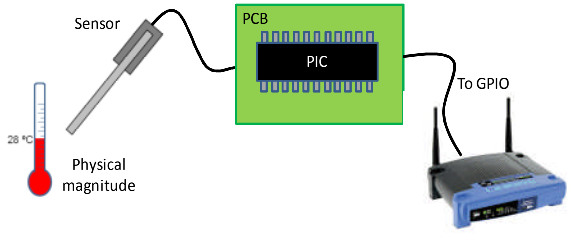


Figure 5. Block diagram

Keeping in mind that we can install an open source Operative System and the GPIO connections which can be found on the Linksys WRT54GL, we can program a script that allows us to take data from the sensors to be processed by our multisensor device. To this end, we design a small electronic circuit that works as an interface between the sensor and multisensor device, in order to adapt the sensor signal to the type of signals that GPIO terminals support. Fig. 5 shows a block diagram with the proposed system. This proposal allows connecting other sensors with a higher bit rate. The sensors, whose interface is based on RS-232 standard, can have bit rates up to 128.000 bit/s. SO, it would enable us to reach speeds up to 200 KBytes/s, namely, the read and write speed of a SD card.

IV. Wireless interface comparison

We can find several wireless interfaces in the market that incorporate the IEEE 802.11 standard and can be used for a multisensor node. They are low energy consumption solutions. At the same time, these devices integrate great variety of resources and functions. Their sizes are quite small; they can be built-in inside portable devices. Because their low power consumption, they can also be fed by batteries. All these integrated circuits are also designed to enable the communication between wireless devices and a serial interface, so a sensor can be connected to it. In this sub-section we will analyze only the chipsets that have their power consumption specifications published. There are other IEEE 802.11 compliant chipsets in the market but they have not embedded serial interfaces. Table II shows some of these devices. n/a means that this information is not provided by the manufacturer.

Most of these devices (those whose price is below $ 300) need a development kit in order to use the device. The price of these kits could be $ 150-200 and can reach to $ 1,480 in some cases. The sizes of these evaluation kits are roughly similar in size to Linksys WRT54GL. We choose the Linksys, because this device offers us the same features than the others, but it is cheapest and allows us to add a Linux-based Operative System based to program and run applications in order to process information gathered by the sensors. Furthermore, this device has the faster microprocessor of the comparison table.

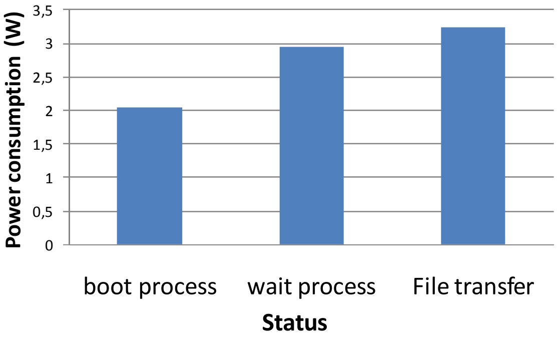


Figure 6. Energy consumption

V. Performance Tests

In this section we can see the tests made with the device. There have been several testbenches.

In the first one we measured the energy consumption in each of the operating modes of the IP multisensor. The second test measured the typical parameters of the network. The tests were performed under the 802.11 b technology.

A. Energy consumption

In order to perform the power consumption tests, a power supply set to 12V (DC) and limited to 1000mA was used. It has also been necessary to use a wattmeter to measure the power consumed in each case.

First, we measured the energy consumed in the boot process of the device. This process takes approximately 15 seconds. Then, we measured the energy obtained when the device is transmitting and receiving information and, finally, the energy when the device is waiting to receive data. Fig. 6 shows the results.

One of the major requirements in WSN is to have very low energy consumption. It is known that IEEE 802.11 protocols are inadequate for energy constrained devices.

Therefore, the operation and the utilities of the different hardware parts of the Linksys WRT54GL have been studied. We have determined that there are some parts that could be unused in our sensor implementation, getting so, that these components will not consume energy if we remove them. For example, the WAN port and 3 of the 4 LAN ports can be removed and with them, all of their passive components that consume energy (such as resistors and diodes). Moreover, the integrated circuits which are not used for sending information via wireless can be also removed. With these modifications, we would reduce the global system consumption of about 15%.

TABLE II. SENSORS NODES COMPARISON

Device	Manufacturer	Current consumption in TX	Current consumption in Rx	Current consumption in Standby	Core CPU	Operating voltage	IEEE Especification	Size in (mm x mm)	Price in $
Nano WiReach [22]	Connect One Ltd	250 mA	190 mA	8 mA	32-bit RISC ARM7TDMI, low-leakage, 0.13 micron, running at 48MHz	3.3 V	802.11 b 802.11 g	17 x 33	545.75
Mini Socket iWiFi [23]	Connect One Ltd	250 mA	190 mA	8 mA	32-bit RISC ARM7TDMI, low-leakage, 0.13 micron, running at 48MHz	3.3 V	802.11 b 802.11 g	41 x 31	545.75
S103 WLAN Compact Serial Module [24]	Rf solutions	325 mA	210 mA	80 mA	n/a	5 V	802.11 b	40 x 60	75
"WiFly" 802.11B Module (RN-111B) [25]	Roving networks	110 mA	40 mA	35 mA	n/a	3.3 V	802.11 b	29 x 50	69
"WiFly GSX" 802.11G Module (RN-131G) [26]	Roving networks	210 mA	40 mA	15 mA	n/a	3.3 V	802.11 b 802.11 g	38 x 26	45
WiFly GSX Super Module "SuRF Board" (RN-134) [27]	Roving networks	212 mA	50 mA	15 mA	n/a	3.3 V	802.11 b 802.11 g	17 x 33	99
MatchPort b/g [28]	Lantronix	360 mA	225 mA	76 mA	Lantronix DSTni-EX x86 CPU, on-chip 256 KB zero wait static SRAM, 2,048 KB Flash, 16 KB Boot ROM, 8 GPIO	3.3 V	802.11 b 802.11 g	29 x 50	86.46
WiPort [29]	Lantronix	650 mA	395 mA	91 mA	Lantronix DSTni-EX 186 CPU, on-chip 256 KB zero wait static SRAM, 2,048 KB Flash, 16 KB Boot ROM,	3.3 V	802.11 b 802.11 g	27 x 51	139.66
MatchPort® b/g Pro [30]	Lantronix	350 mA	260 mA	160 mA	Lantronix 32-bit processor, 166Mhz (159 MIPS - Dhrystone 2.1)	3.3 V	802.11 b 802.11 g	45 x 45	210
RCM4400W RabbitCore [31]	RABBIT	450 mA	450 mA	80 mA	Rabbit 4000 Microprocessors (60 MHz)	3.3 V	802.11 b	13 x 14	131.38
Connect Wi-ME [32]	DigiBoard	346 mA	186 mA	34 mA	32-bit, 55 MHz NS7520 processor ARM7TDMI ARM core	3.3 V	802.11 b	80 x 110	410.41
ConnectCore Wi-9C [33]	DigiBoard	900 mA	700 mA	10 mA	32-bit, 155 MHz NS9360 processor ARM926EJ-S ARM core	3.3 V	802.11 b 802.11 g	91 x78	379
WRT45GL [14]	Linksys by Cisco	270mA	270mA	245mA (regular mode)	Broadcom BCM5352E processor, (200 Mhz). 16 MB RAM	12V	802.11 b 802.11 g	186 x 154	69.99

B. Measures of the wireless network parameters

In order to show the network parameters we measured the worst case, that is, when the sensor node gathers the data from the physical sensor, process it, and sends it to another device using the wireless network. The data collected by the physical sensor could be very different. It can be a temperature value which packet size will not reach more than a few bytes. But the physical sensor could be a video images sensor, which can reach sizes of several MBytes.

We used two identical wireless IP sensors to perform our test bench. They had the same hardware and software described in the previous sections. In order to see what happens in the worst case, we stored a large file in a single sensor which contains different types of data obtained from each physical sensor connected to the multisensor.

Thus, we transmitted a large file from sensor 1 to sensor 2, under the 802.11b technology. The file size was 210 MBytes and the transfer time was approximately 15 minutes.

We measured several parameters in order to know the network performance in such conditions. These parameters were the number of errors, drops, collisions, broadcast traffic, octets, packets and utilization (%). In the first three parameters we obtained a value of zero, because in this test, we only had two sensor nodes one transmitting and one receiving, so there were not errors, drops, collisions during our experiment.

Fig. 7 shows the broadcast packets sent in the experiment. There has not been more than one packet per second. There have been only 7 broadcasts during the 15 minutes (an average value of 0.009).

Fig.8 shows the number of packets per second along the time. An average value of 192 packets per second has been obtained. The maximum number of packets per second was 340.

Fig. 9 shows the number of bytes per second that have been sent over the network. There has been a maximum value of 336575 Bytes per second (in a theoretical channel of 11Mbps). The mean value during the 15 minutes has been 264680.13 Bytes.

Fig. 10 shows the utilization se of the bandwidth of the channel in %. It has been estimated that the system has 95% of channel utilization. The 77.42% of the time the wireless channel had 100% of utilization.

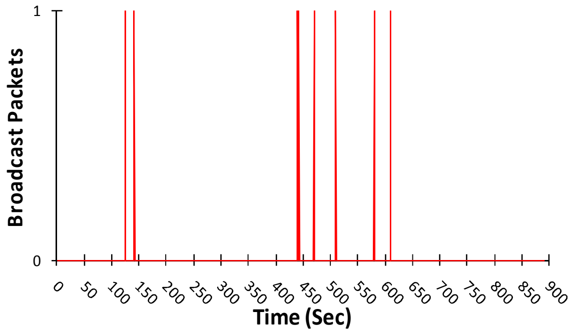


Figure 7. Broadcast packets

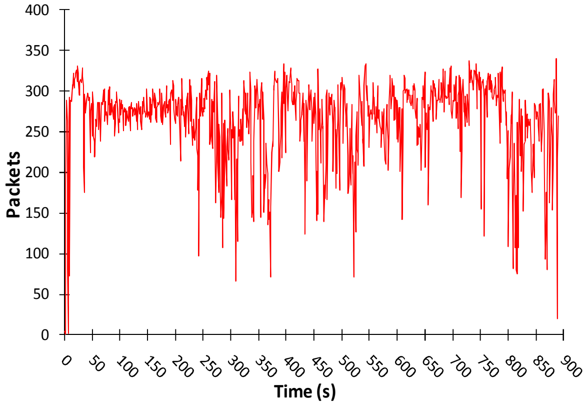


Figure 8. Packets per second.

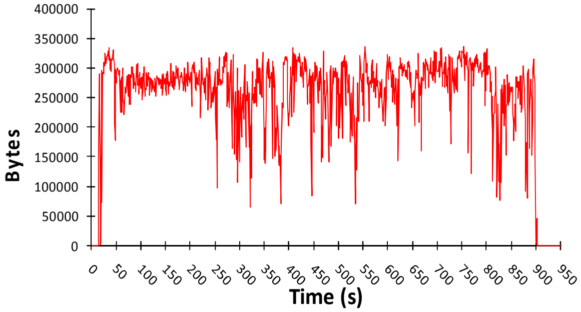


Figure 9. Bytes per second.

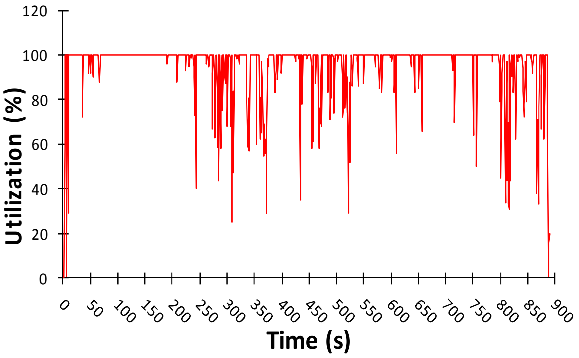


Figure 10. Utilization in %.

VI. Application environments

The presented model can be applied to any environment that needs to be sensed by several types of variables. It is also able to process internally the measures taken from the connected sensors and send the processed information to a remote site. It is flexible and it could be adapted to any type of environment and to any type of sensor with a serial output. The change of the programming code is only needed to adapt the control management to different sensors.

There are more needs multisensor nodes than it is expected. The following list gives some examples:

- *Home Control:* There are several possible home applications where a multisensor node can be used. E.g. in automate control of multiple home systems to improve conservation, convenience, and safety. A multisensor node on the wireless sensor network can check some parameters (temperature, light, humidity, etc) continually in order to make flexible management of lighting, heating, and cooling systems from anywhere in the home. Another home application could be a sensing application that captures highly detailed electric, water, and gas utility usage data. With these types of applications one user could control his/her home expenses easily without the need of place many sensors inside the house.
- *Building Automation:* A multisensor node can be used for multiples purposes in building control. A multisensor could measure some parameters simultaneously as light, switches on, etc. in order to enable the rapid reconfiguring of lighting systems to create adaptable workspaces. The combination of the parameters measured could give better decisions. Also, the wireless sensor network, with multisensors, could be used for building energy monitoring and control. They could improve living conditions for the building's occupants, resulting in improved thermal comfort, improved air quality, health, safety, and productivity.
- *Industrial automation:* Industrial automation applications provide control, conservation, efficiency, and safety. E.g. a multisensor can provide the sound level and the temperature of machines for monitoring the proper functioning of a factory's area. The multisensor can take decisions, when both measures exceed the threshold, before sending an alert to the central site. Moreover the sensing applications using multisensor devices reduce energy costs through optimized manufacturing processes.
- *Medical applications:* A number of hospitals and medical centres are exploring applications of wireless sensor networks technology to a wide range of medical applications, including pre-hospital and in-hospital emergency care, disaster response, and stroke patient rehabilitation. These systems can use multisensor devices to collect several data from the patients simultaneously, such as body temperature, heart rate, etc. This information will be sent to a local central server and to the hospital.
- *Robots:* This is the most common application environment. Robots need to combine multiple sensed measurements in order to take a decision (to walk, to avoid an obstacle, to take something from the floor, and so on). One of the major issues in a mobile robot acting as a gateway is the communication between the robot and the sensor network. Using our multisensor node this issue will be easily arranged, especially if the robot uses serial communication, because this node can manage several serial ports.
- *Habitat monitoring:* Sensor networks represent a significant advance over traditional invasive methods of monitoring. Sensors can be deployed prior to the onset of the breeding season or other sensitive period (in the case of animals) or while plants are dormant or the ground is frozen (in the case of botanical studies). Using a multisensor node we can obtain several parameters in a single device. A multisensor will allow building parallel specialized networks [34]. Thus, we have a system that collects lots of data without providing high visual impact (an important factor in environmental settings).
- *Fire detection and monitoring [35]:* In order to obtain a true fire alarm, it is better to develop a multisensor which combines several sensor measurements such as temperature variation, the humidity, wind direction and CO2 level. The multisensor can check the history of the sensor measurements and process the data in order to give statistical results and combine the sensor values thresholds. The result will be more precise than a single sensor. Currently, there are many critical areas that may be at risk of fire. These multisensor systems could be used to control the zones, so we have a simple system based on wireless sensor networks to monitor fires.
- *Traffic monitoring:* Traffic flow is a sector that is expected to benefit from increased monitoring and surveillance. These systems use multisensor nodes to collect data of many parameters. The systems are installed along major highways; the digital sensor network gathers lane-by-lane data on travel speeds, lane occupancy, and vehicle counts. These basic data elements make possible to calculate average speeds and travel times.

- *Military applications:* For military users, an application focused on wireless sensor networks technology has been area and theater monitoring. Wireless Sensor Networks can replace single high-cost sensor assets with large arrays of distributed sensors for both security and surveillance applications. The Wireless Sensor nodes are smaller and more capable than sensor assets presently in the inventory; the added feature of robust, self-organizing networking makes Wireless Sensor Networks deployable by untrained troops in essentially any situation. A multisensor could improve the system being able to sense environment parameters while detecting the presence of other humans. It can also inform about the combination of the parameters sensed.
- *Indoor people location and behaviour:* In this case, for example, the multisensor could be used in order to know the walking direction of a person in a corridor. It could be easily known if the wind direction is combined with a presence detection sensor. It will allow us to know if a person is coming in or leaving a zone. If we place just one of those sensors it will not give us the right result, because it could be open windows and the wind could cheat us. This system can be also used for intrusion detection, as a visitor counter or to be sure that all people have left a building.
- *To measure the optimum water of a pool in a spa:* There has to be a sensor able to measure the optimum measures of the temperature, the chrome, the water pH and the aroma. In large pools the nodes of the wireless sensor network could be able to sense multiple parameters simultaneously. It will let us have a global view of the state of the pool.
- *Automobile applications:* A modern automobile has about 8 Km of cables to connect hundreds of sensors. A Wireless Sensor Network allows not only reducing the volume and weight required by the cabling, but also the deployment of sensors with more freedom. A multisensor would be able to measure different parameters simultaneously in an automobile in order to have a distributed network.

There are many more specific applications, but in this section we have discussed the most important areas where they can be used.

VII. Wireless Sensor Nodes Comparison

Nowadays, many sensor nodes can be bought in the market. In this section, we are going to compare our proposal with the most well known sensor nodes.

Table 2 shows a summary with the name of the sensor, its microcontroller, the type of transceiver or wireless technology, the internal and external memory, and its operative system. We have provided only the public features of the devices, so, some hardware characteristics are not shown because the owner does not provide them. We can observe that the multisensor node is the sensor node with more internal and external memory. Each node has an operating system, TinyOS is the most common, but the operating system used in our sensor node can be modified to be as efficient as TinyOS. Another important feature of all sensor nodes is the use of UNIX-based operating systems.

On the other hand we can see that the sensor node with worst features is the Dot node. This node only has 1 KB of RAM and up to 16 KB of external memory.

If we look the microcontroller feature of the sensor nodes described in Table 3, we can see that there are nodes that have processors working with around 400 MHz. The worst processors have a speed around 10 MHz. In our case, the multisensor presented in this work uses a processor with a speed of 200 MHz. The device has not the most powerful microcontroller, but it has enough power for its proper operation.

Finally, we have to take into account that our proposal is the only one that is able to have 2 physical sensors directly connected at least, but we are supposing that the other deployments are able to be changed to acquire data from 2 or more physical sensors (making some adjustments in the electronic board). We can observe that our proposal has the most powerful processor and the one with most internal and external memory (we can even add 1 GB in a SD card). On the other hand, Linux is the most well known operative system in that table. But as drawbacks, our proposal works with IEEE 802.11 b/g (although we can build our proposal over an IEEE 802.11n device, if it has the same features) and the energy consumption is higher than the others because of its hardware requirements.

VIII. Conclusions

We have presented a multisensor for Wireless Sensor Networks using IEEE 802.11b/g. It is flexible and it could be adapted to any type of environment and to any type of physical sensor with a serial output. Using the described system with the appropriate Arduino board and code programming for each specific application, we can create a high–complex multisensor system which can be able to sense many physical magnitudes. In future works we will add to our multisensor a JTAG to USB interface to support sensors with the requirement of high bit rates. This is possible to the nature of the JTAG interface that can provide a theoretical data rate of 25 Mbps. We are now studying how to make the Linksys WRT54GL router consume less energy.

TABLE III. SENSOR NODES COMPARISON

Sensor Node	Microcontroller	Wireless Transceiver or Wireless Technology	Internal Memory	External Memory	Operative System
Our proposal	Broadcom BCM5352E at 200 Mhz processor	IEEE 802.11g and IEEE 802.11b	16 MB RAM	4 MB Flash	Linux
BEAN [36]	MSP430F169	CC1000 (300-1000 MHz) with 78.6 kbit/s		4 Mbit	YATOS
BTnode [37]	Atmel ATmega 128L (8 MHz @ 8 MIPS)	Chipcon CC1000 (433-915 MHz) and Bluetooth (2.4 GHz)	64+180 K RAM	128K FLASH ROM, 4K EEPROM	BTnut and TinyOS
Dot [38]	ATMEGA163	ChipCon CC1000 916MHz	1K RAM	8-16K Flash	TinyOS
EPIC Mote [39]	Texas Instruments MSP430 microcontroller	250 kbit/s 2.4 GHz IEEE 802.15.4 Chipcon Wireless Transceiver	10k RAM	48k Flash	TinyOS
Eyes [40]	MSP430F149	TR1001		8 Mbit	PeerOS
EyesIFX v1 [40]	MSP430F149	TDA5250 (868 MHz) FSK		8 Mbit	TinyOS Support
EyesIFX v2 [40]	MSP430F149	TDA5250 (868 MHz) FSK		8 Mbit	TinyOS Support
FlatMesh FM2 [41]	16MHz	802.15.4-compliant		660 sensor readings	Commercial system, for digital sensors
Firefly [42]	Atmel ATmega 1281	Chipcon CC2420	8K RAM	128K FLASH ROM, 4K EEPROM	Nano-RK RTOS Support
GW node [43]	PIC18LF8722	BiM (173 MHz) FSK	64k RAM	128k flash	Custom OS
IMote 1.0 [44]	ARM 7TDMI 12-48 MHz	Bluetooth	64K SRAM	512K Flash	TinyOS
Imote 2.0 [44]	Marvell PXA271 ARM 11-400 MHz	TI CC2420 802.15.4/ZigBee compliant radio	32MB SRAM	32MB Flash	Microsoft .NET Micro, Linux, TinyOS Support
Iris Mote [45]	ATmega 128	Atmel AT86RF230 802.15.4/ZigBee compliant radio	8K RAM	128K Flash	TinyOS, MoteWorks Support
Kmote [46]	TI MSP430	250 kbit/s 2.4 GHz IEEE 802.15.4 Chipcon	10k RAM	48k Flash	TinyOS and SOS
Mica2 [47]	ATMEGA 128L	Chipcon 868/916 MHz	4K RAM	128K Flash	TinyOS, SOS and MantisOS
MicaZ [48]	ATMEGA 128	Transceptor 802.15.4/ZigBee	4K RAM	128K Flash	TinyOS, SOS and MantisOS
Mulle [49]	Renesas M16C	Bluetooth 2.0	31K RAM	256K Flash	
Neo Mote [50]	ATmega 128L	TI CC2420 802.15.4/ZigBee compliant radio	4K RAM	128K Flash	TinyOS, SOS, MantisOS, Nano-RK and Xmesh Support, Industrial end-use product.
Redbee [51]	MC13224V	2.4 GHz 802.15.4	96 KB RAM + 120KB Flash		Contiki; standalone
Rene [52]	ATMEL8535	916 MHz with 10 kbit/s	512 K RAM	8K Flash	TinyOS
SenseNode [53]	MSP430F1611	Chipcon CC2420	10K RAM	48K Flash	GenOS and TinyOS Support
SunSPOT [54]	ARM 920T	IEEE 802.15.4	512K RAM	4 MB Flash	Squawk J2ME Virtual Machine
Telos [55]	Motorola HCS08		4K RAM		
Telos B [56]	Texas Instruments MSP430 microcontroller	250 kbit/s 2.4 GHz IEEE 802.15.4 Chipcon Wireless Transceiver	10k RAM	48k Flash	Contiki, TinyOS, SOS and MantisOS Support
Tinynode [57]	Texas Instruments MSP430 microcontroller	Semtech SX1211	8K RAM	512K Flash	TinyOS
T-Mote Sky [58]	Texas Instruments MSP430 microcontroller	250 kbit/s 2.4 GHz IEEE 802.15.4 Chipcon Wireless Transceiver	10k RAM	48k Flash	TinyOS, SOS and MantisOS
WeC [59]	Atmel AVR AT90S2313	RFM TR1000 RF			
XYZ [60]	ML67 series ARM/THUMB microcontroller	CC2420 Zigbee compliant radio from Chipcon	32K RAM	256K Flash	SOS Operating System

References

[1] Diana Bri, Hugo Coll, Miguel Garcia, Jaime Lloret, A Multisensor Proposal for Wireless Sensor Networks. Second International Conference on Sensor Technologies and Applications (Sensorcomm 2008). Cap Esterel, France, August 25-31, 2008. Pp.270-275

[2] I. R. Sinclair, Sensors and Transducers (Third Edition). Elsevier. UK. December 2000.

[3] H. Pasterkamp, S.S. Kraman, P.D. DeFrain and G.R. Wodicka, "Measurement of respiratory acoustical signals. Comparison of sensors", Chest: Official publication American College of Chest Physicians, Vol. 104, pp. 1518-1525, USA, November 1993.

[4] S. J. Birrell, K. A. Sudduth and S. C. Borgelt. "Comparison of sensors and techniques for crop yield mapping". Computers and Electronics in Agriculture, Vol. 14, Issues 2-3, pp. 215-233, USA, February 1996.

[5] R. C. Luo, M.-H. Lin, and R. S. Scherp, "Dynamic multi-sensor data fusion system for intelligent robots", IEEE J. Robot. Automat., vol. 4, Aug. 1988.

[6] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of Honda humanoid robot". IEEE Int. Conf. Robot. Automat. Vol. 2. Pp.1321-1326. 1998

[7] R. C. Luo, and M. G. Kay, "Multisensor Integration and Fusion in Intelligent Systems", IEEE Transactions on Systems, Man, and Cybernetics, Vol.19, No. 5, pp. 901-931, September/October 1989.

[8] R. C. Luo, C-C. Yih and K. L. Su, "Multisensor Fusion and Integration: Approaches, Applications, and Future Research Directions", IEEE Sensors journal, Vol. 2, Issue 2. Pp. 107–119. USA. April 2002

[9] D. L. Hall and J. Llinas, "An Introduction to Multisensor Data Fusion", Proceedings of the IEEE, Vol. 85, Issue: 1, pp. 6-23, New York, USA, Jan 1997.

[10] J.D. Cullen, N. Athi, M. Al-Jader, P. Johnson, A.I. Al-Shamma'a and A. Shaw, A.M.A. El-Rasheed. "Multisensor fusion for on line monitoring of the quality of spot welding in automotive industry". Measurement 41 (2008) 412–423.

[11] T. Yoshinobu, M. J. Schöning, R. Otto, K. Furuichi, Yu. Mourzina, Yu. Ermolenko and H. Iwasaki, "Portable light-addressable potentiometric sensor (LAPS) for multisensor applications", Sensors and Actuators B: Chemical, Volume 95, Issues 1-3, pages 352-356, 15 October 2003.

[12] E. Kuljanic, M. Sortino and G. Totis. "Multisensor approaches for chatter detection in milling". Journal of Sound and Vibration 312 (2008) 672–693.

[13] T. C. Henderson. W. S. Fai, and C. Hansen, "MKS: A multisensor kernel system". IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-14, No 5. Pp. 784-791, 1984.

[14] Cisco Systems Inc. At http://www.cisco.com [Last Access 7th June 2010]

[15] Paul Asadoorian and Larry Pesce, Linksys WRT54g Ultimate Hacking. Elsevier. USA. June 2007.

[16] Arduino Web page. At http://www.arduino.cc [Last Access 7th June 2010]

[17] Wrt54g-linux web page. At http://freshmeat.net/projects/wrt54g [Last Access 7th June 2010]

[18] BatBox web Page. At www.batbox.org/wrt54g-linux.html [Last Access 7th June 2010]

[19] Open WRT web page. At http://openwrt.org/ [Last Access 7th June 2010]

[20] DD-WRT web page. At http://www.dd-wrt.com [Last Access 7th June 2010]

[21] Nano WiReach specifications, Available at: http://www.connectone.com/media/upload/Nano_WiReach_PB.pdf [Last Access 17th June 2010]

[22] Mini Socket iWiFi, specifications, Available at: http://www.connectone.com/media/upload/Mini_Socket_iWiFi_DS.pdf [Last Access 17th June 2010]

[23] S103 WLAN Compact Serial Module specifications, Available at: http://www.rfsolutions.co.uk/acatalog/DS-S103.pdf [Last Access 17th June 2010]

[24] RN-111B specifications, Available at: http://www.rovingnetworks.com/documents/rn-111b-ds.pdf [Last Access 17th June 2010]

[25] RN-131G specifications, Available at: http://www.rovingnetworks.com/documents/rn-131-ds.pdf [Last Access 17th June 2010]

[26] RN-134 specifications, Available at: http://www.rovingnetworks.com/documents/rn-134-ds.pdf [Last Access 17th June 2010]

[27] Matchport b/g, Web site. Available at: http://www.lantronix.com/device-networking/embeddeddevice-servers/matchport.html [Last Access 17th June 2010]

[28] WiPort Web site. Available at: http://www.lantronix.com/device-networking/embedded-deviceservers/wiport.html

[29] Matchport b/g Pro, Web site. Available at: http://www.lantronix.com/device-networking/embeddeddevice-servers/matchport-bg-pro.html [Last Access 17th June 2010]

[30] RCM4400W RabbitCore module specifications, Available at: http://www.rabbit.com/products/RCM4400W/#description [Last Access 17th June 2010]

[31] Connect Wi-ME specifications, Available at: http://ftp1.digi.com/support/documentation/90000897_G.pdf [Last Access 17th June 2010]

[32] ConnectCore Wi-9C specifications, Available at: http://ftp1.digi.com/support/documentation/connectcore_wi9c_eu_doc.pdf

[33] J. Lloret, M. Garcia, D. Bri and J. R. Diaz. A Cluster-Based Architecture to Structure the Topology of Parallel Wireless Sensor Networks. Sensors. Vol. 9 Issue: 12. Pp. 10513-10544. December 2009.

[34] J. Lloret, M. Garcia, D. Bri and S. Sendra. A Wireless Sensor Network Deployment for Rural and Forest Fire Detection and Verification. Sensors. Vol. 9 Issue: 11. Pp. 8722-8747 October 2009.

[35] M. A. Menezes. "BEAN: Uma Plataforma Computacional para Rede de Sensores Sem Fio". April 2004. Available at http://homepages.dcc.ufmg.br/~mmvieira/publications/bean.pdf [Last Access 7th June 2010]

[36] J. Beutel, O. Kasten and M. Ringwald, "BTnodes -- a distributed platform for sensor nodes", Proceedings of the 1st international conference on Embedded networked sensor systems, November 05-07, 2003, Los Angeles, California, USA

[37] S. Coleri, S. Y. Cheung, and P. Varaiya, "Sensor networks for monitoring traffic". In 42nd Annual Allerton Conference on Communication, Control, and Computing, University of Illinois, September 2004.

[38] P. Dutta and D. Culler. "Epic: An Open Mote Platform for Application-Driven Design". In Proceedings of the 7th international Conference on information processing in Sensor Networks. pp. 547-548. Washington, DC, U.S.A. April 22 - 24, 2008.

[39] EYES Project. Available at http://www.eyes.eu.org/ [Last Access 7th June 2010]

[40] FlatMesh FM2. Available at http://www.senceive.com/Products.aspx [Last Access 7th June 2010]

[41] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal. Firefly-inspired sensor network synchronicity with realistic radio effects. In Proceedings of the 3rd international Conference on Embedded Networked Sensor Systems (SenSys '05). Pp 142-153. San Diego, California, USA, November 2-4, 2005.

[42] Mikko Kohvakka. “Medium Access Control and Hardware Prototype Designs for Low-Energy Wireless Sensor Networks”. Thesis for the degree of Doctor of Technology. May, 2009. Available at http://www.tkt.cs.tut.fi/research/daci/pub_open/Kohvakka-Medium_Access_Control_and_Hardware_Prototype_designs_for_Low-Energy_Wireless_Sensor_Networks.pdf [Last Access 7th June 2010]

[43] R. Kling, R. Adler, J. Huang, V. Hummel and L. Nachman. “Intel mote: Sensor network technology for industrial applications”. The Fourth International Symposium on Information Processing in Sensor Networks (IPSN'05). April 25-27, 2005, UCLA, Los Angeles, California, USA.

[44] Iris datasheet. Available at http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/IRIS_Datasheet.pdf [Last Access 7th June 2010]

[45] N. Madabhushi. “KMote-Design and Implementation of a low cost, low power hardware platform for wireless sensor networks”. May 2007. Available at http://www.cse.iitk.ac.in/~moona/students/Y5111028.pdf [Last Access 7th June 2010]

[46] Mica2 datasheet. Available at http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf [Last Access 7th June 2010]

[47] MicaZ datasheet. Available at http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAZ_Datasheet.pdf [Last Access 7th June 2010]

[48] J. Johansson, M. Völker, J. Eliasson, A. Östmark, P. Lindgren, and J. Delsing., “Mulle: A minimal sensor networking device - implementation and manufacturing challenges” in IMAPS Nordic 2004, pp. 265–271, Helsingør, Germany, September 26-28, 2004.

[49] Neo mote web. Available at http://www.xbow.jp/neokit-e.html [Last Access 7th June 2010]

[50] Redbee web. Available at http://redwirellc.com/store/index.php?route=product/product&product_id=53 [Last Access 7th June 2010]

[51] F. Zhao, L. Guibas, “Wireless Sensor Networks – An information processing approach”, Boston: Elsevier-Morgan Kaufmann publishers, 2004.

[52] Marcos Augusto M. Vieira, Claudionor N. Coelho. Jr, Diógenes Cecílio da Silva Junior, José M. da Mata1, “Survey on wireless sensor network devices”. Proceedings of IEEE Emerging Technologies and Factory Automation 2003, vol. 1, p. 537-544

[53] Sun spot website. Available at http://www.sunspotworld.com/docs/index.html [Last Access 7th June 2010]

[54] J. Polastre, R. Szewczyk, D. Culler. "Telos: Enabling Ultra-Low Power Wireless Research", The Fourth International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors, 2005.

[55] TelosB datasheet. Available at http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf [Last Access 7th June 2010]

[56] TinyNode datasheet. Available at http://www.tinynode.com/uploads/media/SH-TN184-103_rev1.1.pdf [Last Access 7th June 2010]

[57] T-mote sky datasheet. Available at http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf [Last Access 7th June 2010]

[58] J. McLurkin, “Algorithms for distributed sensor networks,” Master’s thesis, University of California, Berkeley, 1999.

[59] D. Lymberopoulos and A. Savvides. “XYZ: a motion-enabled, power aware sensor node platform for distributed sensor network applications”. In Proceedings of the 4th international Symposium on information Processing in Sensor Networks. Los Angeles, California, April 24 - 27, 2005.

Evaluation of Outdoor RSS-Based Tracking for WSNs Aiming at Topology Parameter Ranges Selection

Fotis Kerasiotis, Tsenka Stoyanova, George Papadopoulos
Applied Electronics Laboratory, Dept. of ECE, University of Patras, 26504 Rio-Patras, Greece
Industrial Systems Institute, Stadiou Str., 26504 Platani, Patras, Greece
{kerasiotis, tsstoyanova, papadopoulos}@ece.upatras.gr

Abstract— **The localization and tracking applications are among the most challenging applications, especially when the received signal strength (RSS) is utilized. The RSS is known to be a noisy signal and difficult to use in localization and tracking applications. In the present study we investigate the possibility a target tracking task to be performed with the resources of WSN technology, when using only the RSS of the exchanged messages. We demonstrate that RSS can be used for outdoor localization and tracking application under well-defined topology constraints. The present work presents detailed study about the topology parameter selection when a tracking application is considered. Moreover, the RSS uncertainty is defined in order to be included in the simulation of a tracking scenario. The target tracking considerations by means of tracking techniques, topology parameters and factors influencing the tracking accuracy are combined in simulation examples to evaluate their significance concerning the performance of the tracking task. Furthermore, the propagation model and the topology parameters being identified are used in real outdoor tracking test.**

Keywords – wireless sensor networks, tracking, received signal strength, simulation

I. INTRODUCTION

Wireless sensor network (WSN) is a technology aiming at providing observation of the environmental events and objects with minimal human supervision. The field of WSN encompasses a very broad array of applications including monitoring systems, smart environments, target localization and tracking and a lot of others.

The base of any localization and tracking system is the mechanisms used to determine the position of a fixed or a mobile object by measuring physical distances in indirect way. Different techniques, which provide relation between the distance and some measurable parameter of the transmitted signal, such as time of flight, time difference of arrival, angle of arrival and the strength of the measured signal, can be utilized.

The target tracking applications are based on localization information exchange among the sensor network nodes, which require collaborative sensing, communication and computation among multiple sensors that observe the target objects. The network topology and the influence of the environment on the in-network communication and on the signal used for tracking are also important factors that need consideration [1]. In general, the target tracking application scenario is composed of four functional components, namely deployment, localization, target tracking and information exchange.

Sensor nodes' deployment reflects two main aspects of a WSN, namely sensing and communication. The main sensor network problems that the deployment addresses are related to optimal area coverage and network connectivity [2]. When tracking application is concerned, defining of the deployment constraints and the topology parameters is also part of the deployment phase. After the network topology is formed the localization task begins. Various localization methods, presented in the literature, are generally divided in range measuring algorithms [3-5] and range-free algorithms [6-8]. When the positions of all sensor nodes within the network are determined, the tracking task can be implemented. Since the main goal of tracking is localization of moving objects, some of the above mentioned localization methods, mostly the range measuring algorithms, could be also utilized. Additionally, tracking oriented algorithms and protocols, based on mobile agents and its data fusion or using binary detection have been also developed [9-11].

The RSS-based localization-tracking techniques that deal with position estimation mainly belong to two categories, namely trilateration or multilateration, which depends on the number of used beacon nodes, and fingerprinting. Trilateration uses a signal propagation model to extract the RSS/distance relation, and together with the beacon nodes' position to estimate the position of a mobile object. On the other hand, the fingerprinting approach is based on acquiring reference points within the target area. To these reference points signature vectors are associated consisting RSS data taken by each of the beacon nodes. The signature vector is formed either by real measurements in the target environment, either by simulation performed with an appropriate for the environment RF propagation model. All reference points' positions and their signatures construct a database. During the localization phase, the RSS measurements, taken from the beacons for a particular mobile object, form a tested signature. The tested signature is compared with the signatures in the database to find the likeliest one and the corresponding reference point location is accepted as a location for the mobile object.

Most of the RSS-based state-of-the-art localization and tracking algorithms for outdoor environment utilize a propagation model to discover the distance, relying on that the RSS decreases with the distance by well-known physical low. However, most of the proposed tracking algorithms use simplistic models, usually supposing an ideal exponential characteristic of the RSS variation due to distance, without taking into consideration the influence of the environment and the topology parameters to the propagation characteristic [5, 6, 9]. Thus, the present state-of-the-art localization and tracking techniques do not offer satisfying solutions to major localization/tracking problems and applying them successfully in real outdoor tracking applications is doubtful.

In general, many factors have to be considered when a RSS-based tracking application is designed, starting from selection of proper propagation model, which has to represent in a relatively accurate way the interaction between the RF signal and the environment. The topology parameters, such as number of beacons, beacons' height and optimal beacons' distance, have to be also evaluated for their optimal values.

The study in previous work on RSS behavior for varying impact factors [12], has led to important conclusions that drove the research presented in [1] and extended in the present work. In the present study we investigate the possibility a target tracking task to be performed with the resources of WSN technology, when utilizing only the radio signal strength (RSS) of the exchanged messages. RSS is notorious for being a noisy signal that is difficult to use for ranging-based localization and tracking applications. In this study, we demonstrate that RSS can be used for outdoor localization and tracking application under well-defined topology constraints.

The present work elaborates on [1], by means of presenting more detailed study about the topology parameter selection, the RSS uncertainty identification and including it in the simulation of tracking scenario, more simulation examples and finally, performing of real outdoor tracking test. All real measurements were performed with Tmote Sky [14] or TelosB [21] sensor nodes, which both have CC2420 radio chip. Their working frequency is at 2.4GHz band. Taking advantage of the measurements performed for [23] we selected the central frequency of channel 26, i.e., 2.48GHz, as working frequency due to not overlapping with the available in the areas WLANs. All simulations were performed with the selected radio frequency.

The rest of the paper is organized as follows: In Section II, the most important problems encountered in a tracking application are presented, including the influence of the RF propagation, the deployment constrains and the topology parameters on the tracking performance. Section III discusses the influence of the mobile node position on the RF signal propagation in order to identify the most promising ones. The target tracking considerations by means of tracking techniques, topology parameters and factors influencing the tracking accuracy are presented is Section IV followed by simulation examples. In Section V the propagation characteristics and topology constraints identified in the previous sections are used in real outdoor tracking test. Finally, Section VI concludes this work.

II. Analysis of Tracking Problem Dimensions

The tracking application scenario under consideration is based on a triangular topology of fixed nodes and has two main objectives: (1) localization of the triangle, inside which the mobile node moves and (2) tracking the position of the mobile node, which is placed on human body. The tracking application approach is based on the assumption that the mobile node transmits packets periodically and the fixed nodes determine the mobile node's position by measure the RSS of the packets. Such a tracking algorithm has been implemented making use of mobile agents [9].

The tracking application scenario under study is shown in Fig. 1. It concerns outdoors target-tracking, using the widely-used Tmote Sky or TelosB sensor nodes. The fixed sensor nodes, named also beacons, are deployed in triangular grid. The server, which is to be placed probably in a building, receives information from the fixed WSN nodes about the position of the mobile nodes, i.e., the people, as well as additional information such as: patient vital signs. The application under consideration should satisfy specific constraints:

- Accuracy of the tracking process: up to 10m
- Sampling period of the target node location: 1sec
- Topology: fixed nodes in triangular grid at approximately 50m distance.

The study in the present work is focused on the second objective mentioned above, i.e., tracking the position of the mobile node inside the triangle, and more specifically, on identifying the most crucial RSS-based tracking problems, and determining and evaluating the topology parameters that can guarantee successful tracking. Based on this, the aims of the present study can be formulated as follows: (1) evaluation of the behaviour of the propagated RF signal in order to identify the important RSS-based tracking problems, (2) selection of the topology related parameters, when taking into consideration the tracking task requirements and the deployment constraints, (3) use of the selected propagation model and topology parameters to simulate a target tracking task, and (4) keeping the proposed topology parameters to perform real outdoor tracking test.

In the following sections the factors that may cause tracking problems, i.e., impossibility to perform tracking, are discussed.

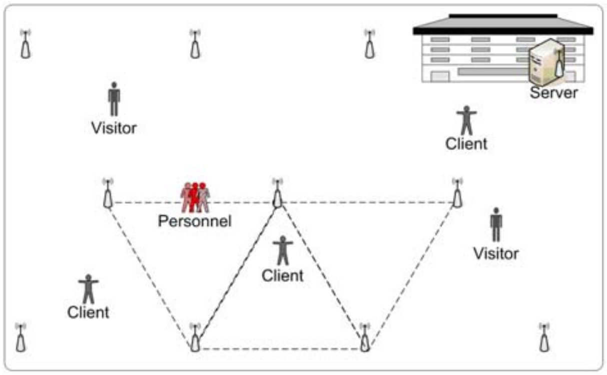


Figure 1. Tracking application scenario

A. Basic characteristics of the behavior of the propagated signal

As shown in [12], theoretical models can describe the behaviour of a transmitted signal from the receiver side, considering free space and ground surface reflections. The main parameters affecting the RF signal propagation are the height from the ground for the transmitter-receiver pair (T-R), the frequency of the transmitted signal and the type of the ground (e.g. grass), assuming that the environment is open space without obstacles. In another work [18] we introduced FOM, a model for free space outdoor environments, where the effect of some other important parameters on propagated signal such as variation of transceivers, radio frequency gain, antenna pattern irregularities, and RSS uncertainty, are also considered. Fig. 2 presents real outdoor measured data and simulated data, performed with FOM. For these measurements and simulations the receiver height is at 2m and the transmitter height is at 1.1m in Fig. 2(a) and at 0.08m in Fig. 2(b). The results show a good correspondence between real measurements and simulation. The results presented here as well as those in [18] confirm that this model can be used during the evaluation of the topology parameters with sufficient accuracy of the tracking task for the target application.

The model described in [18], i.e., FOM, is used in the rest of the paper, when a simulation of the RF propagation is mentioned. Its formulation is given with (1) and (2).

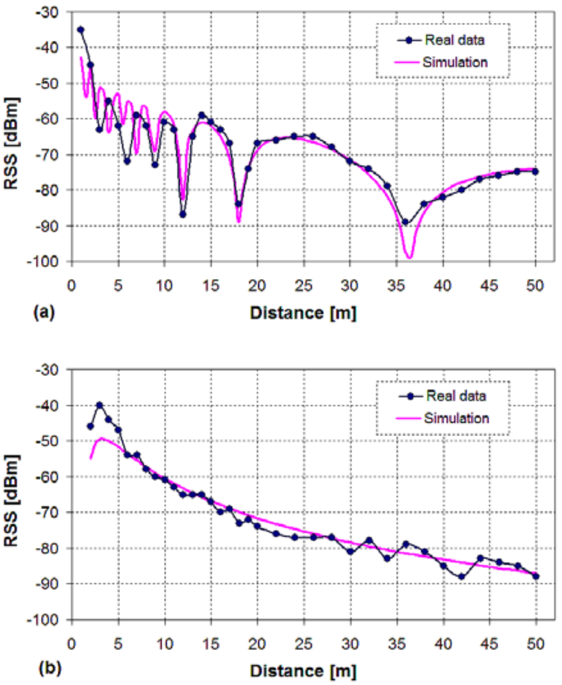


Figure 2. Simulation vs. experimental RSS characteristic for receiver at 2 m and transmitter at: a) 1.1m and b) 0.08m

$$\overline{P_R}(d) = P_T\left(\frac{\lambda}{4\pi d}\right)^2\left(K_1^2 + K_2^2\Gamma^2 + 2K_2\Gamma\cos\left(\frac{2\pi}{\lambda}\Delta L\right)\right) \qquad (1)$$

where $\overline{P_R}(d)$ is average value of the received power.

When P_R is expressed in decibels with RSS uncertainty included as $X_{\sigma(\overline{P_R})}$, then the formulation is:

$$P_R = \overline{P_R}(d) + X_{\sigma(\overline{P_R})} \qquad (2)$$

where P_R is the received power, P_T is the transmission power, d is T-R distance, λ is the wavelength, Γ is the ground reflection coefficient, ΔL is path length difference between the direct and the reflected signals, coefficients K_1 and K_2 are antenna specifics representing the gain in particular antenna orientation, which are described in detail in [18]. The RSS uncertainty is given as a Gaussian random variable X with distribution $\sigma(\overline{P_R})$.

Fig. 3 shoes three simulations of the RF propagation, when the fixed node is at 2m, and the mobile node is at 0.1m, 0.5m and 1.4m. One important observation regarding the effect of the ground reflection phenomenon on the RSS is that the reflected and the direct signals interact and create 'nulls', where the RSS is very low and quite variable, as we observed in [12], [18] and [22]. Another observation is that the 'null' areas move with the change of the height, which is connected with the change of the path length of the direct and reflected signal and phase difference.

From the tracking algorithm point of view, one could conceivably take advantage of these 'nulls' and through trilateration, combine the RSS levels of the three fixed nodes to distinguish the possible positions of the mobile node in a triangular region. The pattern effect on the RSS-based tracking scheme of Fig. 3 is investigated in Section III, so as to highlight the risks and uncertainties that it may produce. A more promising approach is to ensure that the RSS characteristic is "smooth", that is to have as few as possible 'nulls'. This can be achieved by judicious selection of the heights at both transmitter and receiver.

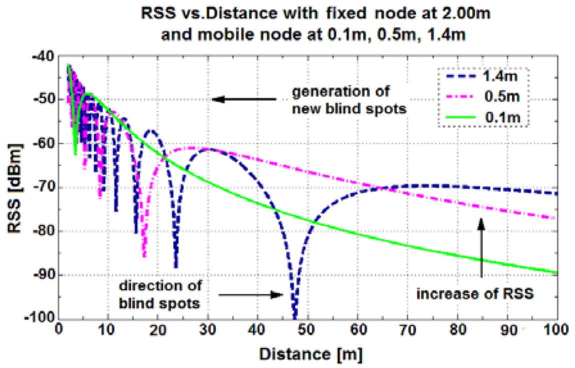


Figure 3. Propagated signal for different mobile node heights and fixed nodes at 2m

B. Deployment constraints

Based on experiments and simulations performed in previous work [22], we discovered that the combination between the fixed node's height and distance is limited, considering the fact that the communication among them has to be as robust as possible with respect to minimum packet loss. This limit, also named safe RSS threshold, is defined by that RSS level, at which the packet loss increases more than 5%, and this is about -80dBm. Therefore, the heights which can be chosen for the fixed nodes are defined according to this RSS level. Fig. 4 shows simulation, where the RSS is simulated with respect to the fixed nodes heights (between 0.10m and 3m) for distance at 50m, which is one of the topology requirements. The red doted horizontal line expresses the safe RSS value, which is the minimum value for reliable communication among the fixed nodes. Therefore, the selected heights of the fixed nodes for reliable communication among them at distance of 50m are: (a) 0.50m–1.65m, (b) 1.85m–2.35m, and (c) 2.55m–2.90m.

C. Mobile node positions on the human body

Having assured a robust communication for the fixed nodes by heights selection as described above, the next parameter under consideration is the height of the mobile node. Possible positions of the mobile node on the human body are: ankle, knee, waist, wrist, chest and arm, shown in Fig. 5. Table I shows the correspondence of each of these positions with the average heights from the ground including also the variation which may be introduced due to movements of the human. There are small height variations for some body parts such as knee and ankle and greater ones for others such as waist, chest and arm and even greater considering the movement of the wrist when raising. These variations are caused by every possible movement the human target can perform, such as walking, running, sitting, bending and raising his/her hands.

TABLE I. MOBILE NODES POSITIONS ON THE HUMAN BODY AND VARIATION DUE TO MOVEMENTS

Position on Human Body	Standing Position Height	Height Variation Including Body Movement
Chest	1.4 m	from 0.9 to 1.45 m
Arm	1.4 m	from 0.9 to 1.45 m
Weist	1.1 m	from 0.5 to 1.20 m
Wrist	1.0 m	from 0.5 to 2.00 m
Knee	0.55 m	from 0.45 to 0.70 m
Ankle	0.15 m	from 0.15 to 0.30 m

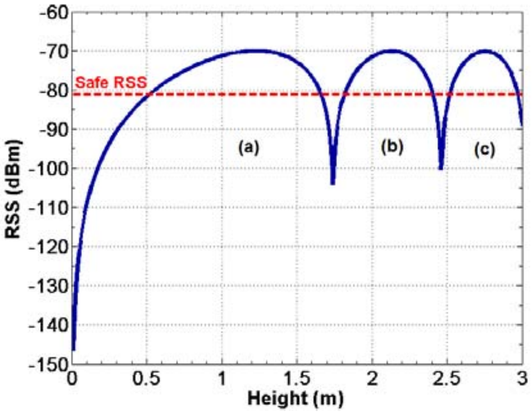


Figure 4. Deployment constraints for the choice of fixed nodes' height

D. RSS-based mobile node position estimation

When the tracking task is performed through beacon nodes, the position estimation of a mobile object can be performed via two techniques: fingerprinting and trilateration. Trilateration uses the signal propagation model and the beacon nodes positions to convert RSS values to a distance measurement so as to estimate the position of a mobile object with trilateration techniques. On the other hand, the fingerprinting approach is based on acquiring reference points within the target area (the triangle in our case). To these reference points are associated signature vectors consisting of RSS data taken by each of the beacon nodes. The signature vector is formed either by real measurements in the target environment, or by simulation performed with an appropriate for the environment RF propagation model. All reference points positions and their signatures construct a database. During the localization phase, the RSS measurements taken from the beacons for a particular mobile object form a tested signature. The tested signature is compared with the signatures in the database to find the likeliest one and then the corresponding reference point location is accepted as a location for the mobile object.

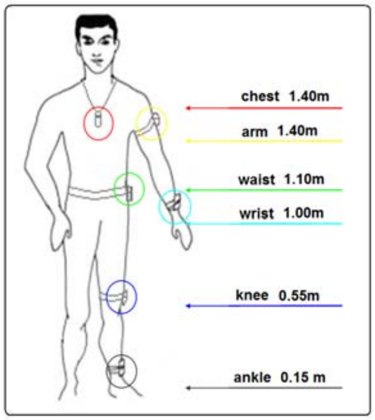


Figure 5. Possible positions of the mobile node on the human body

For the purpose of this work the fingerprinting approach is considered and two techniques for defining the locations of the reference points are evaluated, namely quantized levels and square grid. In the quantized levels, the RSS in the range between -40dBm and -90dBm is divided in discrete levels, to which correspond different distance regions. Using the RF propagation model to simulate the propagation within the triangle, the location of the reference points is chosen so that at least one reference point can exist in each region, differentiated by the discrete levels. The square grid approach for positioning of the reference points is quite simple. The target area is divided out in small squares, the size of which determines the minimum localization accuracy. Detailed analysis of these two techniques is offered in Section IV.

Summarising the issues discussed in this section the following conclusions were derived:

- The RF signal propagation has to be considered while choosing the topology parameters, such as beacons' distance and height, in order to assure reliable communication between the beacons.
- The position of the mobile node on the human body should be selected so as to minimize the node's height variation due to body movement and exclude the 'nulls' from the propagation curve.
- When using fingerprinting approach the selection of the reference points' locations needs preliminary evaluation according to well defined criteria such as: number of reference points, accuracy, simplicity of generation, etc.

III. Mobile Node Position Evaluation

In previous work [12, 18], it has been shown that the height of the sensor nodes heavily affects the behavior of the propagated signal. Moreover, the deployment constraints for reliable communication limit the possible fixed nodes' height within the ranges determined in the previous section as 0.50m – 1.65m; 1.85m – 2.35m, and 2.55m – 2.90m, for 50m distance between the beacon nodes. Therefore, in this section, the RSS behaviour is studied with respect to the possible mobile node positions listed in Table I.

A. *Analysis of the mobile node at upper heights*

Assuming that higher placements are selected for the mobile node on the human body e.g. chest, arm, wrist and waist, we select the height at 1.1m as a representative height for the mobile node. The corresponding behaviour of the propagation signal when the mobile node is at 1.1m is shown in Fig. 6. The simulation is performed through (1) for four fixed nodes' heights selected from the regions suggested by the deployment constraints for fixed nodes, as follows: 0.55m and 1.40m are from region (a), 2.10m is from region (b) and 2.7m is from region (c) in Fig. 4. Analysing the results shown in Fig. 6, it is noticeable that when the mobile node is placed at 1.1m and the fixed nodes' height is 2.10m or 2.70m, the RF propagation between 1m and 50m is full of 'nulls' and to extract distinguishable RSS/distance combinations for the quantizing tracking technique is impossible. The case when the fixed nodes are at 1.40m height presents less 'nulls', but the problem with separating the signal propagation on distinguishable RSS/distance combinations still exists. The last beacons' height of 0.55m is more promising in the sense that the RSS curve is smooth and some distinct regions can be derived for the distances between 15m and 50m. The drawback is that the difference between the RSS in 15m and the RSS in 50m is only 12dBm, which implies separation in no more than two regions. Using only two regions cannot provide acceptable tracking accuracy.

As a conclusion of this analysis, when the mobile node is at about 1.10m height, there is no appropriate height for the beacon nodes so as to guarantee reliable communication between the beacon nodes and consequently successful tracking.

B. *Analysis of the mobile node at low heights*

According to Fig. 5 and Table I the lower heights for placement of the mobile node on the human body are: the knee at height of 0.55m and the ankle at height of 0.15m. To analyse the behaviour of the propagated RF signal when the mobile node is at 0.55m and 0.15m, simulations are performed. The results for the mobile node height at 0.55m are shown in Fig. 7(a), and for the mobile node height at 0.15m in Fig. 7(b). For both mobile node's heights the fixed nodes are at heights selected from the regions suggested by the deployment constraints for fixed nodes, i.e., 0.55m and 1.60m from region (a); 2.10m from region (b) and 2.7m from region (c) in Fig. 4.

Referring the results in Fig. 7(a) for mobile node at 0.55m, the RF propagation results for the beacons' height at 2.7m, 2.10m and 1.60m show similar problems to those of the waist case. This makes the particular beacon heights inappropriate for a successful tracking task. The smallest beacon height of 0.55m seems more promising for the distances between 6m and 50m. The propagation curve has good slope, which gives variance between the RSS at 6m and the RSS at 50m of about 25dBm. This variance suggests the use of the traditional tracking methods with linear models and filters as well as the quantized levels technique.

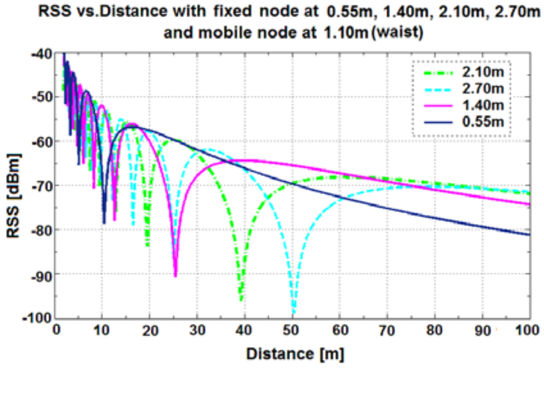


Figure 6. Behavior of the propagated signal or RSS concerning different heights for the fixed nodes and waist mobile node position

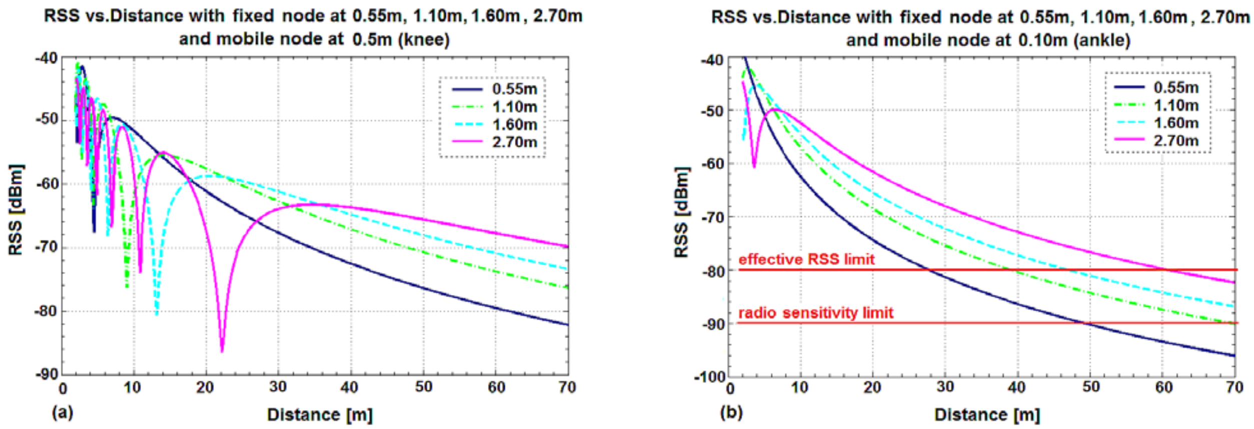


Figure 7. Behavior of the propagated signal or RSS concerning different heights for the fixed nodes and:
a) knee and b) ankle mobile node position

Almost the same applies to the ankle case for which RF propagation results for the beacons' heights at 2.7m, 2.10m, 1.60m and 0.55m are shown in Fig. 7(b). The propagation curves are without 'null' regions and resemble the propagation of the widely used long-distance path loss model [13]. The beacons height at 0.55m seems the most suitable due to the most inclined curve slope. However, considering that the Tmote-Sky radio sensitivity level is about -90dBm,, then the communication between the mobile node and the beacons, when the distance is approximately 45m-50m, could be seriously limited or totally fail, which results in insufficient received packets for performing the tracking task. This implies that, for the ankle case, the fixed nodes should be placed at a height above 1.00m in order to maintain an "effective" RSS level at 50m assuring robust communication between the mobile node and the beacons.

As a conclusion of this analysis, the combinations of 0.55m mobile height with 0.55m beacon height, and 0.15m mobile height with beacon height above 1.00m present high potential for successful tracking.

C. *Analysis of the mobile node height variability due to movement*

From the analysis performed in the previous two sections, some of the mobile-node-height/beacon-height combinations are selected for further investigation for their approval like appropriate topology parameters. These combinations are:

- height of 0.55m for the mobile node and height of 0.55m for the beacon nodes, i.e., knee case,
- height of 0.15m for the mobile node and height above 1.00m for the beacon nodes, i.e., ankle case.

Another factor that may influence the behaviour of the propagated signal is the height variation of the mobile node due to the movement of the human body. Depending on the mobile node position, its height varies during the body movement as shown in Table I. This variance ranges from 10cm at the knee to 1 m at the wrist. Since the knee and ankle positions, were selected as the most promising positions, only the variation of these two positions will be analysed.

According to Table I the ankle movement varies between 0.15m and 0.30m during walking. In order to study the behaviour of the propagated signal due to the variation of the mobile node height, several simulations are performed. The height of the beacons is chosen at 1.50m, and for the mobile node three heights are selected: 0.10m, 0.20m and 0.30m. The simulated results for the three mobile heights are shown in Fig. 8(a). If we suppose that the ankle can change its height within 20cm during walking, then this corresponds to about 10dBm variation of the RSS measurements. If the RSS is measured during walking, this variation will be expressed like spikes up and down around the average propagation curve.

The average value for knee height is selected at 0.55m. During walking the knee height does not change more then ±5cm. However, there is another factor that imposes more variability of the knee height namely the difference among the people's body heights. Taller people have knee height approximately at 0.65m, while short people about 0.45m. Based on this, for the simulation about the influence of the knee height on the RF signal propagation we select four knee heights: 0.40m, 0.50, 0.60 and 0.7m. The selected values are a bit lower or higher that the real ones, so as to assure that we do not underestimate the knee-height variability effect over the RF propagation. Fig. 8(b) depicts the RF propagation results for the four knee heights when the fixed node is at 0.65m. The results show that the knee case is less influenced by the knee height variation, as a variation of 30cm causes the RSS-level to vary about 5 dBm. As in the ankle case, the height variation due to walking will be expressed like spikes up and down around the average propagation curve, which is discussed in Section V.

Considering the analysis in this section, obstacles are either absent or have an equal effect on all cases. In any other case, obstacles should be modeled appropriately.

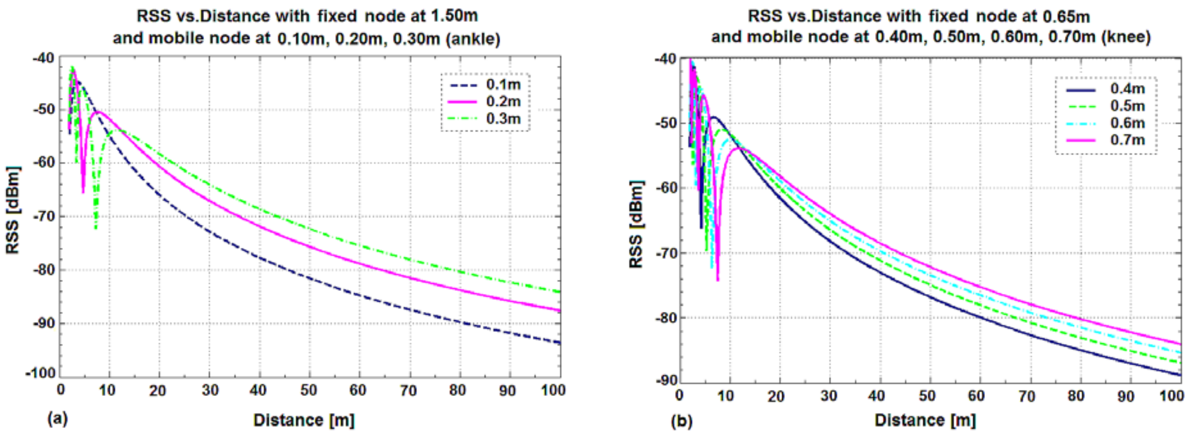


Figure 8. RSS behavior for different fixed node heights and: a) 0.1m or b) 0.5 m for the mobile node

IV. Target Tracking Considerations and Simulations

The analysis in Section II and in Section III clarified that knowing the RF propagation of the target environment and making careful selection of the heights and positions of the beacons and the mobile nodes, theoretically present possibility for performing RSS-based tracking with WSNs successfully.

In this section, the tracking technique, the factors influencing the tracking accuracy and the simulation procedure are discussed.

A. *The Tracking Technique*

As it was mentioned in Section II the fingerprinting approach is utilized in the present work. It is based on acquiring reference points within the target area (the triangle in our case). To these reference points are associated signature vectors consisting RSS data taken by each of the beacon nodes. The signature vector is formed by simulation performed with (1), i.e., FOM. All reference points positions and their signatures construct a database. During the localization phase, the RSS measurements, taken from the beacons for a particular mobile object, form a tested signature, which is compared with the signatures in the database to find the likeliest one and then the corresponding reference point location. Two techniques for defining the locations of the reference points are evaluated, namely quantized levels and square grid.

(a) Quantized Levels Approach

The quantized levels approach is based on dividing the RSS range between -40dBm and -90dBm in discrete levels, to which correspond different distance regions. The division is made so that each level can have almost the same width in dBm, and at the same time the width to be approximately 10dBm. In this sense, five RSS levels are chosen for the RF propagation curve when the fixed node height and the mobile node height are at 0.55m, as shown in Fig. 9(a). The corresponding distance ranges suppose tracking accuracy of 10m. It is observed that at short distances, up to 10m, the signal is a bit variable with small and narrow 'nulls', which

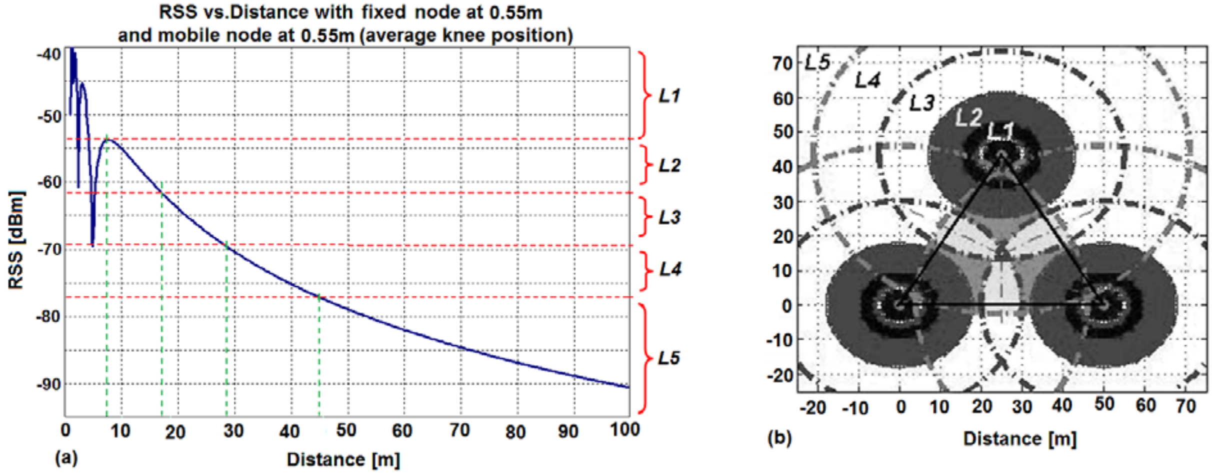


Figure 9. Quantized levels topology at 0.55 m (knee position) and fixed node at 0.55 m:
a) RSS vs. distance and RSS-levels, b) separation in recognizable RSS levels

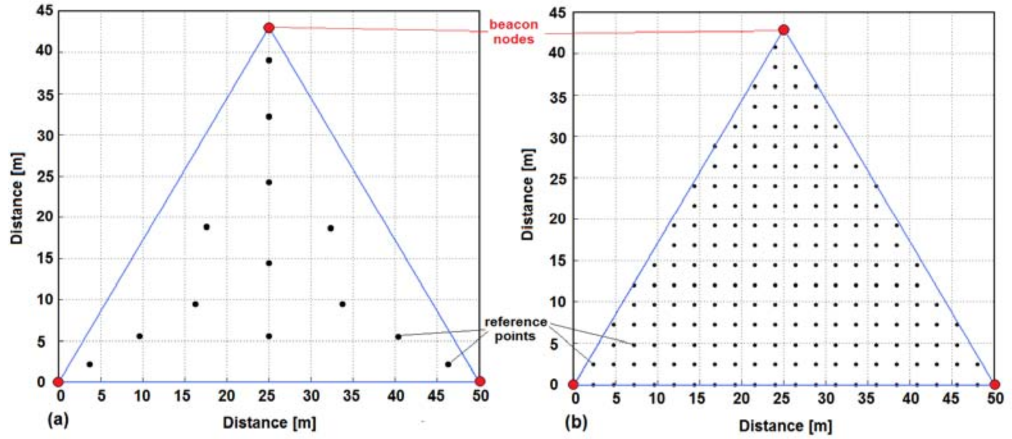


Figure 10. Reference points topologies for: (a) quantized levels techniques with 13 points, and (b) square grid technique with 289 points

may impose difficulty when estimating the mobile object position. Nevertheless, it can be filtered or handled by the use of an additional sensor such as ultrasound or passive infra-red (PIR) with respective techniques proposed [15].

The location of the reference points is chosen so that at least one reference point can exist in each region, differentiated by the discrete levels. An example of separation of the triangle area in recognizable RSS regions is shown in Fig. 9(b). According to this simulation 13 distinguishable regions are identified and hence 13 reference points are selected. The topology of the 13 reference points is shown in Fig. 10(a). The positions of the reference points have to be recalculated for each beacons-height/mobile-node-height combination.

The tracking task takes place inside the triangle when the RSS level provided by each of its fixed nodes is above level L5. If one of the fixed nodes receives RSS level 5, then the triangle may have to be changed and the tracking task to be transferred to the neighbourhood triangle. Concerning the interior of each triangle, there are some well-defined regions being characterized by different combination of each fixed node RSS. Using these regions, the position of the mobile node can be tracked with accuracy of about 10m at the centre of any of these regions.

(b) Square Grid Approach

The square grid approach for generation of the reference points is quite simple. The target area is divided out in small squares, the size of which determines the minimum localization accuracy. Such a reference-point grid is shown in Fig. 10(b). The size of the squares could be 1/10, 1/20, 1/25 and 1/50 of the beacons' distance to which correspond the following number of reference points: 81, 289, 441 and 1764.

During the tracking task, to every reference point is associated a vector-signature, consisting of the simulated RSS between the reference point and each beacon. The positions of all reference points as well as their signatures compose the signature database.

(c) Extension of the 50m Topology

The above analysis is based on the assumption that the distance among the beacon nodes in a triangle is 50m. Having selected the maximum transmission power of 0dBm, a combination of heights for both fixed and mobile nodes is required for guaranteeing fulfilment of the tracking task requirements. However, this particular analysis can also be held for other dimensions, since the combination of heights can be matched to the new required distances in order to produce similar RF signal propagation as for 50m. The same behavior can be achieved by increase of the beacons' height or by increase of the transmission power through connecting an external antenna with higher gain or through using another platform with greater transmission power. Fig. 11 shows simulations of the RF signal propagation, when the mobile node height is at 0.55m, i.e. knee. Three cases are presented: fixed node at 0.55m and 0dBm transmission power, fixed node at 0.55m and 4.5dBm transmission power, and fixed node at 1m and 0dBm transmission power.

The aim of these simulations is to investigate the situation where the communication distance between the beacons has to be increased, for instance to 70m, and thus how the topology parameters have to be modified in order

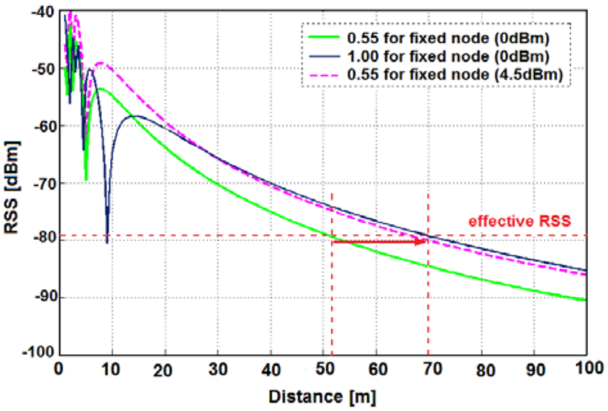


Figure 11. Extension of the 50m topology to a 70m topology

the new RF propagation to fit the one produced by the topology of 50m beacons' distance. The results from Fig. 11 show that it may not be necessary any parameters to be modified, since the RSS in 70m is around -85dBm, which is above the sensitivity threshold of the radio chipCC2420. Nevertheless, this value is lower than the adopted safe threshold level of -80dBm and according to [22] doubles the percentage of the packet loss. Therefore, it is advisable the RSS in the end distance of 70m to be around -80dBm, which require some parameter modifications.

The first and easier solution is to increase the beacons' height, since the mobile is fixed at the knee. As it is shown in Fig. 11, if the beacons' height is 1m, then the propagation curve shifts up and the RSS at 70m is around -80dBm. Unfortunately the last 'null' now is deeper and wider, and it is located at greater distance. In addition, the curve slop is not that inclined, which may worsen the tracking accuracy.

The second solution, which requires other sensor node platform, or more powerful antenna, is to increase the transmission power. As it is shown in Fig. 11 if the transmission power is 4.5dBm the propagation curve is exactly like the one for 0dBm, but up-shifted. This shift is enough so that the RSS at 70m to be -80dBm.

The two solutions have advantages and disadvantages. The first uses the same hardware, but may have less accuracy, while the second keep the same accuracy as for 50m distance but require hardware modification and will have more energy consumption.

B. Other Factors Influencing the Tracking Accuaracy

In the sections above we analysed the most important topological parameters to determine their importance and effects on the tracking task performance. However, knowing them is not enough for realistic tracking simulation or for contracting a reliable tracking algorithm. There are two factors that also have to be known and included during the simulation procedure or to be used for designing a tracking algorithm, and these are:

- The variability of the RSS due to its noisy nature and due the movement of the human body, and
- The sensor nodes' hardware variability, which may cause difference of 10dBm in the RSS measurements for the same parameters distance, height and transmission power.

(a) RSS uncertainty

As it is well known, RSS does not have a deterministic behavior, but presents random variation. This is most likely due to the radio hardware, nodes movement and incomplete description of the RF communication link. It is of great importance for the localization and tracking applications to identify and characterize the RSS uncertainty sources as well as to model them for more realistic tracking simulations [23]. In order to investigate the RSS variability when the transmitter and receiver are static, and when the transmitter is mobile and the receiver is static, several outdoor measurements were performed. Detailed description of the measuring setup and modelling procedure are given in [23].

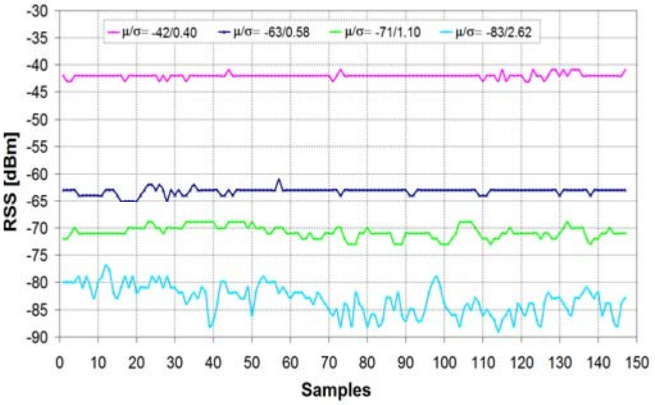


Figure 12. RSS measurements for four different combinations of μ and σ

The measurements for static transmitter and receiver were performed for different heights from the ground and distances in order to have more distinct RSS levels. Then the mean value μ and the standard deviation σ were calculated. Since the noise is considered to derive from multiple factors we assume Gaussian distribution. Fig. 12 presents RSS measurements for four different μ/σ combinations in dBm: -42/0.40; -63/0.58; -71/1.10 and -83/2.62. As Fig. 12 presents, the strongest signal, i.e., highest μ, has the smallest deviation, and the lowest signal has the highest deviation, i.e., the measured values are more dispersed regarding the mean value. It is noticeable that with the decrease of the signal strength the standard deviation increases, which could be expressed as $\sigma = f(\mu)$.

The measurements for fixed receiver and mobile transmitter were performed during human walking along a circle arc of 20m and 30m for the knee position case and of 15m and 25m for the ankle position case with speed of approximately 1m/s. The fixed node is in the centre of the circle arc and receives the transmitted signals of the mobile node positioned above the knee. In Fig. 13, we present the results for the RSS variation due to the movement of the

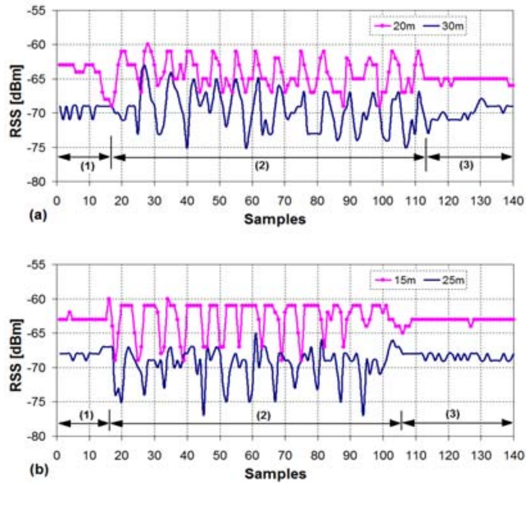


Figure 13. RSS measurements for (a) knee and (b) ankle transmitter position

transmitter, while the human walks. These plots have three distinguishable regions: (1) two seconds in immovable state, (2) walking state and (3) three to four seconds in immovable state. The average RSS value corresponds to the steady position of the human body, where the mobile node has stable height 0.55m from the ground. As it was expected, except for the RSS uncertainty owning to the RSS nature, there is an additional uncertainty introduced by the motion of the mobile node. The RSS variation due to the body movement is periodical and can be correlated with the trajectory of the legs movement. In particular, during walking, the height and antenna orientation change in a periodic manner, which results into periodic variations, i.e., spikes up and down around the mean value, of the RSS.

The modeling of the relation σ = *f* (*μ*) consists of data-to-curve-fitting approximation over the measured data [23]. The modeling equations for static transmitter, shown in Fig.12, and for mobile transmitter, shown in Fig. 13, are (3) and (4) respectively.

$$\sigma(\mu) = a.\exp(-b.\mu) + c\,, \qquad (3)$$

where $a = 0.04$, $b = -0.045$, $c = 0.2$

$$\sigma(\mu) = a.\mu + b\,, \qquad (4)$$

where $a = -0.029$ and $b = 0.48$.

Equation (3) is to be used in RSS uncertainty generation process depending on the mean value when the sensor nodes are fixed in outdoor unobstructed environment, while (4) includes both the uncertainty due to RSS nature and the uncertainty due to movement and it is to be used in noise generation process depending on the RSS mean value when the transmitting node is mobile.

(b) Sensor Node Hardware Variability

It is well known that there are differences in the radio circuits among the same type of transceivers. This hardware variability, due to production tolerance, leads to difference among actual transmission powers for the same sensor node types. To study the transmitter and the receiver variability several experiments were conducted [12]:

- one receiver and five different transmitters, and
- one transmitter and five different receivers.

The orientation and the position of the receivers and the transmitters are exactly the same. The height from the ground of the receivers and the transmitters is 0.70m. The experiments are performed for three different transmission powers of 0dBm, -3dBm and -11dBm at 50m, 30m and 15m T-R distances, respectively. The results are shown in Fig. 14.

The hardware variability of the transmitters and receivers results in difference of the measured RSS when five transmitters send and one receiver receive, as shown in Fig. 14 in the first column, and when one transmitter sends and five receivers receive, as shown in Fig. 14 in the second column. Moreover, the results present that the variation of the RSS among the five transmitters and receivers does not follow the same logic when change the transmission power. For instance, the node with ID 3, as a transmitter, has the highest RSS value for transmission power -3dBm at 30m in Fig. 14(c) and the lowest RSS value for transmission power -11dBm at 15m in Fig. 14(e). Similar observations hold for the results of the five receivers.

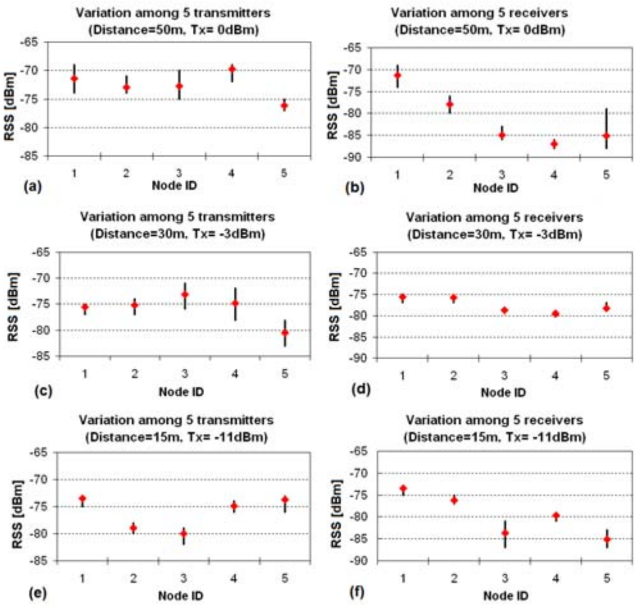


Figure. 14 Variation of the transceivers

Taking into account the results from Fig. 14, there is one more parameter to be included in (1), which represents the offset that the hardware imposes, and that is the P_{offset}. Thus, the received power can be expressed as:

$$\overline{P_R}(d) = P_T\left(\frac{\lambda}{4\pi d}\right)^2\left(K_1^2 + K_2^2\Gamma^2 + 2K_2\Gamma\cos\left(\frac{2\pi}{\lambda}\Delta L\right)\right)P_{offset} \qquad (5)$$

where P_{offset} is given in mW in this equation and expresses the difference between the RSS produced by simulation with (1) and the real measured RSS. Therefore, the hardware variability strongly requires a calibration procedure before any tracking task.

C. Simulation procedure

The simulation process aims at simulating a target-tracking scenario considering all the topology parameters and factors that influence the tracking performance which have been discussed in the previous sections. As a location positioning technique the fingerprinting approach is utilized. It is based on acquiring reference points within the target area. The reference point topology is either constructed as a square grid with eligible square size, or is based on the points extracted by the quantized levels technique. Both of them were discussed in Section IV A and will be evaluated during the simulation procedure. Furthermore, in order to deal with the uncertainty of the propagated curve introduced by the first 10 meters, as shown in Section III, an additional sensor is assumed providing almost 10m range detection around a fixed node.

```
Part 1: Initialization of the  input parameters
1. Load antenna model: (InvertedF_model)
2. Define input parameters: f (working frequency), c (velocity of
   light), $\lambda$ (wave length), $\varepsilon_r$ (relative permittivity of the ground),
   $P_{TX}$ (transmission power)
3. Define model input parameters:
      (1) $H_{ref}$ : Reference Height for RSS signature definition
          (knee or ankle)
      (2) $H_r$ : Height for the fixed nodes (anchors)
      (3) $H_t$ : Height for mobile node (knee or ankle)
      (4) $H_{var}$ : Height variance due to movement
      (5) $Range_{sensor}$ : Additional sensor range detection
4. Define area input parameters:
      (1) Distance X (e.g. FieldX=50m or 70m)
      (2) Distance Y (e.g. FieldY=50m or 70m)
      (3) Number of anchors (AnchorsNumb=3)
5. Calculate anchors coordinate:
   Anchor3Y= round (FieldY*sqrt(3)/2);
   X_Anch=[0,FieldX,FieldX/2];
   Y_Anch=[0,0,Anchor3Y];
6. Define localization input parameters:
    Define number of grid points (NumbGrPoints)
    Define grid points coordinates

Calculation of the point's signature
7. for i=1:NumbGrPoints
8.    for m=1:AnchorsNumb
9.          Compute distance between anchors and the
            points: AnchDist
            (Compute  RSS)
10.         RSS = RSSmodel ($H_r$, $H_{ref}$, AnchDist, $\varepsilon_r$, $P_{TX}$, $\lambda$)
11.         Save X, Y, AnchDist, RSS  in {GridPoints}
12.         RSS_Signature (i, m) = RSS;
13    end for
14. end for

Neural Network initialization & creation
15. [rows,columns] = size (RSS_Signatures);
16. T = [1: rows];
17. net = PNN (RSS_Signatures,T);

Generate mobile node path
18. Create mobile node's path, or load mobile node's path
      as  RealCoord (X,Y)
19. Compute the number of the mobile positions,
      NodesNumb
```

Figure 15. Pseudo code for Part 1: Initialization

```
Part 2: Mobile node tracking
20. for i=1:NodesNumb
21.    Xn=NodesCoord (i, 1);
22.    Yn=NodesCoord (i, 2);
23.    HtVar=Ht+sign* $H_{var}$ *rand (1);
24.    for m=1:AnchorsNumb
25.       Compute Dist (X_Anch, Y_Anch, Xn, Yn)
26.        if  Dist < $Range_{sensor}$
27.              CloserAnchorID = m;  FlagCloserAnchor = 1
28.        end if
           (Compute RSS)
29.        RSSnoNoise = RSSmodel( Hr,HtVar, Dist,$\varepsilon_r$,$P_{TX}$,$\lambda$);
           (Include noise to RSS)
30.        RSSmeasured = noise(RSSnoNoise);
31.        MobileNodeRSS_signature (1,m)= RSSmeasured;
32.    end for
33.    [S, P] = PNNsim(net, MobileNodeRSS_signature );
       (Filter signatures-probabilities if an anchor is close)
34.    if  FlagCloserAnchor==1 (An anchor is close)
35.          [S, P] = Filter (S, P, CloserAnchorID);
36.    end if
       (Acquire the maximum probability index)
37.    index = maxProbability (P);
       (Obtain the coordinates of the estimated position)
38.    MobileNodeCoord (i,1)=GrPoints(index,1);
39.    MobileNodeCoord (i,2)=GrPoints(index,2);
40. end for
41 (Possible use of Filtering algorithm, such as Kalman
    filtering)
42. for i=1: NodesNumb
       (Compute absolute distance errors for each position)
43.    ErrorDist (i) = (MobileNodeCoord, RealCoord);
44. end for
    (Compute RMSE)
45. RMSE = sqrt (mse (ErrorDist));
```

Figure 16. Pseudo code for Part 2: Tracking

In general, the simulation procedure is divided into two parts: (1) defining the needed functions, and initialization of the input parameters and the topology of the reference points named as virtual grid points; and (2) tracking of a mobile node on preliminary generated path. The pseudo code of the simulation procedure is given in Fig. 15 and Fig. 16 for its two parts respectively and is described briefly as follows.

Part 1: From step 1 to step 5 all needed parameters concerning the deployment are initialized, where reference height for RSS signature definition is the average height obtained by the corresponding to the knee or ankle movement presented in Table I. In Step 6, the coordinates for the virtual grid points are calculated. With steps 7 to 14, for all grid points, a signature consisting of three RSS measurements from the fixed anchor nodes is calculated and stored in {*RSS_Signature*}. Steps 15 to 17 initialize and create a probabilistic neural network (PNN) representing the search and matching algorithm used by the particular fingerprinting tracking scheme. PNN combines non-parametric probability density estimation with minimum risk decision making [20]. The density estimation implements the Parzen window estimator [19] by using a mixture of Gaussian basis functions.

After the estimation of the probability density functions for all signatures in {*RSS_Signature*}, the posterior probabilities are computed, and then the Bayes' optimal decision rule is applied to select the index of the winning signature. Finally, the last steps 18 and 19 create or load the mobile node path as *RealCoord*(*X,Y*) and calculate the mobile node movement positions. This part concluded the initialization of deployment parameters, functions and grid point attributes, forming the input to part 2.

Part 2: In steps 24 to 28, for each position of the mobile node track, the distance *Dist* with respect to each of the three beacon nodes is calculated. In the steps 29 and 30, this distance is used as a parameter in the RF propagation model, which generates the respective RSS including the uncertainty due to movement variance calculated with (4). The variable height, *HtVar*, which the mobile node has in each position of the corresponding track, is calculated on the mobile node height H_t (ankle or knee) base within ±H_{var} deviation range because of the movement. In step 31, the RSS values from the three fixed nodes are stored as *MobileNodeRSS_signature* for further processing with PNN. The latter is used to discover the best matching signatures (vector *S*) from *RSS_Signatures* to the mobile node by computing their respective probability (vector *P*). In steps 34 to 36, depending on the presence of the mobile node in the detection range of the additional sensor attached to the fixed nodes, an appropriate filtering to the RSS signatures is applied selecting those corresponding to points close to a particular fixed node. Steps 37 to 39 gives the index ID of the signature with the highest probability and its corresponding coordinates, extracted from the *GrPoints* database, are coordinates for the mobile node. After the coordinates of the mobile node are identified it is possible of using filtering algorithm, such as Kalman filtering [24] in order to improve the tracking performance. Finally, the absolute distance error is computed for every mobile node position in the track as well as the root mean square error metric (RMSE) of the whole track.

D. Simulation results

The simulated area is an equilateral triangle with three fixed nodes on its vertices acquiring the RSS values of the received message from the position of the mobile node being tracked. The simulated path of the latter is comprised of points inside the triangle considering the fact that for real deployments if the mobile node inserts an adjacent triangle area, then the tracking task is undertaken by the respective fixed nodes. Moreover, for more realistic simulation over the simulated RSS, noise is added, calculated with (4), to represent the RSS variability due to human walking.

The main objective of the performed simulations is to obtain the tracking accuracy expressed by the RMSE metric in order to evaluate the selected topology parameters. The evaluation includes:

- the ankle and the knee positions' accuracy,
- the number and topology of the reference points,
- the filtering of the mobile node localization positions to perform tracking.

(a) Mobile Node Localization without Filtering

The first simulation performs localization of a mobile node located at 0.55m height in triangular area. For more realistic simulation over the simulated RSS, noise is added, calculated with (4), to represent the RSS variability due to human walking. The results from the simulation are presented in Fig. 17 for the two reference points' topology: square grid in Fig. 17(a) and quantized levels in Fig. 17(b). The located positions of the mobile node in these simulations are not filtered and it is difficult to estimate the simulated track. However, the average accuracy for both reference points' topologies is expressed as RMSE over the entire path and could be improved with any filtering algorithm. The RMSE is calculated over the entire mobile path for each point. The mobile path is calculated so as to represent walking with 1m/s. The results present that the localization accuracy is a bit better with the quantized levels reference point topology, than with the square grid topology. In this sense, the more the number of reference points increases, the less accuracy is obtained. Hence, we can conclude that without filtering algorithm the quantized levels reference point localization is better as a choice. To confirm this, more simulations are performed with square grid reference point topology for different points' number, presented in Table II. The results presented in Table II may vary within 5% from one simulation to another due to the randomness of the noise. However, the results show clearly that the number of reference points when the walking noise is included is minor factor for the accuracy of the localization process. But, if the computation time and computational resources are considered, then less points produce better performance.

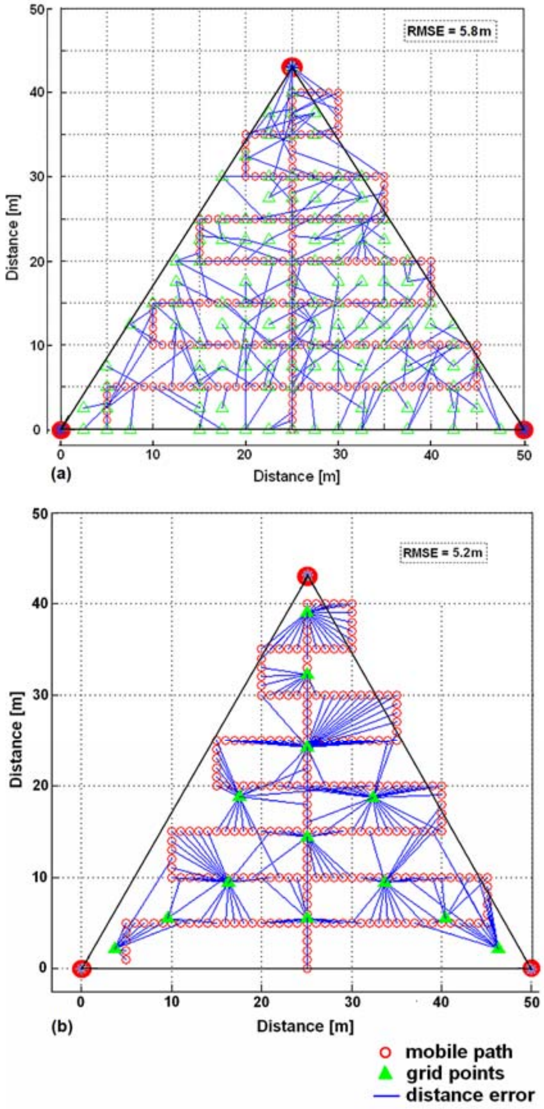


Figure 17. Tracking accuracy distribution for virtual grid density of: a) 289 points and b) 13 points

TABLE II. RMSE FOR DIFFERENT REFFERENCE POINTS NUMBER

Number of reference points	RMSE without filtering algorithm	RMSE with filtering algorithm
81	6m	2.8m
289	5.8m	2.7m
441	5.9m	2.6m
1764	6m	2.8m

(b) Mobile Node Tracking with Kalman Filtering

The second simulation repeats the first but with Kalman filtering [24, 25] performed on the localized mobile node positions, in order to complete the tracking task. The results are presented in Fig. 18 for the two reference point topologies.

The results present that the quantized-levels-reference-point topology, shown in Fig. 18(b) does not have resources for better tracking performance with Kalman filtering due to irregular location and small number of the reference points. Referring again the Table II, for reference point topology with squares' size smaller than 1/10 of the beacons' distance, i.e., more than 80 number of reference points, the tracking task can be completed through Kalman filtering with 50% improvement of the RMSE. Therefore, when tracking application is implemented through any filtering algorithm, the square reference point topology is preferable due to the uniform location and sufficient number of the reference points, unless more reference points are associated to the quantized levels topology.

(c) Ankle and Knee Positions' Accuracy

The input parameters for the last simulated scenarios are presented in Table III. The main goal is to study the localization accuracy, expressed as RMSE, depending on the mobile node positions, the beacon nodes height and the reference point topology.

According to Table III, the knee position is simulated for a 50m beacons' distance and for various fixed node heights. The signature of the reference points is calculated for the average knee position of 0.55m. However, during the simulation, the mobile node height varies from 0.45m to 0.65m, to represent the difference in the people's height. Ten simulations per each height are performed and the RMSE is averaged. The same reasoning implies also for the ankle case, the average height in which is considered as 0.20m and varies from 0.15m to 0.30m. The simulation results are presented in Fig. 19 and present the change in tracking accuracy, represented as average RMSE, for the two reference-points topologies. The average RMSE is produced by the localization procedure without Kalman filtering. The reference points are either 289 or 13 depending on the corresponding strategy used.

Fig. 19(a) and (b) present the situation when the mobile node is at knee position, i.e., 0.55m. The average RMSE is calculated for each of the allowed beacons' heights, discussed in Section II, when the heights of the mobile node

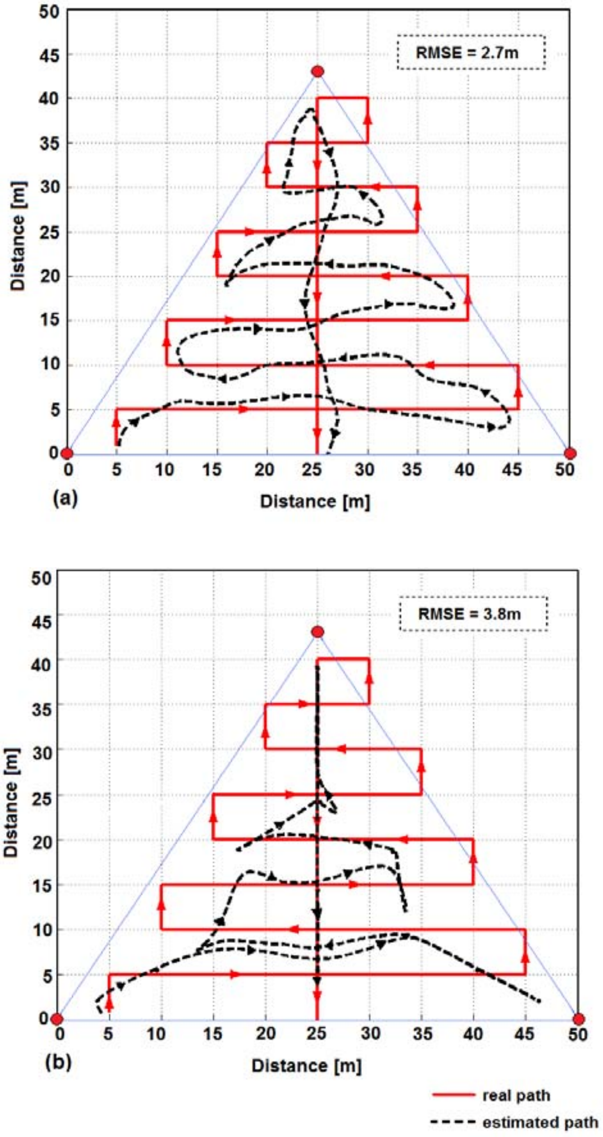


Figure 18. Tracking accuracy with Kalman filtering for reference point density of: a) 289 points and b) 13 points

TABLE III. INPUT PARAMETERS FOR THE SIMULATED SCHENARIOS

Scenario No	Mobile Node Reference Height	Fixed Node Height	Triangle Edge Size	Tx Power	Grid Points Number
1a	0.55m (knee)	0.55-1.60m	50 m	0 dBm	289
1b	0.55m (knee)	0.55-1.30m	50 m	0 dBm	13
2a	0.20m (ankle)	1.30-2.85m	50 m	0 dBm	289
2b	0.20m (ankle)	1.25-2.75m	50 m	0 dBm	13

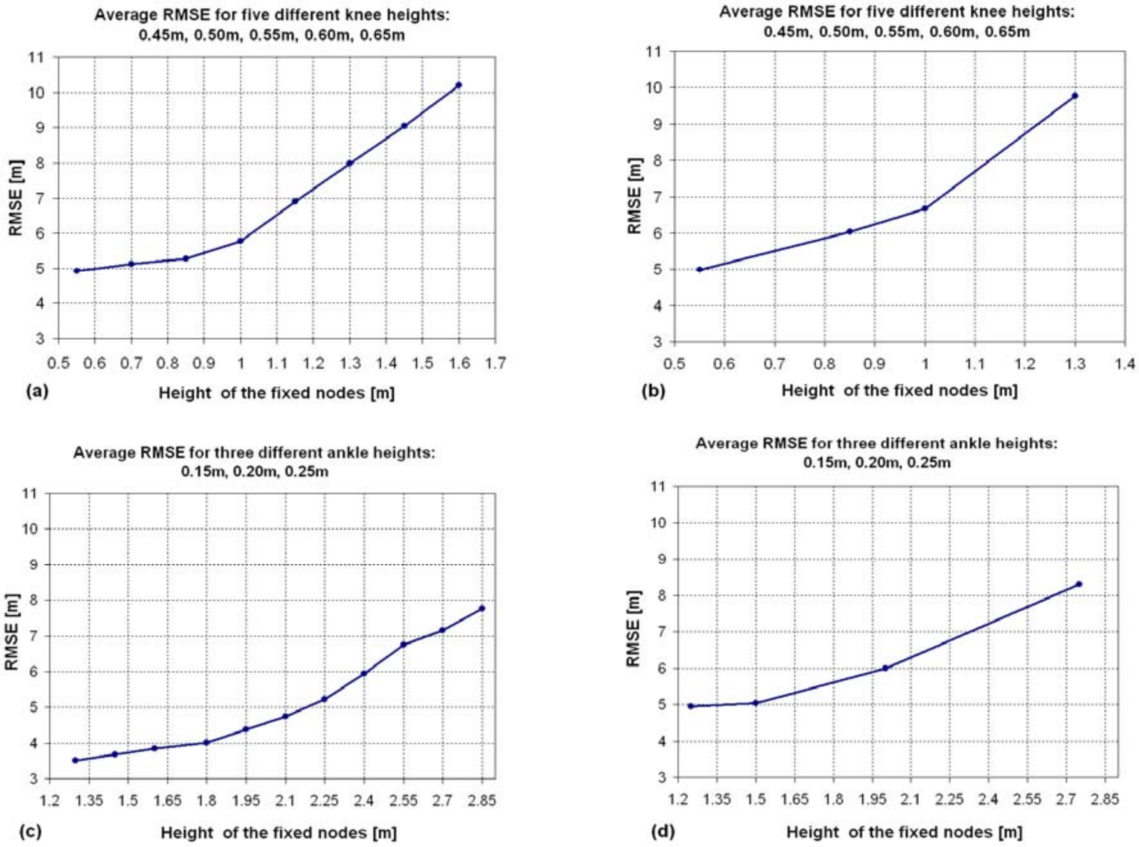


Figure 19. Tracking simulation RMSE distance based on different mobile node position on human body and number of RSS fingerprinting signature points: a) knee / 289, b) knee / 13, c) ankle / 289, and d) ankle / 13

change from 0.45m to 0.65m. Fig. 19(a) presents the square grid topology with 289 reference points, while Fig. 19(b) presents the quantized levels topology with 13 reference points. As it is shown, there is an increase in the loss of accuracy as the fixed node height increases. This observation stands for both the fingerprinting schemes of different reference point numbers having also similar accuracy values. As a conclusion, if the beacons' height is between 0.55m and 1m, then the localization accuracy is within about 6m for both the reference points' topologies.

In the second case, the ankle position is simulated in the same triangular area when the height variation of the mobile node ranges from 0.15m to 0.25m, while the beacons' node height ranges from 1.25m to 2.85m, according to the results from Section III B. The results are presented in Fig. 19(c) for square reference point topology and Fig. 19(d) for quantized levels topology. As in the knee case, the results for the two reference points' topologies present similarities. However, the case of the square grid topology with 289 reference points presents smaller RMSE than the case with the quantized levels topology with 13 reference points. Nevertheless, analyzing both topologies results, the obvious conclusion is that the RMSE is noticeably smaller when the beacons' height is in the range of 1.25m-1.65m.

V. Practical Tracking Experiment

In this section, a real experiment is presented keeping the proposed topology parameters such as beacons' height and distance, and knee or ankle position for the mobile node, so as to perform a real outdoor tracking test.

A. *Experimental Setup*

From experiments performed for defining the variance of the RSS due to the body movement, we conclude that if the mobile node has not line-of-sight communication with beacon, for instance if it is attached to the inner side of a leg, the RSS drops about 20dBm compared to the value taken when the node is attached to the outer side of a leg. To overcome this problem, we attach one sensor node to each knee, as it is shown in Fig. 20. Thus, we take measurements from both mobile nodes and keep the strongest ones for the tracking task. The real results are much worse than the simulated ones. According to our understanding, this happens due to the offset that each node, mobile or beacon, imposes on the RSS measurements, and also due to rotation of the antenna, i.e., the mobile node and the non-equivalent antenna radiation in all directions.

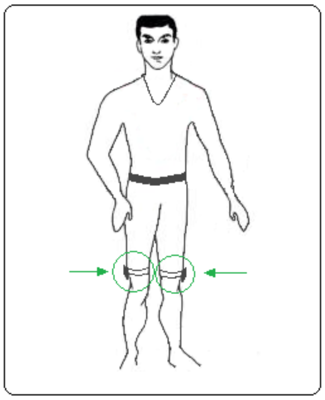


Figure 20. Positioning of two mobile nodes on the human body

The start-up requirements are:

- beacon nodes' distance of 45m,
- beacon nodes at 0.7m height,
- two mobile nodes (one at each knee) at 0.55m,
- walking with approximate speed of 1m/s,
- seven different measurements on the same walking trajectory,
- calibration measurements.

B. Measuring Process

During the tracking measurements a person with attached mobile nodes to his knees walks on preliminary drawn trajectory with speed of approximately 1m/s. The two mobile nodes send, one by one, a packet every second to the beacon nodes. The beacons send, within the time frame left of one second, their messages to the base station, including the RSS of the mobile nodes' packets.

After the tracking measurements, we perform also calibration measurements to evaluate the variability concerning the node hardware. The beacon nodes and the mobile nodes are located at 0.7m height and at distance of 20m. The two mobile nodes send 150 packets to the three beacons. From the measurements the average RSS value is calculated for each beacon. The difference between this value and the value produced through simulation with the propagation model (1) at 20m distance and 0.70m height gives the offset which can be included in (5). In addition, we compare the RSS measurements taken from each beacon during the walking with the RSS generated for the same node with the simulator (including the walking noise) to evaluate the offset. Finally, the results have shown that the beacons 1 and 3 have offset of about 5dBm regarding the ideal simulation and beacon 2 about 10dBm.

According to the specific results, an experimental system is not ready for real time filtering and tracking without dealing first with the calibration problems. Because of the hardware difference and the antenna radiation asymmetry the development of a generic calibration algorithm is extremely challenging. Therefore, the parameters for the calibration in the particular experiment have been calculated post-process.

C. Kalman Filtering

The next step is to apply Kalman filtering on the localized mobile coordinates in order to perform tracking. In order to use the standard Kalman filter to estimate a position, the process under consideration has to be able to be described by linear system equations. A linear system is a process that can be described by the following equations [25]:

$$x(k+1) = F\,x(k) + w(k)\,, \tag{6}$$

$$y(k) = H\,x(k) + v(k)\,, \tag{7}$$

Equation (6) is a state equation, while (7) is an output equation.
Here $x(k)$ is state vector and consists of the x, y positions and the corresponding velocities, $w(k)$ is the process noise with normal probability $p(w)$~$N(0,Q)$, $v(k)$ is the measurement noise with normal probability $p(v)$~$N(0,R)$, F is state transition matrix, H is observation matrix, Q is process noise covariance matrix and R measurement noise covariance matrix.

The implemented Kalman filter for the purpose of this work is based on the Kevin Murphy's toolbox [25]. The input parameters and the initialization of the needed matrices for the Kalman filter are shown in Fig. 21. A brief description of the code is as follows:

From Step1 to Step10 the state matrices are initialized. Step 13 and Step 14 calculate the process noise and measurement noise covariance matrices. Usually the process noise represented as *varianceQ* is small with values 0.1 or 0.01. The value *varianceR* represents the uncertainty in the measurement process and we accept that *varianceR* is 6 or 7, representing 6m/7m of measurement error. The matrices Q and R may change with each time step or measurement. Step 15 and 16 call the kalman_filter and kalman_smoother functions, which are from the Kevin Murphy's toolbox, and return the filtered coordinates as *xfilt*, *Vfilt* and *xsmooth*, *Vsmooth,* respectively. The new coordinates are saved in *Kalman_localized* and *Kalman_smooth* matrices during the Step 17 to Step 20. Step 21 plots the result and from Step 22 to Step 32 the RMSE of the results for pure localization, the filtering and the smoother are calculated.

D. Tracking Results

As it was mentioned above, seven different measurements on the same walking trajectory have been performed. The calibration of the beacons are fulfilled through comparison of the RSS measurements taken from each beacon during the walking (only for one measurement) with the RSS generated for the same node with the simulator (including the walking noise) to extract the offset for each beacon.

The tracking test has been performed for the two reference point topologies with the same input measurements and calibration values. For the square grid topology, 289 points are generated. For the quantized levels, this time, more regions are identified and consequently the number of the reference points is 19, instead of 13 in the simulation example. The calculated RMSEs for all seven measurements,

```
Kalman filtering

(input matrixes: NodesCoord_real and  NodesCoord_localized)
X0=10;              %  X beginning coordinate
Y0=0;               %  Y beginning coordinate
Vx0=0;              %
Vy0=1;              %
ss = 4;             %  state size
os = 2;              %  observation size
x = zeros (ss, T);  % T -number of the measurements
y = zeros (os, T);
initx = [X0 Y0 Vx0 Vy0]';
initV = 1*eye (ss);
F = [1 0 1 0; 0 1 0 1; 0 0 1 0; 0 0 0 1];
H = [1 0 0 0; 0 1 0 0];
Q = varianceQ *eye (ss);      % Process noise covariance
R = (varianceR)^2*eye(os); % measurement noise covariance
[xfilt, Vfilt, VVfilt, loglik] =
               =kalman_filter(y, F, H, Q, R, initx, initV);
[xsmooth, Vsmooth] =
               =kalman_smoother(y, F, H, Q, R, initx, initV);
Kalman_localized(:,1)= xfilt(1,:)';
Kalman_localized(:,2)= xfilt(2,:)';
Kalman_smooth(:,1)= xsmooth(1,:)';
Kalman_smooth(:,2)= xsmooth(2,:)';
plot(Kalman_smooth(:,1), Kalman_smooth(:,2),'ks');
 yR=[NodesCoord_real(:,1), NodesCoord_real(:,2)]';
y=[NodesCoord_localized(:,1), NodesCoord_localized(:,2)]';
allsr=length(NodesCoord_localized(:,1));
alls=length(Kalman_localized(:,1));
dReal = yR([1 2],:) - y([1 2],:);
rmse_real = sqrt(sum(sum(dReal.^2))/allsr);
dfilt = yR([1 2],:) - xfilt([1 2],:);
rmse_filt = sqrt(sum(sum(dfilt.^2))/alls)
dsmooth = yR([1 2],:) - xsmooth([1 2],:);
rmse_smooth = sqrt(sum(sum(dsmooth.^2))/alls)
```

Figure 21. Tracking with Kalman filter

for the two reference points' topologies are presented in Table IV. Two columns for each topology case are given to illustrate the difference in the RMSE when only localization of the node is required and when the tracking with filtering is implemented. As it is obvious, the filtering improves the accuracy. Another observation is that the quantized levels topology presents a bit better results most likely due to the careful selection of the position of the reference points. Visually there is no big difference between the tracked path from the one or the other reference point topologies, as it is shown in Fig. 22. Fig. 22(a) presents the real path and the tracked path from measurement 4 with square grid topology, while Fig. 22(b) presents the real path and the tracked path from measurement 2 with quantized levels topology.

The RMSE of all seven measurements, for the two reference point's topologies, presents relatively stable tracking accuracy, which is within the application requirements of 10m. However, without proper topology parameter selection and proper calibration procedure, the RSS-based tracking is impossible.

TABLE IV. RMSE FOR SEVEN MEASUREMENS

Number	**RMSE Square grid, 289 points**		**RMSE Quantized levels, 19 points**	
	Localization	**Filtering**	**Localization**	**Filtering**
1	13.2m	10.8m	11.8m	10m
2	13m	9.5m	10.3m	8.2m
3	12.3m	9.2m	10.9m	8.5m
4	10.8m	8.8m	9.4m	7.6m
5	11.2m	7.6m	9.3m	7m
6	12.3m	9.7m	11.3m	9m
7	12m	10m	10.8m	8.9m

VI. DISCUSSION AND CONCLUSIONS

The present study investigates the possibility a target tracking task to be performed with the resources of the WSN technology, when using only the RSS of the exchanged messages. In order to evaluate such a possibility the present study was focused on identifying the most crucial RSS-based tracking problems and to determine and evaluate the topology parameters that can guarantee successful tracking. More specifically we aimed at: (1) evaluation of the behaviour of the propagated RF signal in order to identify the important RSS-based tracking problems, (2) selection of the topology related parameters, when taking into consideration the tracking task requirements and the deployment constraints, (3) use of the selected propagation model and topology parameters to simulate a target tracking task, and (4) use of the proposed topology parameters to perform real outdoor tracking test. Hence, summarizing the obtained results the following conclusions were derived:

- During the modelling of the RF signal propagation several factors have to be considered, such as the height from the ground of the transmitter and the receiver, the frequency of the transmitted signal, the type of the ground (e.g. grass, road, etc.), the sensor nodes' hardware variability, the difference of the antenna gain for each of the supported frequency bands, the antenna pattern irregularities, and the RSS uncertainty.
- The RF signal propagation has to be considered when choosing the topology parameters, such as beacons' distance and height, in order to assure reliable communication between the beacons. The combination between the fixed node's height and distance is constrained by the fact that the communication among them has to be as robust as possible with respect to minimum packet loss. Therefore, the selected heights of the fixed nodes for reliable communication among them at distance, for instance of 50m are: 0.55m–1.60m; 1.85m–2.35m; and 2.55m–2.90m.
- The position of the mobile node on the human body should be selected so as to minimize the possibility for great node's height variation due to body movement in order to minimize the variability on the RSS due to this movement.

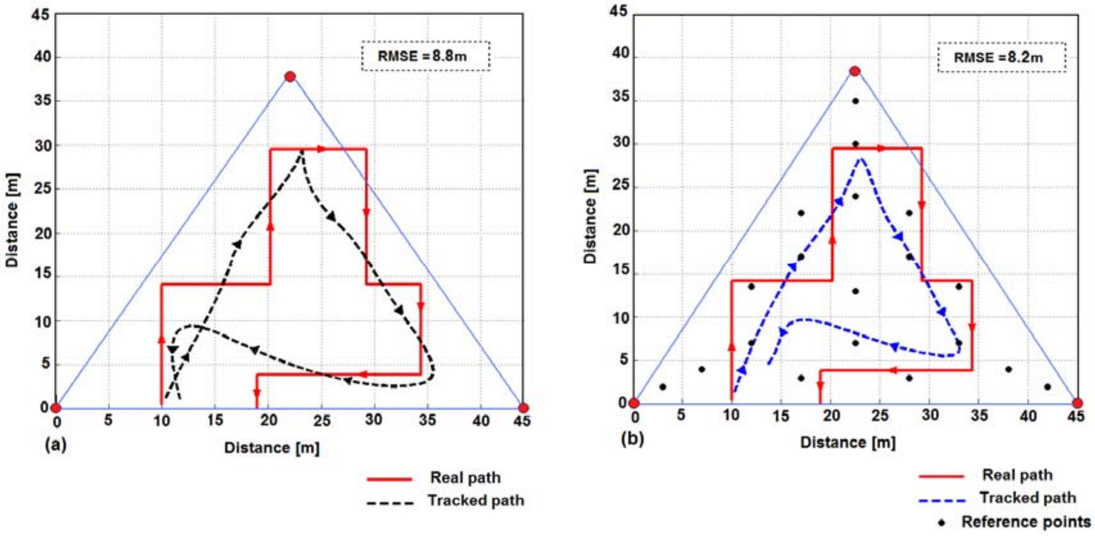


Figure 22. Tracking of a mobile node with Kalman filtering for:
(a) square-grid reference point topology with 289 points and (b) quantized-levels reference point topology with 19 points

- The RF propagation between the beacons and the mobile node has to be as smooth as possible to ensure that it has as few as possible ‘nulls’. This can be achieved by judicious selection of the heights at both beacons and mobile node.
- When using the fingerprinting approach, the selection of the reference points’ locations and topology needs preliminary evaluation according to well defined criteria such as the number of points, which define the computational load and the granularity of the localization accuracy as well, the required tracking accuracy, and finally, the simplicity of points generation, where the square grid is easier to be generated, while the quantized levels approach needs multiple calculations for each case of beacon - mobile node pair of heights.
- The hardware variability strongly requires calibration procedure before any tracking task. Because of this, the antenna radiation asymmetry and the possibility of unpredictable and continuous rotation of the mobile node, the development of generic calibration algorithm is extremely challenging. Without proper calibration procedure the RSS-based tracking is impossible.
- The filtering, such as the Kalman filtering, applied on the localized mobile node coordinates is absolutely essential for performing tracking, but it is not necessary when only localization is performed.
- The real outdoor tracking test demonstrates that the RSS can be used for outdoor localization and tracking applications under well-defined topology constraints and only after the proper calibration.

ACKNOWLEDGEMENT

The work reported here was performed as part of the uSWN FP6-2005 IST-034642 research Program and funded by the European Social Fund (ESF).

REFERENCES

[1] F. Kerasiotis, Ts. Stoyanova, A. Prayati, G. Papadopoulos, “A Topology-Oriented Solution Providing Accuracy for Outdoors RSS-Based Tracking in WSNs”, Second International Conference on Sensor Technologies and Applications, SENSORCOMM 2008, August 2008, Cap Esterel, France, pp. 239–245 doi: 10.1109/SENSORCOMM.2008.122

[2] M. Cardei, J. Wu, “Energy-efficient coverage problems in wireless ad-hoc sensor networks”, Computer Communications 29, pp. 413–420, Elsevier B.V, doi:10.1016/j.comcom.2004.12.025 2006.

[3] A. Savvides, C. Han, M. B. Srivastava. “Dynamic fine-grained localization in ad-hoc networks of sensors”, 7th annual international conference on MobiCom 2001, Rome, Italy, 2001, pp. 166–179, doi: 10.1145/381677.381693

[4] D. Niculescu, B. Nath, “DV Based Positioning in Ad hoc Networks”, Journal of Telecommunication Systems, Springer, Vol.22, Numb.1-4, January, 2003, pp. 267-280, doi: 10.1023/A:1023403323460.

[5] M. L. Sichitiu, V. Ramadurai, P. Peddabachagari, "Simple algorithm for outdoor localization of wireless sensor networks with inaccurate range measurements”, International Conference on Wireless Networks, 2003, pp. 300-305.

[6] T. He, C. Huang, B.M. Blum, J.A. Stankovic, T. Abdelzaher, “Range-free localization schemes for large scale sensor networks”, 9th annual international conference on MobiCom 2003, San Diego, CA, USA 2003, pp. 81–95, doi: 10.1145/938985.938995.

[7] F. Reichenbach, J. Blumenthal, D. Timmermann, ”Improved Precision of Coarse Grained Localization in Wireless Sensor Networks” 9th Conference on DSD 2006, , Dubrovnik, Croatia, 2006 pp. 630-637, doi: 10.1109/DSD.2006.61.

[8] Hu, L., Evans, D., “Localization for mobile sensor networks”, 10th Annual International Conference MOBICOM 2004, ACM, Philadelphia, US, 2004, pp. 45 - 57, doi: 10.1145/1023720.1023726.

[9] Y.-C. Tseng, S.-P. Kuo, H.-W. Lee, and C.-F. Huang. "Location tracking in a wireless sensor network by mobile agents and its data fusion strategies", 2nd International Conference on Information Processing in Sensor Networks, IPSN 2003, Palo Alto, CA, USA, 2003, pp. 625–641.

[10] K. Mechitov, S. Sundresh, Y. Kwon, "Cooperative Tracking with Binary-Detection Sensor Networks", 1st International Conference on Embedded Networked Sensor Systems, SenSys 2003, Los Angeles, California, USA, November 5-7, 2003, pp. 332-333.

[11] H. Yang, B. Sikdar, "A Protocol for Tracking Mobile Targets using Sensor Networks", Proceedings of IEEE Workshop on Sensor Network Protocols and Applications, WSNA 2003, Anchogare, AK 2003.

[12] Ts. Stoyanova, F. Kerasiotis, A. Prayati, G. Papadopoulos, "Evaluation of Impact Factors on RSS Accuracy for Localization and Tracking Applications in Sensor Networks", Special issue of Telecommunications Systems Journal, Springer, Vol.42, pp. 235-248, Dec. 2009, DOI: 10.1007/s11235-009-9183-8)

[13] T. S. Rappaport, Wireless Communications: Principles and Practice, 2nd Edition. Prentice Hall, 2001

[14] Ultra low power IEEE 802.15.4 compliant wireless sensor module, TmoteSky datasheet, Moteiv Co., 2006

[15] A. Smith, H.Balakrishman, M. Goraczko, N. Priyanha, "Tracking moving devices with cricket location system", In Proc. 2th ACM MobiSys, Boston, MA, 2004, pp. 190-202, doi: 10.1145/990064.990088

[16] C. Alippi, G. Vanini, "A RSSI-based and calibrated centralized localization technique for Wireless Sensor Networks", In Proc. 4th annual IEEE International Conference on Pervasive Computing and Communication Workshops 2006, PERCOMW'06, pp.301-305, doi: 10.1109/PERCOMW.2006.13

[17] Jian Ma, Quanbin Chen, Dian Zhang, Lionel M. Ni, "An empirical study of radio signal strength in sensor networks in using MICA2 nodes," HKUST, Technical Report, 2006

[18] Ts. Stoyanova, F. Kerasiotis, A. Prayati, G. Papadopoulos, "A Practical RF Propagation Model for Wireless Network Sensors", In Proceedings of the 3d International Conference on Sensor Technologies and Applications, SENSORCOMM 2009, June 2009, Athens, Greece , doi: 10.1109/SENSORCOMM.2009.39

[19] E. Parzen, "On estimation of a probability density function and mode.", Annals in Mathematical Statistics, 33:1065-1076, 1962

[20] D. F. Specht, "Probabilistic neural networks." , Neural Networks, 3(1), pp:109-118, 1990

[21] "TelosB mote platform", Crossbow Technology. Copyright 2009, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf

[22] Ts.Stoyanova, F. Kerasiotis, A. Prayati, G. Papadopoulos, "Communication-aware deployment for Wireless Sensor Networks", SENSORCOMM 2008, August 2008 - Cap Esterel, France.

[23] Ts.Stoyanova, F. Kerasiotis, K. Efstathiou, G. Papadopoulos, "Modeling of the RSS Uncertainty for RSS-based Outdoor Localization and Tracking Applications in Wireless Sensor Networks" SENSORCOMM2010, July 2010, Vinecia, Italy, (accepted for publication).

[24] Welch, Bishop, "An Introduction to the Kalman Filter", UNC-Chapel Hill, TR 95-041, July 24, 2006, available from: http://www.cs.unc.edu/~welch/kalman/

[25] K. Murphy, "Kalman filter toolbox for Matlab" 1998 , available from: http://www.cs.ubc.ca/~murphyk/Software/Kalman/kalman.html

Deployment Considerations for Reliable Communication in Wireless Sensor Networks

Tsenka Stoyanova, Fotis Kerasiotis, George Papadopoulos
Applied Electronics Laboratory, Department of ECE, University of Patras, 26500 Rio-Patras, Greece
Industrial Systems Institute, Stadiou Str., 26504 Platani, Patras, Greece
{tsstoyanova, kerasiotis, papadopoulos}@ece.upatras.gr

Abstract — **When deployed in the real-world, many sensor nodes fail to deliver the expected data or deliver a small percentage of them. The reasons for such failures can be hardware problems, software problems or communication problems. Among the communication problems is the insufficient received signal strength (RSS), which may cause unsuccessful packets delivery. For that reason, during the pre-deployment phase the RF propagation must be considered for predicting the RSS so that it guarantees reliable connectivity level. The present work is devoted to study of the deployment factors and requirements, which can ensure reliable communication links among the sensor network nodes. Subsequently, they are considered by the proposed RF signal propagation-based connectivity algorithm (RFCA). RFCA utilizes an outdoor RF signal propagation model for predicting the RSS in the positions, where sensor nodes are supposed to be deployed. Several factors are considered during RSS prediction, namely RF frequency, transmission power, transmitter–receiver distance, height from the ground and antenna's characteristics (gain, polarization, orientation, etc.). Different outdoor propagation models are discussed and analyzed with respect to their applicability to RSS prediction. Finally, an example illustrates that RFCA is able to find the most appropriate, from the communication point of view, deployment parameters (height, distance, and transmission power) for positioning the sensor nodes in outdoor environment.**

Keywords-WSN deployment; wireless communication; RSS; signal propagation model

I. Introduction

Sensor nodes' deployment reflects two main aspects of a wireless sensor network (WSN), namely sensing and communication. The deployment process can be considered as consisting of three consequent phases: pre-deployment, actual deployment and post-deployment. The pre-deployment phase includes analysis of the application requirements and of the area environment followed by deployment simulation and planning in order to ensure WSN requirements such as coverage, connectivity, optimal energy budget and low packet loss in the physical layer.

Several deployment algorithms propose solutions for the coverage problems when minimizing the number of deployed nodes [2] and [3] and the connectivity problems with respect to the optimal number of neighbors [4]. However, one communication aspect is not addressed adequately as a deployment prerequisite and that is the RF signal propagation. The knowledge of radio wave propagation and its modeling is essential for deploying a WSN in such a way to ensure reliable communication among the network nodes. A communication link is considered as guaranteed when the received signal strength (RSS) is sufficient.

Different RF propagation models have been introduced in the literature for supporting the wireless communication system design [5-7]. Most of these propagation models have been studied for high-power wireless communication systems, which operate at distances in the range of kilometers and they are not directly applicable to WSNs. However, some of them still could be used in RSS prediction with justification of the environment and conditions for their use. In general, the RF propagation is environment-dependant and the choice of an appropriate propagation model, for a specific environment, is crucial for the success of any RSS prediction.

For predicting the RSS, several parameters of the system have to be considered: the distance between transmitter–receiver pair (T-R), their height from the ground, the antenna's characteristics (gain, polarization, etc) and the terrain specifics.

In some studies, involving WSN, the deployment and the RF propagation are not discussed at all as a factor influencing the communication link between two sensor nodes, but just mentioned that the communication link is good within certain distance. In others [9], [10] and [11] the received power is modeled with log-normal path loss model, which model does not include the influence of the T-R heights and the ground reflection over the RF signal propagation. Such assumptions lead to the common belief that the communication between two sensor nodes is good until certain distance and worsen after it. As we will discuss in the fallowing sections there are several factors that influence the successfulness of the wireless communication and they have to be considered during the pre-deployment planning.

This work discusses the deployment factors that are of importance for ensuring reliable communication among the sensor nodes. It elaborates on the previous work [1], where the RF signal propagation-based Connectivity Algorithm (RFCA) is discussed. RFCA incorporates the signal propagation model and the deployment factors into the pre-deployment WSN planning. It utilizes a RF signal propagation model to predict the RSS within the radio ranges in order to identify the most appropriate communication-based deployment parameters, i.e. T-R distance, height from the ground, and transmission power. It consists of four sequential steps: (1) distance and height prediction; (2) transmission power simulations; (3) non-neighbor interference minimization; (4) deriving the optimal deployment parameters. RFCA is generic enough to be combined with any number-of-nodes optimization algorithm like [2-4] and [8].

In the present work, we motivate the importance of using proper simulation model for a specific environment. For that reason, four propagation models are overviewed and taxonomy for their use in outdoor environments is offered. Real outdoor measurements are compared with the simulation results for verification of the proposed RF propagation models.

The remaining of this paper is organized as follows. In Section II, the pre-deployment simulation framework is outlined and RFCA is presented. Section III presents a detailed description and analysis of the RFCA. Furthermore, an overview of some outdoor propagation models is presented together with simulated and measured results. A classification of the outdoor environments and the corresponding propagation models is also given. In Section IV, the RFCA reasoning is illustrated with an example for an environmental monitoring application scenario in an unobstructed environment. Finally, Section V concludes this work.

II. Pre-Deployment Simulation Framework

The pre-deployment simulation aims at analyzing the application requirements against the deployment environment to satisfy the sensor network's prerequisites for degree of coverage, complete network connectivity, optimal energy budget and low percentage of packet loss due to the physical layer. Therefore, the pre-deployment simulation framework includes three basic simulations: sensing coverage, communication connectivity and in-network localization, as shown in Fig. 1.

The sensing coverage simulation aims at ensuring the application requirement for coverage with the optimal number of sensor nodes. This component provides, along with the number of nodes for the given area, information about the possible heights from the ground of the sensor nodes and the minimum and maximum distance boundaries.

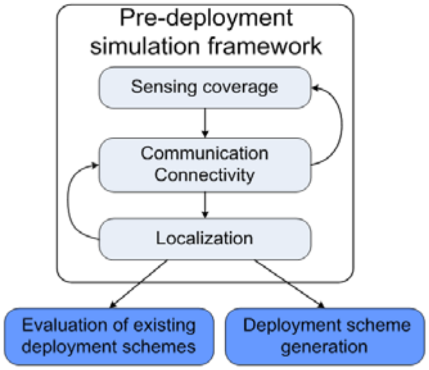

Pre-deployment simulation framework
Sensing coverage
Communication Connectivity
Localization
Evaluation of existing deployment schemes
Deployment scheme generation

Figure 1. Pre-deployment framework diagram

The communication connectivity simulation extends the simulation process by matching the preferred nodes height and distance with ones proposed by the signal propagation model. The most appropriate height and distance are the output from this component along with suggestions for reducing the transmission power. The last simulation is localization, which uses information about the number of nodes and preferred topology to simulate the nodes localization process based on already known parameters as height, distance and signal propagation. This step may also suggest correction of the nodes height and distances for better localization accuracy. The pre-deployment simulation framework can be used in two directions: (1) to evaluate quality provisioning of existing deployment topologies, and (2) to propose a deployment scheme based on input parameters.

This work focuses mainly on the RFCA as the communication connectivity component, in which the deployment considerations for reliable communication are discussed and implemented.

III. RF Signal Propagation-Based Connectivity Algorithm

In the literature the communication radius is often assumed to be at least twice as the sensing radius to support coverage with minimum number of nodes and complete communication connectivity [6, 7]. In some cases, this assumption might not be valid. For instance, given that the passive infrared (PIR) sensing radius is 10m, then the communication radius should be at least 20m according to the above-mentioned assumption. In this case, distance of 20m cannot be reached, with sufficient RSS level, by sensor nodes such as Tmote Sky [14] and TelosB [15] with internal antenna only, even at the maximum transmission power of 0dBm, if they are placed horizontally on the ground. Therefore, the RF signal propagation and certain deployment factors have to be considered and to go along with the optimal sensing coverage and localization algorithms when a WSN is designed.

A. *Algorithm Overview*

In order to incorporate the signal propagation and the deployment factors into the deployment planning an algorithm named RFCA was developed. Fig. 2 presents its block-diagram. RFCA consists of four steps: (1) Step 1 - distance and height prediction: It aims at discovering the most appropriate heights and distances of the sensor nodes, based on the input parameters; (2) Step 2 - transmission power simulations: Different levels of the transmission (Tx) power are simulated to evaluate the possibility of reduction; (3) Step 3 - non-neighbor interference minimization: In this step simulations are performed to evaluate the RSS level of the non-neighbor nodes in order to discover the best combination of distance, height and Tx power to minimize the interference from the unwelcome nodes; (4) Step 4 - Deriving the optimal deployment parameters: It aims at summarizing the results and at proposing the most appropriate deployment parameters according to initial criteria.

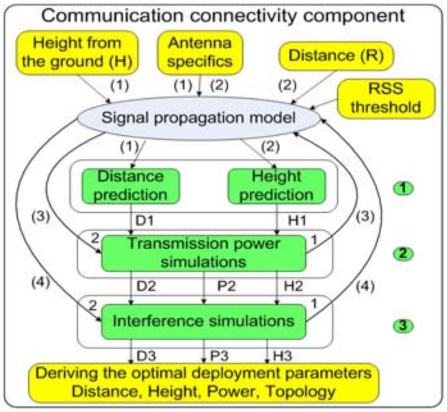


Figure 2. RFCA block diagram

RFCA is applicable for outdoor applications, for manual and for random deployments:

- For manual deployment when one parameter (height or distance) is known, the simulation aims at predicting the other parameter to guarantee sufficient RSS for deployment. For example, path (1) of Fig. 2 uses as input parameter the height from the ground and calculates the appropriate distance for nodes' positioning.
- For random deployment scenario (nodes thrown out from a airplane) the nodes end up on the ground and the height of the antenna is between 2cm and 10cm. Based on the propagation model, on Tx power and on the node's antenna specifics, it is possible to predict the maximum distance, where the signal has sufficient RSS level to guarantee successful communication. Considering this distance as maximum communication radius, the necessary number of nodes per unit area may be calculated to ensure network connectivity and certain degree of coverage.

B. *Algorithm attributes*

The factors that are of importance for successful physical communication are inputs parameters for the RFCA and are discussed in the following Section. As shown in Figure 2, RFCA takes as input height from the ground or the transmitter–receiver (T-R) distance and the RSS threshold. The antenna specifics (gain, polarization orientation) and propagation model need also be known in advance.

1) RF propagation models, height from the ground and distance: The height from the ground and distance between any two sensor nodes are the most important parameters influencing the RSS. These two parameters participate directly in the propagation model equation. The relation between distance, height and RSS forms the propagation of the RF signal.

Many different RF propagation models have been introduced for supporting the wireless communication system design. Most of these propagation models have been studied for high-power wireless communication systems like those for UHF/VHF band, satellite, cellular, etc. [5 - 7]. The assumption made by these models does not take into account some of the most distinctive features of WSNs, i.e. low-power radio, the antenna low height and the feature of omnidirectionality, which render the models unsuitable for WSNs. However, some of them still could be used in RSS prediction when justifying the environment and conditions under use. In general, the RF propagation is environment-dependant and the choice of an appropriate propagation model, for a specific environment, is crucial for the success of any RSS prediction

In the following some propagation models are overviewed and taxonomy for their use in outdoor environments is offered, in conjunction with their dependence on the height.

a) Log-normal Path Loss Model (LPLM): LPLM [5] considers the received power as a function of the T-R distance raised to some power. Since this model is a deterministic propagation model and gives only the average value, another propagation model, known as log-normal shadowing model, defined by (1), was introduced to describe the RSS irregularity [5]. The received power, $P_r(d)$, at distance d is given as:

$$P_r(d)[dBm] = P_0 - 10n\log_{10}\left(\frac{d}{d_0}\right) + X_\sigma \qquad (1)$$

where X_σ is a Gaussian random variable with zero mean and standard deviation σ, P_0 is the received power at reference distance d_0, and n is the path loss exponent factor. The path loss exponent n is environment-dependant. In general n is 2 in free space model. However, in indoor obstructed environment n may take values between 4 and 6, and for outdoor obstructed – values between 3 and 5 [5].

$P(d_0)$, n and σ describe statistically the path loss model for any location in specific T-R distance. This model is used for computation and simulation of the received power at random locations. In Fig. 3(a) the LPLM model with n=3 is presented.

b) Free-space + Two-ray Ground Reflection Model (FS+GR).

LPLM is actually a simplistic approach as it does not allow obstructions to be taken into account. The greatest and unavoidable obstruction, when the wireless nodes are placed close to the ground is the earth surface. In order to model the ground influence on the signal, the basic Free Space + Ground Reflection (FS+GR) model equation of [5] is adopted for recalculating by including the ground reflection coefficient with its formulation given also in [5]. Thus, the final equation for $P_r(d)$ is given as:

$$P_r(d) = P_r(d_0) + 20\log_{10} d_0 + 10\log_{10}\left(\left(\frac{\cos\Delta\theta}{L_D} - \frac{\Gamma}{L_R}\right)^2 + \left(\frac{\sin\Delta\theta}{L_D}\right)^2\right) \qquad (2)$$

where L_D and L_R are the path length of the directed and the path length of the reflected signal, $\Delta\theta$ is phase difference, Γ is the reflection coefficient, which depends on the polarization of the radio waves given in [5] as:

$$\Gamma(\theta)=\frac{\sin(\theta)-X}{\sin(\theta)+X} \tag{3}$$

- for perpendicular polarization: $X_{\perp}=\sqrt{\varepsilon_r-\cos^2(\theta)}$
- for parallel polarization: $X_{\parallel}=\frac{\sqrt{\varepsilon_r-\cos^2(\theta)}}{\varepsilon_r}$

where θ is the angle of incidence and ε_r is the complex permittivity for lossy reflecting surface [5]:

$$\varepsilon_r=\varepsilon_1-j60\sigma_1\lambda \tag{4}$$

where λ is the wavelength, ε_1 is the relative dielectric constant and σ_1 is the conductivity of the reflecting surface material in S/m.

The two-ray ground reflection model is useful to predict the RSS variance concerning the ground reflection over distances. It also could be used to calculate the most appropriate height from the ground of the T-R pair for certain T-R distance.

c) Free-space Outdoor Model (*FOM*): This model was introduced firstly in our previous work [13]. Various RF signal-influencing factors, such as free-space path loss, ground reflection path loss, RSS uncertainty, variation of the transceivers, radio frequency gain inequality and antenna pattern irregularity are included in the model. The FOM formulation is given as :

$$\overline{P_R}(d)=P_T\left(\frac{\lambda}{4\pi d}\right)^2\left(K_1^2+K_2^2\Gamma^2+2K_2\Gamma\cos\left(\frac{2\pi}{\lambda}\Delta L\right)\right) \tag{5}$$

where $\overline{P_R}(d)$ is the average value.

If P_R is expressed in decibels with RSS uncertainty included as $X_{\sigma(\overline{P_R})}$, then the formulation is:

$$P_R=\overline{P_R}(d)+X_{\sigma(\overline{P_R})} \tag{6}$$

where P_R is the received power, P_T is the transmission power, d is T-R distance, λ is the wavelength, Γ is the ground reflection coefficient given with (3), ΔL is path length difference between the direct and the reflected signals, coefficients K_1 and K_2 are antenna specifics representing the gain in particular antenna orientation. The RSS uncertainty is given as a Gaussian random variable X with distribution $\sigma(\overline{P_R})$ [10]. A simulation performed with (5) is presented in Figure 3(b).

d) Tree-obstructed Outdoor Model (TOM): In a tree-obstructed environment the trees can be located in any order -random, line, or grid. The impact factors of the RF signal propagation through forest environment, apart from the common ones like height from the ground, T-R distance, ground reflection, antenna radiation pattern, and RSS variance, are also the trunk and vegetation diffraction and scattering.

For sparse tree environment the base model could be FOM (equation (6)) with counting additionally the single-tree effects over the RF signal such as: trunk diffraction and vegetation scattering. Considering the total propagation loss in decibels as $L_{TOT}= L_M + L_{veg}$, where L_M is the basic signal propagation loss in the media and L_{veg} is the propagation loss because of vegetation, we recalculate the P_R trough L_M and L_{veg}. Consequently, the final equation for the received power P_R in decibels is:

$$\begin{aligned}&P_R[dBm]=P_T[dBm]-L_{TOT}[dBm]=\\&=P_T-L_M-L_{veg}=P_{R_M}-L_{veg}\Rightarrow\\&\Rightarrow P_R[dBm]=P_{R_M}[dBm]-L_{veg}[dBm]\end{aligned} \tag{7}$$

where P_{R_M} is the received power in the transmission media.

A simulation, utilizing TOM, is presented in Fig. 4, where RF propagation in an area of 20m x 30m is simulated with one transmitter located at height of 0.70m. Here L_{veg} is the signal propagation loss due to the trunk diffraction, obtained with the diffraction equations given in [5].

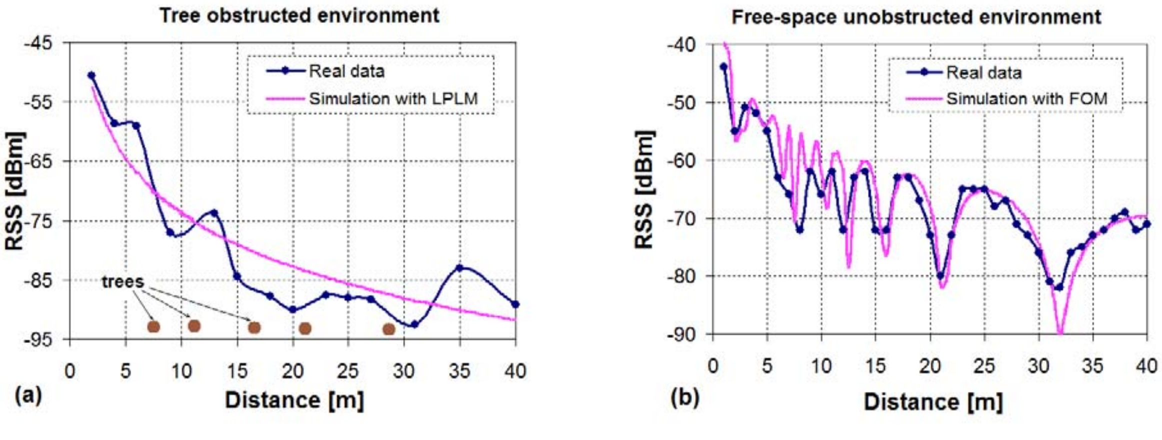


Figure 3.Real-field measurements in tree environment (a) and free space (b) and simulations with LPLM (a) and FOM (b).

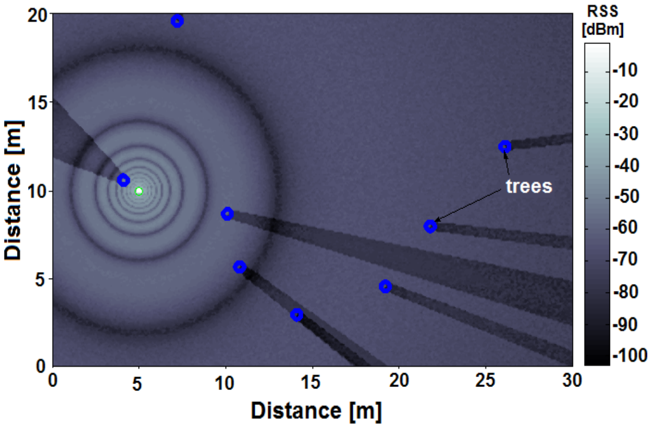


Figure 4. Simulation with TOM

When the trees are located close to each other, as it is in the thick forest environment, their influence over the RSS of the radio signal have to be examined as a whole, and the TOM may not be the best suited model. As it is shown in Fig. 3(a) the LPLM with n=3 successfully could be used to model the propagation in thick forest environment.

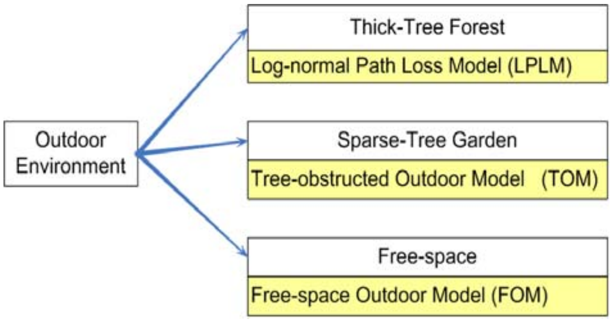

Outdoor Environment
Thick-Tree Forest
Log-normal Path Loss Model (LPLM)
Sparse-Tree Garden
Tree-obstructed Outdoor Model (TOM)
Free-space
Free-space Outdoor Model (FOM)

Figure 5. Classification of the outdoor models

e) Classification of the outdoor models: Based on the WSN application scenarios three outdoor types are identified depending on the presence of trees: thick forest, sparse garden and free space. A specific RF propagation model corresponds to each environment, predicting the RSS at any T-R distance. Fig. 5 depicts the three outdoor environments and the corresponding propagation models. The choice of the appropriate propagation model for a specific environment is crutial for the success of any RSS prediction. As shown in Fig. 3, where the measured and simulated results are compared, when using inappropriate RF propagation model and not taking into consideration the deployment factors during the pre-deployment phase, may lead to communication and data loss.

f) Measured and simulated results: In order to investigate the eligibility of the proposed models, measurements and simulations have been conducted. The results presented in Fig. 3, Fig. 6 and Fig. 7 confirm that the propagation model is environment-dependent.

Real-field experiments have been conducted in a free space area of 100m x 50m, using Tmote Sky nodes [13], operating in the band of 2.4GHz with internal antenna and Tx power at the maximum of 0dBm. The process of measuring the RSS consists of the following steps: Every 100msec the transmitter sends to the receiver a packet of length 21 Bytes, including one counter holding the number of packets sent. The counter reaches a maximum of 150,

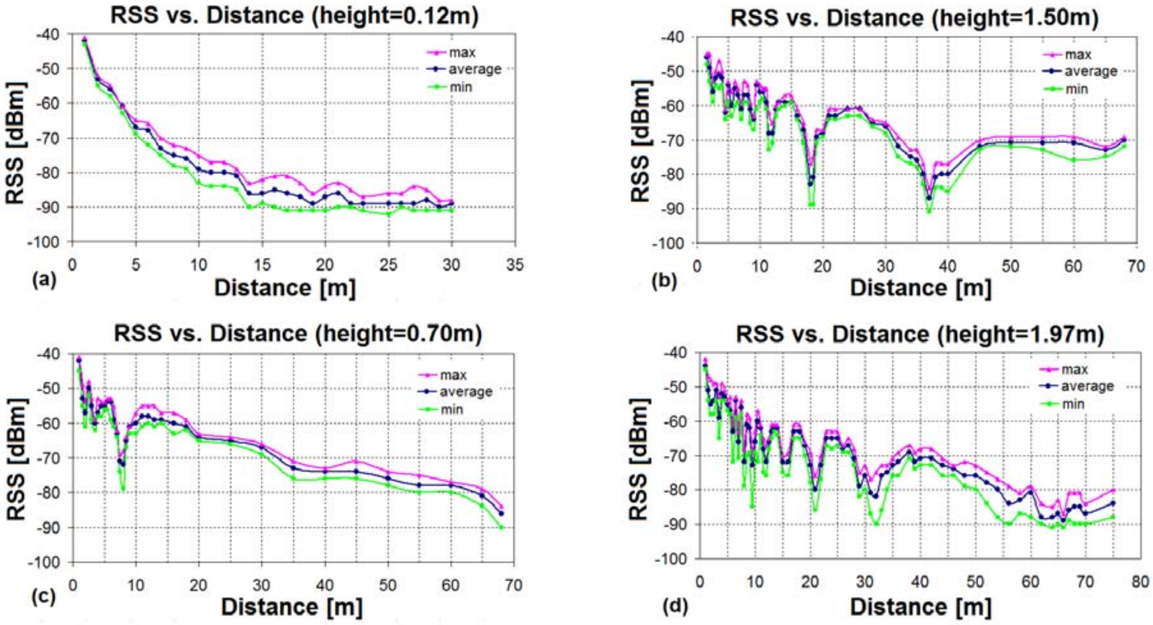


Figure 6. Real outdoor measured results in free space (min, average, max) for T-R heights of the ground: (a) 0.12m, (b) 1.50m, (c) 0.70m and (d) 1.97m

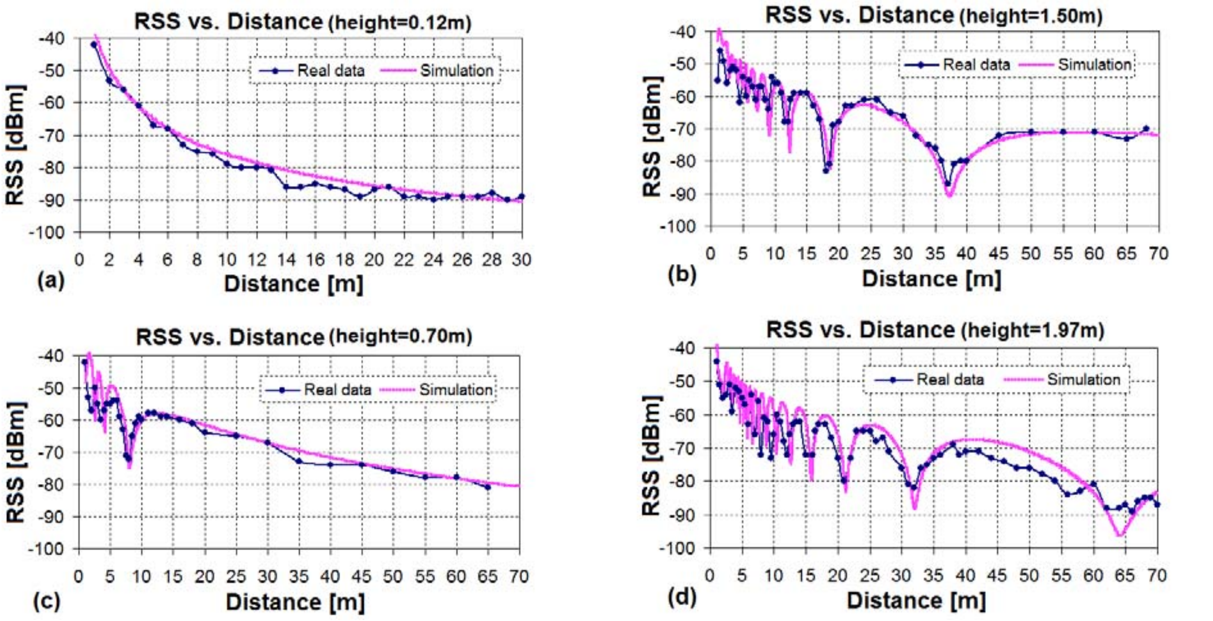


Figure 7. Real outdoor measured results and simulation results with FOM for T-R heights of the ground: (a) 0.12m, (b) 1.50m, (c) 0.70m and (d) 1.97m

indicating that at most 150 packets are sent for each distance. The receiver incorporates the RSS of the received packet in a 1-Byte field to the received packet and forwards it to the base station, connected through USB to a laptop.

Fig. 3 shows that a thick tree obstructed environment, with trees located every 4-5m, is best modeled using LPLM with n=3 (Fig. 3(a)), while a free space without trees is adequately modeled with FOM, shown in Fig. 3(b). Both measurements are taken for T-R height of 1.97m.

Fig. 6 shows the measured RSS max, average and min value when the T-R pair is deployed at four different heights from the ground, 1.97m, 1.50m, 0.70m and 0.12m respectively. One important observation regarding the influence of the ground reflection phenomenon over the RSS is that the reflected and the direct signals interact and create regions, called ‘nulls’, with very low and unstable RSS. The plots in Fig. 6 present that the RSS is more variable in the areas with low RSS value, below -80dBm, and in the areas with ‘nulls’.

Fig. 7 shows the average value of the real-field measured RSS for the fourth T-R heights from Fig. 6 compared with simulated RSS with FOM. All plots clearly demonstrate that data measured are in conformance with the simulation results.

2) Antenna specifics: The antenna orientation, polarization and gain also influence the RSS. The antenna gain participates directly in the propagation model equation as coefficient K_1 and K_2 in (5), while the antenna polarization predetermines the reflection coefficient, as presented in (3). In the simulations, the coefficients ε_1 and σ_1 take values according to material parameters given in [3][4], that is $\varepsilon_1 = 15$ and $\sigma_1 = 0.008$S/m. The importance of the antenna particularities for the RSS prediction is described in detail in [12] and [13].

3) RSS threshold: A major communication consideration is the correct message reception. It is assumed that a packet sent by a transmitter can be received by a receiver with certain confidence, only if the intensity of the received power is above a given threshold. The relation between the intensity of the received power and the packet loss was empirically investigated to find whether such a threshold exists. The results are presented in Fig. 8 and the following conclusions can be drawn:

- For RSS above -80dBm, packet loss is less than 5%
- For RSS between -80dBm and -87dBm, packet loss rises to 10%.
- For RSS approximately -90dBm, which corresponds to the sensitivity threshold of the CC2420 radio, packet loss can reach 100%.

From the results in Fig. 8, we can conclude that the RSS value of -80dBm is an acceptable threshold level. It indicates that signals with RSS above -80dBm will be received with probability over 95%. Similar results have been presented in [8].

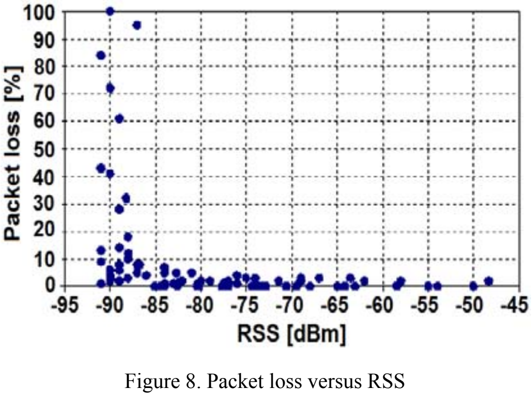


Figure 8. Packet loss versus RSS

C. RFCA Definition

This section presents a comprehensive description and analysis of the four steps of RFCA. The four steps are subsequent and the results of one task are inputs to the subsequent one.

1) Step1 - Discovering the most appropriate heights and distances of the sensor nodes: The most important task for the first step is the selection of the propagation model that corresponds to the target environment. When assuming unobstructed environment, the appropriate model is FOM. Using FOM, the relation among RSS, height from the ground and distance are determined by simulation. During this simulation all practical heights from the ground and distances are combined to produce the RSS for the maximum Tx power of 0dBm. Fig. 9 presents the results of the simulation, where heights vary from 0m to 3m and the distance from 1m to 100m. The gray-scale bar is marked with numbers indentifying RSS areas. For instance, the area marked as 4 designates RSS values between -70dBm and -80dBm. In general, Fig. 9 gives an idea about which heights and distances can be combined to achieve sufficient RSS for reliable communication.

Based on the real-field measurements and simulation results presented in Fig.6, Fig.7 and Fig.9, the following constraints for choosing heights and distances are formulated:

- the sensor node should not be placed in area with deep and wide 'nulls' due to the high variability of the signal and high possibility of signal and packet loss,
- the sensor node should not be placed in areas with RSS lower then -80dBm due to great variance of RSS and higher probability of packet loss.

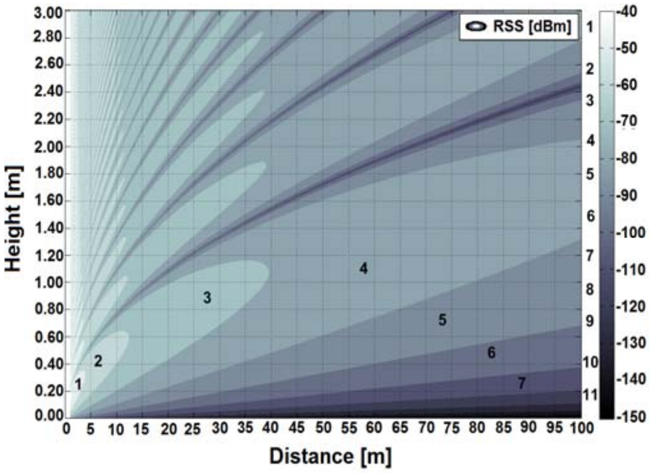


Figure 9. Simulation of the RSS

2) Step 2-Reducing Tx power. The RSS simulation may also help to reduce the Tx power. Referring to Fig. 7, power simulations are performed for three heights from the ground (1.97m, 1.5m and 0.70m) and for distances of 25m and 50m for each height case. These distances were chosen according to Fig. 7, cases (b), (c) and (d), so that the RSS, at 25m, has the maximum value after a 'null' at maximum Tx power of 0dBm. The power simulation results, presented in Fig. 10, illustrate that the Tx power could be reduced and still the RSS be above the threshold of -80dBm, as shown in Table I.

TABLE I. REDUCTION OF TX POWER

Height	Distance 25m	Distance 50m
1.97m	up to -16dBm	up to -8dBm
1.50m	up to -16dBm	up to -8dBm
0.70m	up to -15dBm	up to -4dBm

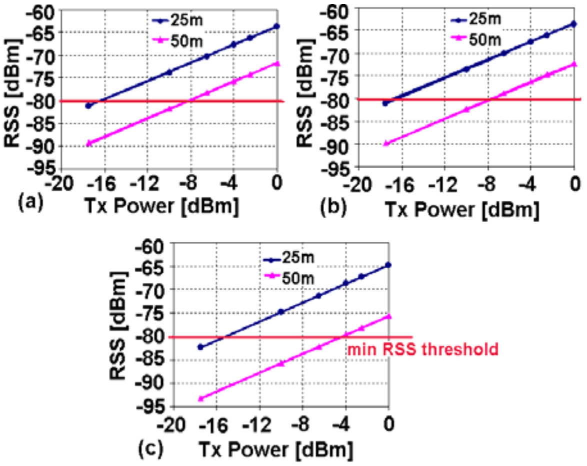


Figure 10. Power simulation for heights:
(a) 1.97m (b) 1.50m and (c) 0.70m

3) Step 3-Minimizing the interference from non-neighbor nodes:

The neighborhood nodes should lie at distance where the RSS is more than -80dBm to guarantee min allowed percentage of packet loss, while the non-neighbor nodes should lie at distance where the RSS is about -90dBm, low enough to guarantee maximum percentage of packet loss.

The simulation results about height and distance of *Step1* and the power simulations of *Step 2* are input parameters for *Step 3,* where nodes' height, T-R distance and power are combined in order to satisfy the following requirement:

- the RSS of a neighbor node should be above the min RSS threshold of -80dBm in order to guarantee that the messages will be received with packet loss below 5%.
- the RSS of a non-neighbor node should be below the sensitivity threshold of -90dBm, in order to minimize the possibility to receive and send signals and thus to interfere the neighboring communication.

For instance, let us assume the simple line topology of Fig. 11, where the distance between any two neighbor nodes is 25m and node 1 has to communicate with node 2 but should not communicate with node 3. In order to satisfy the above requirements, the T-R height is selected at 0.70m and the Tx power at -15dBm according to Fig. 10 (c). For this combination, the RSS is predicted as -80dBm at 25m for the neighbor node and as -91dBm at 50m for non-neighbor node, as shown with the ellipse in Fig. 12.

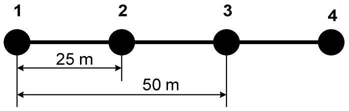


Figure 11. Packet loss versus RSS

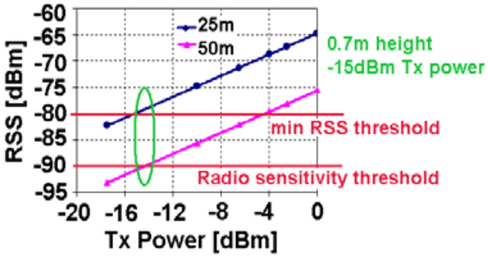


Figure 12. T-R height 0.70m (extracted from Fig. 10(c))

4) *Step 4-Deriving the optimal deployment parameters*: The final step aims at summarizing results and proposing the most appropriate deployment parameters according to initial criteria i.e. minimum node numbers, less neighbor nodes, minimum disturbance from the non-neighbors, etc.

D. *RFCA implementation*

The pseudo code of the RFCA implementation is shown in Fig. 13, consisting of four sequential tasks. Results of one task are inputs to the subsequent one. The final output is a vector with the optimal $h/P_{TX}/d$ combination or a matrix with a series of $h/P_{TX}/d$ satisfying the criteria. Input parameters are, the minimum and maximum height from the ground, H_{min} and H_{max}, the smallest and biggest neighbor distance, d_b and d_e, the minimum and maximum Tx power, P_{TXmin} and P_{TXmax}, the velocity of light, c, the radio frequency, f, and the permittivity of the reflecting surface, ε_r.

IV. Case Study: Application of RFCA to a Real Deployment Example

In this section, the RFCA is illustrated through an example for an environmental monitoring application in unobstructed environment, where environmental parameters such as temperature and humidity are under observation.

A. *Application Requirements*

Application scenario: Environment monitoring
Area type and size: 200m x 200m, open space
Sensor nodes: Tmote Sky or Telos B
Antenna type: Internal, inverted-F
Deployment criteria: minimum disturbance from the non-neighbor nodes with minimum Tx power

B. *Application Analysis*

The environmental parameters do not change in short distance. Therefore, the sensing radius is not decisive for the area coverage and the number of nodes. Referring to the RF propagation, shown in Fig. 7 and the requirements that the RSS should be more than -80dBm we can assume that one node per 50m - 70m would provide an acceptable degree of sensing coverage and reliable communication. Based on that, the distance as the input parameter is already known: d = 50m. The first simulation assumes maximum Tx power of the Tmote Sky sensor nodes at 0dBm and distance for the neighbor nodes of approximately 50m.

```
RFCA_Step1

Input parameters: $H_{min}$, $H_{max}$, $d_b$, $d_e$, $c$, $f$, $\varepsilon_r$
for $\forall\ d \in \{d_b : \text{step} : d_e\}$ and $\forall\ h \in \{H_{min} : \text{step} : H_{max}\}$
    Compute RSS ($d, h, \lambda, \varepsilon_r, P_{TX}$(0dBm))
     if RSS < min_RSS_treshold than
          Save $h$ in$\{h_{step1}\}$, $d$ in $\{d_{step1}\}$
     end if
  end for
end for
```

```
RFCA_Step2

Input parameters: $\{\{h_{step1}\},\{d_{step1}\}\},c,f,\varepsilon_r,P_{TXmin},P_{TXmax}$
for $P_{TX} \in \{ P_{TXmin}$: step : $P_{TXmax}\}$
   Compute RSS($d,h,\lambda,\varepsilon_r,P_{TX}$) for $\forall\ d/h$ combination,
   where $d \in \{d_{step1}\}$ and $h \in \{h_{step1}\}$
    if RSS < (min_RSS_treshold) than
         Save $h$ in $\{h_{step2}\}$, $d$ in$\{d_{step2}\}$, $P_{TX}$ in $\{P_{TXstep2}\}$
    end if
end for
```

```
RFCA_Step3

Input parameters: $\{\{h_{step2}\},\{d_{step2}\},\{P_{TXstep2}\}\}$, $c, f, \varepsilon_r$
Compute the non-neighbor node distance: $d_{non}$
for $\forall\ d/h/P_{TX}$ combination
     Compute RSS($d_{non},h,\lambda,\varepsilon_r,P_{TX}$) for $d_{non} \in \{d_{non} \pm 5\}$,
     $h \in \{h_{step2}\}$and $P_{TX} \in \{P_{TXstep2}\}$
     if RSS < (radio_sensitivity_treshold) than
          Save $h$ in $\{h_{step3}\}$, $d$ in $\{d_{step3}\}$, $P_{TX}$ in$\{P_{TXstep3}\}$
     end if
end for
```

```
RFCA_Step4

Input parameters: Deployment array $D(h, P_{TX}, d) \in$
                  $\{\{h_{step3}\},\{P_{TXstep3}\},\{d_{step3}\}\}$
Define deployment criterion: minimum disturbance
  from the non-neighbors; min Tx power, etc.
Sort $\{\{h_{step3}\},\{P_{TXstep3}\},\{d_{step3}\}\}$ according to the
  Deployment criterion 1
Sort $\{\{h_{step3}\},\{P_{TXstep3}\},\{d_{step3}\}\}$according to the
  Deployment criterion 2
Sort $\{\{h_{step3}\},\{P_{TXstep3}\},\{d_{step3}\}\}$according to the
  Deployment criterion N
 Obtain the final deployment parameters: $h$, $P_{TX}$, $d$
```

Figure 13. RFCA pseudo code

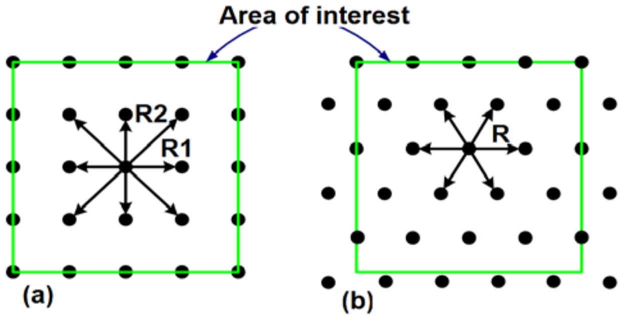


Figure 14. Deployment schemes:
(a) square grid, (b) triangular grid

One of the deployment goals is to optimize the number of nodes, which supposes manual deployment in grid topology [3]. The two most widely used deployment schemes for manual deployment are: square grid and triangular grid as shown in Fig. 14. While the triangular grid uses equal communication ranges among all six 1-hop neighbors, for the square grid a stable communication must be guaranteed in two radii R1 and R2 in order for a sensor node to communicate with its eight 1-hop neighbors. Assuming that the preferred communication range is 50m, then the communication range with complete connectivity is between R1=45m and R2=75m for square grid deployment, and R=45m–55m for triangular deployment with 5m deployment tolerance.

The number of nodes N, where complete coverage and connectivity is assured, can be calculated as:

- For the square grid

$$N=\left(\frac{L}{R}+1\right)\left(\frac{W}{R}+1\right) \qquad (7)$$

- For the triangular grid

$$N=\left(\frac{L}{R}+1\right)\left(\left(\underbrace{\frac{W}{R\sqrt{3/4}}}_{RoundUP}\right)+1\right)+\left(\underbrace{\frac{W}{2R\sqrt{3/4}}}_{RoundUP}\right) \qquad (8)$$

where W is the area width, L is the area length, R is the communication range. Consequently, for the square grid topology the number of nodes is N=25, and for the triangular grid is N=33.

1) Step 1-Discovering the most appropriate heights and distances of the sensor nodes

The most appropriate T-R heights, after the first simulation step are listed in Table II for square grid and Table III for triangular grid, respectively. Table II and Table III show only those heights, where the RSS is above -80dBm for the whole range of 45m–75m for the square grid and range of 45m–55m for triangular grid, as -80dBm is considered as an acceptable RSS threshold. The simulated T-R heights are between 0m and 3m.

2) Step 2-Reducing the transmission power

This step uses the results for T-R heights discovered in *Step 1* to investigate the possibility of reducing the Tx power. Simulations are performed for both the square and triangular grids for the T-R heights depicted in Table II and Table III as follows: (a) for square grid: distance range 45m–75m, T-R height range 1m–1.5m and (b) for triangular grid: distance range 45m–55m, T-R height ranges 0.70m–1.50m; 2.00m–2.20m; and 2.70m.

TABLE II. SQUIRE GRID SCHEME

Height range	RSS range for the 45m÷75m interval
1.00m – 1.50m	-68dBm – -77dBm

TABLE III.TRIANGULAR GRID SCHEME

Height range	RSS range for the 45m÷55m interval
0.70m – 1.50m	-68dBm – -77dBm
2.00m – 2.20m	-68dBm – -75dBm
2.70m	-69dBm – -76dBm

Results presented in Fig. 15 show the minimum RSS value for the whole distance range at specific T-R height, for five levels of Tx power. The conclusions based on the results are:

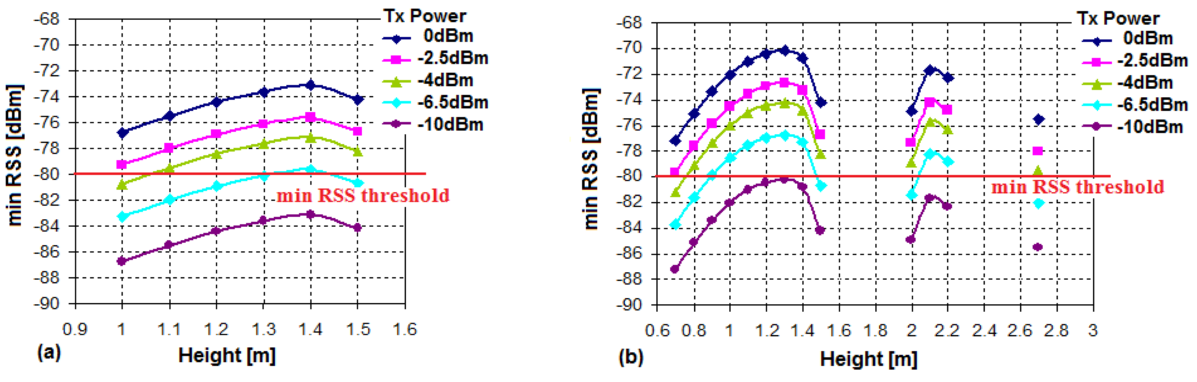


Figure 15. Power simulations for: (a) square and (b) triangular grid

- the most energy efficient combination of T-R height and Tx power for square grid is 1.4m and -6.5dBm, respectively.
- the most energy efficient combination of T-R height and Tx power for the triangular grid is 1.3m and -10dBm, respectively.

3) Step 3-Minimizing non-neighbor nodes interference

Step 3 aims at satisfying the requirements discussed in Section III C, which are:

- the RSS of a neighbor node should be above the min RSS threshold of -80dBm in order to guarantee that the sent messages will be received with packet loss less than 5%.
- the RSS of a non-neighbor node should be below the sensitivity threshold of -90dBm, in order to minimize the possibility to receive and send signals and thus to interfere the neighboring communication and to create huge data flow.

For the square and triangular grid schemes, the distance to non-neighbor node is about double R and R1, i.e. 90m–110m. Simulations are performed for the T-R heights and the Tx powers determined in *Step 2*, presented in Fig. 15 and listed in Table IV.

The simulation results for thus step are presented in Fig. 16 and Fig. 17, where the simulated RF signal propagation for distances between 45m and 110m for the T-R heights and Tx power taken from Table IV is shown. Based on this results the following conclusions are derived:

(a) The square grid scheme cannot fulfill the requirement for minimizing the interference from non-neighbor nodes. For the chosen T-R heights and Tx powers of Table IV at distance 90m–110m the RSS still has value significantly above the radio sensitivity threshold of -90dBm, as shown in Fig. 16.

(b) The triangular grid scheme can fulfill the requirement for minimizing the interference from non-neighbor nodes while ensuring good communication with the neighbor nodes only for the following heights and Tx power levels: 0.7m with -2.5dBm; 0.8m with -4dBm, 0.9m with -6.5dBm and 2.7m with -4dBm. This is presented in Fig. 17(a) and (b) with ellipses. The optimal combination for energy saving and reliable communication is T-R height at 0.9m and Tx power of -6.5dBm.

TABLE IV. T-R HEIGHTS AND TX POWERS

Square grid		Triangular grid	
T-R height	Tx Power	T-R height	Tx Power
1.00m	-2.5dBm	0.7m	-2.5dBm
1.10m	-4dBm	0.8m	-4dBm
1.20m	-4dBm	0.9m	-6.5dBm
1.30m	-4dBm	1.00m	-6.5dBm
1.40m	-6.5dBm	1.10m	-6.5dBm
1.50m	-4dBm	1.30m	-10dBm
		2.00m	-4dBm
		2.10m	-6.5dBm
		2.20m	-6.5dBm
		2.70m	-4dBm

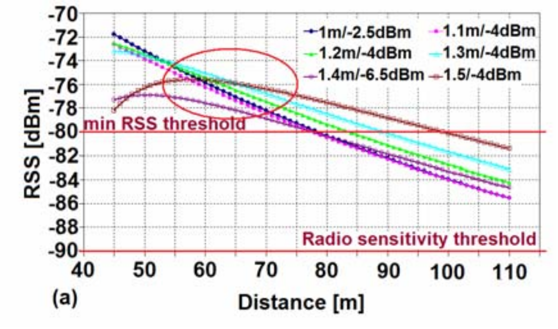


Figure 16. Non-neighbor interference simulations for squire grid scheme

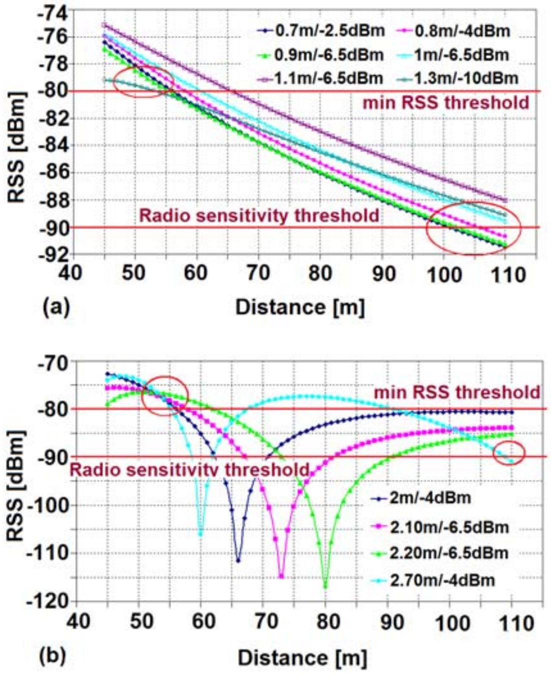


Figure 17. Non-neighbor interference simulations for triangular grid scheme: (a) for heights 0.7m–1.5m and (b) for heights 2m–3m

C. Deriving the final deployment parameters

Step 4 of RFCA aims at summarizing the results and at proposing the most appropriate deployment parameters. In this step, the evaluation of results from *Step 3* is according to the initial criteria defined as application requirement. For the described example those are minimum disturbance from the non-neighbors nodes with minimum Tx power. Base on that are the following conclusions:

1) Square grid scheme:

- The necessary number of nodes is *N*=25, neighbor nodes=8,
- The most energy efficient height-power combination is T-R height of 1.4m and Tx power of -6.5dBm,
- Cannot fulfill the requirement for minimizing the interference from non-neighbor nodes.

2) Triangular grid scheme:

- The necessary number of nodes is *N*=33, neighbor nodes=6,
- The most energy efficient height-power combination is T-R height of 1.3m and Tx power of -10dBm,
- The height-power combinations, which satisfy the criterion of minimum disturbance from non-neighbor nodes, are: 0.7m/-2.5dBm, 0.8m/-4dBm, 0.9m/-6.5dBm and 2.7m/-4dBm.

3) Deployment parameters:

The final deployment parameters for reliable communication, which satisfy all the initial criteria, are determined as follows:

- *Height:* 0.9m,
- *T-R distance:* 50–55m,
- *Tx power:* -6.5dBm,
- *Preferred deployment scheme:* triangular grid,
- *Number of nodes:* 33.

The proposed methodology could be used also to scale down the topology in applications where the network performance is studied, when adjust the deployment parameters: height, distance and transmission power. For instance, the deployment parameters from the example above require distance between the sensors about 50m and number of sensors 33, which is quite a big area for coverage. In order to form the same topology, but in smaller area the deployment parameters are recalculated as follows: for distance 5m, the height from the ground is 12cm with transmission power of -11dBm. These parameters' values ensure that the 1-hope neighbors can transmit and receive packets with RSS above -80dBm, when in the same time the RSS of the non-neighbor nodes transmission is approximately -90dBm. This approach is used in [16] when the topology is formed in order to be tested the network performance expressed as message delivery delay, network throughput and packets dropping percentage due to multi-hopping.

V. Discussion and Conclusion

This work discusses the importance of employing an algorithm, which can evaluate the sensor nodes deployment locations against communication-aware deployment criteria, into the pre-deployment phase. In addition, a pre-deployment simulation framework has been introduced and in its context a RF signal propagation-based connectivity algorithm (RFCA) has been developed to fulfill three deployment provisions: discovering the most appropriate height and distances for the sensor nodes, reducing the transmission power and minimizing the interference from non-neighbor nodes.

RFCA uses the RF signal propagation model to predict the received signal strength (RSS) in order to identify the most appropriate communication-based deployment parameters, i.e. T-R distance, height from the ground, and transmission power. The choice of the RF propagation models is crucial for the success of the prediction procedure, which is our motivation to study the importance of using proper simulation model for a specific environment. Since our work is focused on outdoor environment, we differentiate three types of outdoor environments: free unobstructed space, thick-tree forest and sparse-tree garden. In addition, four propagation models are overviewed and classified for their applicability to these three environments. Furthermore, real outdoor measurements are compared with the simulation results to confirm the applicability of an RF propagation model to a particular environment.

Another important factor, which was considered during the pre-deployment phase, is the criteria that distinguish the radio range for neighbors and non-neighbors. It is common belief that the communication between two sensor nodes is good until certain distance and gets worsen after it. To investigate how reasonable this statement is, we performed a set of measurements. From the results we concluded that a packet sent by a transmitter can be received by a receiver with certain confidence, only if the intensity of the received power is above a given threshold. In addition, it was also determined that the RSS in closer distance is not always stronger. For instance, in Fig. 3(b) the RSS at 32m is -82dBm, but at 40m is -70dBm. Thus, the threshold was empirically investigated as relation between the intensity of the received power and the packet loss. As a conclusion, the criteria for neighbor node distance should not be the distance as maximum value, but the distance where the RSS is high enough to guarantee min allowed percentage of packet loss and for non-neighbors - the distance where the RSS is low enough to maximize the probability for packet loss.

Finally, a deployment case study illustrated the RFCA functionality in deriving the optimal deployment parameters, consisting of height, distance, and Tx power, for reliable communication. RFCA is generic enough to be combined with any topology optimization algorithm such as min number of nodes deployment, maximum battery life deployment, power-aware deployment, etc.

Analysis of the results presented in this work has shown that a communication-aware pre-deployment simulation for preliminary planning of the deployment can significantly reduce the link quality problems due to the physical layer communication loss.

Acknowledgment

The work reported here was performed as part of the research Program uSWN FP6-2005 IST-034642 and funded by the European Social Fund (ESF).

References

[1] Ts. Stoyanova, F. Kerasiotis, A. Prayati, G. Papadopoulos, "Communication-aware deployment for Wireless Sensor Networks", Proc. of SENSORCOMM 2008, August 2008, Cap Esterel, France, pp. 217–222, DOI: 10.1109/SENSORCOMM.2008.97.

[2] S. Shakkottai, R. Srikant, N. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter", Ad Hoc Networks, Vol. 3, Issue 6, November 2005, pp. 702–716, Elsevier Science Publishers, DOI:10.1016/j.adhoc.2004.02.001.

[3] H. Zhang, J. C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks", Ad Hoc & Sensor Wireless Networks, Vol. 1, pp. 89–124, March 2005.

[4] F. Xue, P.R. Kumar, "The number of neighbours needed for connectivity of wireless networks", Wireless Networks, Vol. 10, Issue 2, 2004, Kluwer Academic Publishers, pp. 169–181, DOI:10.1023/B:WINE.0000013081.09837.c0.

[5] T. S. Rappaport, Wireless Communications: Principles and Practice, 2nd Edition, Prentice Hall, 2001.

[6] H. R. Anderson, Fixed Broadband Wireless System Design, John Wiley & Sons, Ltd, 2003.

[7] S. Haykin, M. Moher, Modern Wireless Communications, Pearson Education Inc., Pearson Prentice Hall, 2005.

[8] K. Srinivasan, P. Levis, "RSSI is Under Appreciated", Third Workshop on EmNets, 2006.

[9] L. Barboni, M. Valle, "Wireless Sensor Networks Power-Aware Deployment", SENSORCOMM 2008, August 2008, Cap Esterel, France, pp. 252–257, DOI: 10.1109/SENSORCOMM.2008.48.

[10] C. E. Otero, I. Kostanic, L. D. Otero, "Development of a Simulator for Stochastic Deployment of Wireless Sensor Networks", Journal Of Networks, Vol. 4, No. 8, October 2009, pp.754-762, 2009 Academy Publisher, DOI:10.4304/jnw.4.8.754-762

[11] A. Fanimokun, J.Frolik, "Effects of natural propagation environments on wireless sensor network coverage area", Proc. of the 35th Southeastern Symposium System Theory (SSST03), 16-18 March 2003.

[12] Ts. Stoyanova, F. Kerasiotis, A. Prayati, G. Papadopoulos, "Evaluation of Impact Factors on RSS Accuracy for Localization and Tracking Applications in Sensor Networks", Special issue of Telecommunications Systems Journal, Springer, Vol. 42, pp. 235–248, Dec. 2009, DOI:10.1007/s11235-009-9183-8)

[13] Ts. Stoyanova, F. Kerasiotis, A. Prayati, G. Papadopoulos, "A Practical RF Propagation Model for Wireless Network Sensors", Third International Conference on Sensor Technologies and Applications, SENSORCOMM 2009, June 18–23, 2009, Athens, Greece, DOI: 10.1109/SENSORCOMM.2009.39

[14] 'Ultra low power IEEE 802.15.4 compliant wireless sensor module', Moteiv Corporation, 2006.

[15] 'TelosB mote platform', Crossbow Technology. Copyright 2009, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf

[16] C. Antonopoulos, A. Prayati, F. Kerasiotis, G. Papadopoulos, "CSMA-MAC Performance Evaluation for WSN Applications", Third Int. Conference on Sensor Technologies and Applications, SENSORCOMM 2009, June 18–23,pp.13-18, 2009, Athens, Greece, DOI: 10.1109/SENSORCOMM.2009.10

Practical Deployments of Wireless Sensor Networks: a Survey

[1]Miguel Garcia, [2]Diana Bri, [3]Sandra Sendra, [4]Jaime Lloret
Integrated Management Coastal Research Institute, Universidad Politécnica de Valencia
Camino Vera s/n, Valencia, Spain
[1]migarpi@posgrado.upv.es, [2]diabrmo@posgrado.upv.es, [3]sansenco@posgrado.upv.es,
[4]jlloret@dcom.upv.es

Abstract—**Sensor networks are becoming one of the most used technologies in our lives. Although there are several surveys about sensor networks, almost all of them focus on theoretical basis with little simulations. Some studies only explain where this kind of networks can be used, without details on how they can be applied. This paper classifies applications of wireless sensor networks and describes real implementations. There is not any survey like the one presented in this paper. The target is to complement the existing surveys, by presenting details on real implementations in order to understand how these networks run, and how they are designed, maintained and operated.**

Keywords: sensor, sensor networks, sensor applications, motes.

I. Introduction

Latest advances in sensor technology are leading to the development of distributed mechanisms and small devices with both low cost and low energy consumption. In addition, these devices are capable of processing information locally and communicating wirelessly with other elements. These devices are called sensor nodes or motes.

In some cases, amount of sensors are necessary to sense an environment or take measurements from the surroundings. While they are sensing, they have also to communicate between them and/or with a central server. On the other hand, a monitored environment doesn't have infrastructure to supply energy for communication. So, motes must work with small batteries and use wireless channels.

Another feature of the sensor networks is their capacity of distributed processing. It is necessary because, the communication is the process that consumes more energy. A distributed system means that some sensors need to communicate through long distances. So, it is a good idea to process locally data as much as possible in order to minimize the bit rate.

Wireless Sensors Networks (WSNs) are formed dynamically because the connectivity between nodes depends on their position and their position variation over the time (if they are mobile). This kind of networks is characterized as being easy to be deployed and self-configuring. A sensor node is composed by a transmitter, a receiver, and it offers services of routing between nodes without direct vision, as well as records data from other sensors.

The following are some of the main features of WSNs:

- Dynamic topologies: In a wireless sensor network, the topology is always changing because nodes can fail or new nodes can join the network. These changes affect the communication between sensors.
- Variability channel: The radio channel is highly variable. There are several phenomena, such as the attenuation, fast fainting, slow fainting and interference that can cause data errors.
- Ad hoc networks: Generally, sensor networks do not have a wired network infrastructure. All motes are transceivers and routers simultaneously. However, the concept of sink node is important; this node collects the information and sends it to a central computer capable of processing these data.
- Failure tolerance: A sensor node should be able to continue operating despite of the existence of errors in the system.
- Multi-hop or broadcast communications: This type of networks use any routing protocol to enable communications multi-hop, although it is also very common the use of messages sent in broadcast.
- Power saving: It is one of the most important features in these networks. Currently, the motes have limited energy. A sensor node should have an ultra low consumption processor and transceiver radio. It is one of the most restrictive features.
- Limited hardware: In order to get an adjusted consumption, the hardware should be simple; this brings a limited process capacity.
- Production costs: Sensor networks are formed by high number of nodes. Motes must be economic to create a reliable network.

Since sensors can collect data from environment, a sensor network has many application areas, such as habitat monitoring, fire detection, motion tracking, reservoir water controlling, or intruders controlling. In order to control, monitor, tracking or detect, it is necessary a large quantity of sensor nodes which detect the monitored event (light, pressure, sound, heat, humidity, electro-magnetic field, proximity, location, etc.), and transmit it to a base station, where last action will be made. Sensor networks have become very useful for our lives and they have penetrated in domains such as health, home care, environment monitoring, etc [1].

The structure of the paper is as follows. The related work and our motivation are presented in Section 2. Section 3 describes the most important parts of the WSN architecture. Our classification of the applications of WSNs is shown in Section 4. This section is divided in five parts and each of them analyses an application and shows several real implementations. Then, in Section 5, we present a comparative study of different real deployments. In section 6, we show limitations and challenges of WSN. Finally, in section 7, we conclude the paper.

II. Related work and motivation

Wireless sensor networks and their applications have been a popular research field because they provide a lot of facilities in our society.

Nowadays, the WSNs literature has large number of researches and studies. A pioneering work related to sensor networks identified the main characteristics of this type of networks. An example is the paper “A Survey on Sensor Networks” [2], which analyses the state of the art of sensor networks and describes their characteristics, architecture, and protocols. Another work is the paper presented by K. Akkaya and M. Younis [3], where only some routing protocols and types of networks are analyzed. The authors show the advances in designing WSN’s new algorithms. Ning Xu, in his technical report [4], describes several installed systems of sensor networks, but only in the health and environment fields. Standards, protocols, and others important aspects about sensor networks are described in the paper “Wireless Sensor Networks and Applications: a Survey” [5].

Most of the surveys related to sensor networks describe their characteristics in great detail, but, in our case, the main contribution is that the survey is focused on analyzing and describing the fields of application. In addition, examples about real implementations will be explained in each field of application.

III. Wireless sensor network architecture

In this section, we will present the generic architecture of sensor networks. First, we introduce the motes, their architecture, and their main features. After it, we will describe the communication protocol stack used in these networks and we will briefly analyze each layer.

A. Motes

A network of sensors is formed by sensor nodes, also called motes. Despite of what they are measuring with their sensing unit, a mote needs to process the data, to store it, using a processing unit, and transmit the information to the network, by using the transmitter/receiver unit and taking care of how much energy is available in its power unit [2][6]. The processing unit has a processor chip, usually a microcontroller, a volatile memory, and a non-volatile memory if the data needs to be stored. In order to transmit the information, some specific protocols for sensor networks have been created. When the information of many motes is coordinated using a protocol, the environment can be measured in great detail. Its main components are shown in Fig. 1.

The main issue in today’s mote development is to minimize their size for optimizing their power supply.

A sensor network is a group of motes that cooperate to do a specific task [7]. The precision of their tasks depends on the density of the scatter and on their coordination. Originally, they were formed by a small number of motes that were connected with a central station. Nowadays, WSNs allow distributed networks having more measurements and closer to the event.

B. Embedded operating systems for sensors platforms

In order to manage the hardware of a mote, an operative system is needed. It is responsible to make the mote to carry out its operations and tasks.

A major difference between sensor networks and more traditional computing platforms, it is the extreme emphasis in sensor networks on power management. A large number of applications require battery-powered operation for extended periods of time. In order to manage power efficiently, each subsystem of the platform is powered individually.

For example, a radio should have to be turned on only during active communication, and it should be possible to shut down the CPU between processing requests. Similarly, it should be possible to power down the sensor and I/O subsystems individually when not in use.

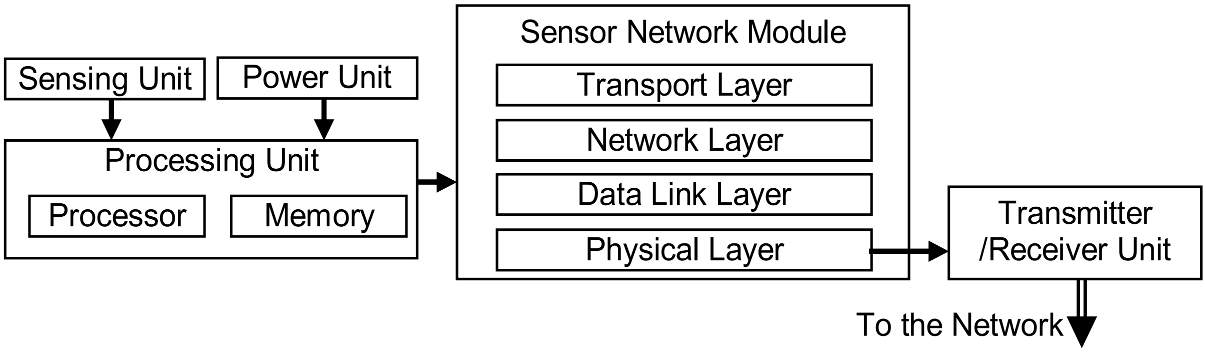


Figure 1. Components of a mote

TABLE I. COMPARATIVE OF MOTES

	Micaz	Mica2	Mica2dot	Tmote	TinyNode
Distributed by	Crossbow			Moteiv	Shockfish
Clock Frequency	7.37MHz		4MHz	8MHz	
RAM	4KB			10KB	
Batteries	2 AA		Coin cell	2 AA	Solar
Microcontroller	Atmel Atmega 128L			Texas Instruments – MSP 430 microcontroller	

In spite of WSNs have a short history, there are several enterprises working in this technology. For example, Crossbow [8] is a company that develops hardware and software platforms for WSNs. Some of their products are Mica, Mica2, Micaz, Mica2dot, telos and telosb. Moteiv [9] developed the Tmote Sky and Tmote Invent platforms. Tmote Sky is a platform focused for low consumption and high collection of data applications. It has integrated sensors, radio, antenna and microcontroller. Besides it can be easily programmed. Finally, Shockfish is a Switzerland enterprise that developed TinyNode [9]. Table I summarizes the main features of these motes.

Several operative systems have been developed for the motes such as Bertha[11], MagnetOS [12], LiteOS [13], TinyOS [14], and so on. Their main features are:

- *Bertha (pushpin computing platform).* It is a software platform designed to deploy a distributed WSN with a lot of identical nodes. Their main functions are divided in the following subsystems:
  - Management of processes.
  - Management of structures of data.
  - Organization of neighbours
  - Network interface
- *MagnetOS:* It is a distributed operative system for sensor or adhoc networks. Its objective is to run network applications of low consumption devices. It is very adaptive and easy to implement.
- *LiteOS:* It is a multi-threaded operating system that provides Unix-like abstractions for wireless sensor networks. It offers a hierarchical file system and a wireless shell interface for user interaction.
- *TinyOS:* It is used for Tmote Sky. It is a reduced multi tasks core useful for small devices, like the motes. It is an "event-driven" operative system, that is, when an event happened, this calls to the corresponding functions. It has been developed for WSNs with limited resources.

The programming of sensors is quite complicated because they have a limited calculus capacity and very few resources. Several programming languages have been developed to program the motes. Some of them are nesC [15], Protothreads [16], Giotto [17], and so on.

The libraries and applications of TinyOS are written in nesC, a version of C designed to program embedded systems. In nesC, the programs are composed by linked components.

C. *Protocol stack*

In order to enable the communication and data transfer between two sensor nodes there have to be some rules and conventions. A protocol defines the behavior of their connection and how they exchange information over a network medium. The communication process between motes and how information from a mote moves through a network medium to another mote is implemented using layers.

Each layer is reasonably self-contained so that the tasks assigned to each layer can be implemented independently. This enables the solutions offered by one layer to be updated without adversely affecting the other layers.

The sensor module shown in Fig. 1 has 4 layers. The first three layers handle data transport issues. A wide variety of communication protocols exist for each one of them. Here it is a brief description and some protocol examples:

1. Physical Layer: It is the closest to the physical network medium and tells a sensor node how to transmit to the medium and how to receive from it. Sensors transmission medium could be any wire, such as copper or fibber, or wireless (wireless sensors could use 433 MHz, 915 MHz or 2.4 GHz).
2. Data Link Layer: It provides the functional and procedural means to transfer data between network sensor nodes and to detect and possibly correct errors that may occur in the physical layer. Some data link layer protocols are mainly deployed for sensors such as S-MAC [18], IEEE 802.15.4 [19], Zigbee [20], and so on.
3. Network Layer: It provides the functional and procedural means of transferring variable length data sequences from a source to a destination through a path. Many routing protocols have been designed for ad hoc and sensor networks. They can be grouped in Pro-active Routing, Reactive Routing, Geographical Routing, and so on [3].
4. Transport Layer: This layer becomes necessary if it is wanted to access the system from Internet or other external networks. To our current knowledge, there is no any work discussing or proposing other mechanisms, different from the ones used in Internet, to be applied in sensor networks exclusively.
5. Application Layer: It offers to the applications the possibility of gaining access to the services provided by the rest of the layers. In addition, this layer defines the protocols used by the applications to exchange data.

The following section shows the main fields where wireless sensor networks can be applied.

IV. Wireless Sensor Networks Applications

This section presents a few applications where WSNs have been usefully implemented or have a big potential to be implemented.

A. Real deployments for Health

Fielding health-related deployments, WSNs can provide a better quality of life for people with physic or psychic difficulties. There are several main sub-fields, such as:

1) Human body implementations and body parameter measurements

Sensors let control important parameters of the human body like the heart rate or the blood pressure in order to diagnose the illness and identify a particular health problem. Some similar applications include Glucose level monitors [21], organ monitors [22], cancer detectors [23] and general health monitors. The idea of embedding wireless biomedical sensors inside human body is promising, although many additional challenges exist: (i) the system must be ultra-safe and reliable, (ii) it requires minimal maintenance, and (iii) must deal with the energy-harnessing from body heat. With more researches and progress in this field, better quality of life can be achieved and medical cost can be reduced. One of the examples given is the Swallowable sensor [24]. It is a wireless capsule that is ingested and can help to diagnose of stomach disorders [25]. A little medical device developed by Buffalo [26]. The intelligent pill is taken in by a patient and begins to transmit information to a receiver. This receiver is carried by same patient and gathers data from the stomach while the pill goes over it. Some of these devices include a microscopic camera. This system helps to diagnose stomach-ache and others stomach disorders that affect 20% of the humans. Sensor (pill) transmits information on levels of acidity, pressure, or time of digestion, while it is travelling in stomach. The capsule is thrown out two days after and it is recovered for its analysis and for downloading data. Another example was presented by Loren et al. in [27]. This work describes a biomedical application:, i.e., the artificial retina. Retina prosthesis chip consists on 100 microsensors that are built and implanted within human eye. This allows patients with no vision or limited vision to see at an acceptable level. The wireless communication is required to suit the need for feedback control, image identification and validation. TDMA is used for this application to serve the purpose of energy conservation because the communication pattern is deterministic and periodic.

Huan-Bang Li et al. proposed body area networks for three categories Medical and Healthcare Applications, Applications for assisting persons with disabilities, and Entertainment applications [28]. Authors proposed Zigbee and Bluetooth technology for the sensors.

Bartosz P. Jarochowski et al developed an application to rehabilitation centres [29]. In this system a personal node is located on the patient, for example on a belt clip or incorporated into an arm-band. This node stores information about the patient's session rehabilitation exercises and it is possible to obtain some statistics. Finally, the information can be sent to the medical control centre. Then, it can be analyzed by the doctors in order to improve the treatment for the next session, if it is needed.

2) Health control, monitor, and tracking systems

These systems have many applications. They can be used by patients who are very ill and can't go out. They can be controlled remotely by doctors. There are many proposals for monitoring elderly people and for tracking people with Alzheimer. In 2006, Bo Sun Hwang et al. developed a monitoring system which was focused on the activities of an individual daily living in a home [30]. Sensors communicate through the Bluetooth protocol [31]. The system detects the movements of a subject and then his/her activity pattern and position in a home is analyzed by a tracking algorithm. The system can be used to monitor disabled and elderly people by means of the graphical activity obtained from their system. Another work was presented by Hyung Jun Kim et al., in 2006 [32]. They proposed a home-based monitoring system that was continuously and unobtrusively monitoring a patient's condition. The system was implemented using Zigbee technology. A. Wood et al. presented ALARM-NET in [33], a wireless sensor network for assisted-living and residential monitoring. It integrates environmental and physiological sensors in a heterogeneous architecture to determine circadian activity rhythms of residents. In 2007, Yaw-Jen Lin et al. presented a ubiquitous monitor system integrated with biosensors and Radio Frequency Identification (RFID) technology [34]. The system was expected to improve the Activity of Daily Living of the disabled and elderly people, to detect the emergencies or accidents in order to enhance the quality of care.

3) Developments to make the life easier for disabled and elderly people

Marjorie Skubic presented in [35] a multidisciplinary project to investigate the use of sensor technology to provide early identification of problems in mobility and cognition, helping residents manage illness and impairments and stay as healthy and independent as possible. It uses an event-driven, video sensor network that hides identifying features of the residents and a reasoning component that fuses sensor and video data and analyzes patterns of behavioural activity. F. Brunetti et al. presented a system in [36] for motion caption and assessment in biomechanics using a wireless inertial sensors network using the IEEE 802.15.4 protocol. The platform expands the frontiers of movement analysis for motion caption. Many other sensor networks deployments exist for a wheelchair to avoid collisions such as the one presented by Holly A. Yanco et al in [37] and the one presented by R W. Gunderson et al. [38]. Their development consisted on a range of sensors mounted onto wheelchairs to provide navigation feedback and obstacle detection.

More sub-fields can be found but there are not so many deployments as the ones described before. One of them is to control drug administration in hospitals. Patients have sensor nodes that monitor their diseases and required medications. So, any doctor will prescribe always correctly drugs for that patient. In addition, doctors may also carry a sensor node, which allows other doctors to locate them within the hospital. The other one is the use of sensor networks to control and monitor epidemics produced in any place of the world. Maps of risk can be extracted using sensors, that is, knowing which places are infected or could be in the near future.

Paulo Bartolomeu Vasco Santos et al. have combined both health care and home automation in [39]. This modular system is formed by different subsystems and it can combine them according to the patient needs. Its most interesting contribution is the capacity of automating the house. It is a great advance for reduced mobility people. For example, the system can open a door; close a window or turn down a blind. Besides, this system can sense some vital signals.

A healthcare service for home has been presented by Nuri F.Ince et al. in [40]. It has a useful application for elderly and cognitively disabled people. This system is composed by several fixed sensors that locate a patient at home and, working together with wearable sensors, collect data to determine what bathroom activities are being done in order to know the state of the patient.

To control the main vital statistics and the intensity and duration of rehabilitation exercises that a patient has to make at home is possible with the system proposed by Chris Otto et al. [41]. However, its main advantage is its ubiquity, that is, wherever the patient is placed. It is possible thanks to the combination of ZigBee (or Bluetooth) and a personal device like a mobile phone or personal digital assistant (PDA) with GPRS/3G data network connection.

B. Real applications for the environment

This type of applications should have a not-invasive character in order to avoid alterations to the environment. Besides, it should be a robust and precise system in order transport all the data to the control point without errors. It should also be a low cost system, because it is implemented outdoors with batteries and maybe some nodes break down and should be replaced. Applications based on the detection of natural disasters, monitoring and control of agriculture, ecosystems and geophysical studies, flood detection, precision agriculture, biological complexity mapping of the environment and forest fire detection can be also included in this field. Some real applications for the environment are the following:

At the island Great Duck, close the coasts of Maine, USA, there is a network of sensor nodes used for monitoring microclimates at refuges and surrounding areas where marine birds nest [42]. It allows investigators to supervise at-risk species and its habitats. Intel Research Laboratory in collaboration with Atlantic College (Bar Harbor) and University of California in Berkeley distributed 32 motes in the island. Each mote had a microcontroller, a low potency radio, a memory and some batteries to monitoring temperature, humidity, pressure, and infrared emissions at mid-range. Motes send their data to station bases of the island and they are connected to Internet by satellite to permit the access.

A network of little sensors was placed in the River Ribble for disasters detection [43]. They monitor the water level and the flow preventing the flooding. The system obtains more data and with bigger precision than monitoring present-day systems and they provide the opportune photos of decision to prevent imminent risks. The final network consisted of three types of sensors. One measures the pressure under the water line to determine the depth and the others take care of the velocity of the flow of water, using ultrasounds underneath the surface and cameras web above it to accomplish a follow-up of objects. All these data are sent to a control point, in order to accomplish the opportune tasks according to the received data.

A deployment of a network of sensors has been performed by the Australian agriculture departments [44]. It is intended to save great quantities of water, and to help to hold agriculture by using wireless sensor nodes. The system irrigates better than other systems, so it saves water. It has been set up in a greenhouse and in a vineyard with satisfactory results.

A system to irrigate cotton fields at arid zones of Israel and Texas is shown in [45]. These systems are based on plantation temperature. Infrared sensors are localized near the trees and when they detect temperature superior to 82ºF for more the 4 hours, water is activated. It is proved that an optimum temperature for a correct growth of cotton is between 73ºF and 90ºF.

Fire Information and Rescue Equipment (FIRE) is a project developed by Berkeley's Mechanical Engineering department and the Chicago Fire Department (CFD) [46]. Fire-fighting and catastrophes in general generate extremely chaotic environment, where it is decisive to make a fast decision. The possibility of having information about different aspects of the catastrophe in these situations becomes an inestimable value. The WSN is used to monitor the firemen, policeman, and other players in the building, sending to a central point the status of the fire, status of health, etc. Each fireman gathers data from the sensors (SmokeNet) with a small computer which has the maps of the building.

There are also implementations for fire detection and verification, such as the one presented by the same authors of this paper in [47]. The system was developed to detect and monitor any fire in rural and forest environments, while it is being used to show the rural area to the visitors. It is composed of several wireless sensors distributed on an area and some wireless IP cameras. The information is distributed in a different manner based on the type of content delivered (video steaming, sensor information or sensor alarms). When a fire is detected by some sensor node, this send a signal to the server and the nearest camera focus this zone. The system has multiple video and sensor sources and two types of users: the regular users and the firefighters. An extension of this paper has been presented by the same authors in [48].

The ALERT system [49] is the most significant example of the flood detection. In this system, there are several types of sensors that are deployed, such as rainfall, water level, and weather sensors. These sensors supply information to the centralized database system in a pre-defined way.

The COUGAR Device Database Project [50] presented a distributed data management infrastructure where sensor nodes interact with in the field to provide snapshot and long-running queries. The system resides directly on the sensor nodes and creates the abstraction of a single processing node without centralizing data.

Environmental Observations and Forecasting System (EOFS) is a large distributed system that spans large geographic areas and monitors, models and forecasts physical processes, such as environmental pollution, flooding, etc. Usually it consists of 3 components: sensor stations, a distribution network, centralized processing farm [51].

There are also wireless sensor networks developed for food monitoring. G. Manes et al. presented in [52] a user defined scenario oriented to agro-food production phase monitoring. Their results show advantages both in terms of cost and complexity reduction and experienced QoS enhancement as well. Several indoor environmental control researches can be found. The applications are focused on the control of the air conditioning in the buildings. These systems are based on sensors of temperature distributed in different plants of the building. These sensors take upon themselves to send information of the temperature to a central or distributed system and it takes the appropriate actions.

C. *Real deployments for industry*

WSNs can improve many industrial and commercial applications for monitoring office buildings, intelligent museums, industrial sensing, product quality control, robot control and guidance, etc. These areas are very different, but WSNs can be used in all of them in order to monitor and help to manage the systems.

WSNs are also widely used for vehicle industry. They are used for a large variety of issues related with monitoring and control functions for motor management and systems of certainty and comfort (ASR, ABS, airbag, adjustment of the safety belt, air conditioning, etc.). Sensors are used to register the real status of the motor, motor's pressure of oil, temperature of motor, the number of revolutions and so on. There are other types of applications, such as to detect car thefts [53] and sensor networks for vehicle tracking [54].

An example given about the use of sensors in production is the one presented by Jenna Burrell et al in [55]. Using ethnographic research methods, the authors studied the structure of work activities and the needs and priorities of people working in a vineyard in order to understand the potential for sensor networks in agriculture. They discovered a need for localization functions to track the movement of objects, people, and equipment through the vineyard.

The application of WSNs in the household space is called smart house. We could have a system of remote metering with the networks of sensors, intelligent household appliances, a technical electric management of conditioning, illumination, water resources, home security, etc. There is a wireless sensor network deployment for light control in reference [56]. Fig. 2 shows an example of a smart house with many types of sensors placed.

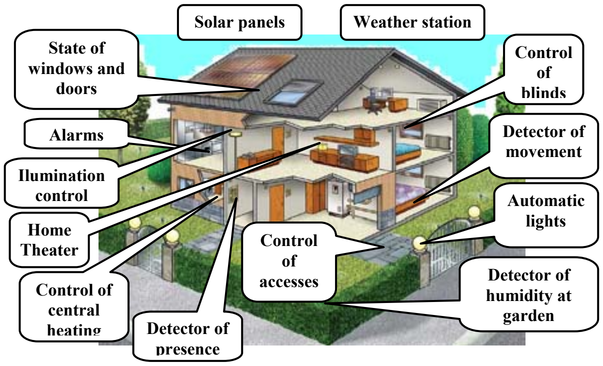


Figure 2. Smart house.

MeshNetics [57] has developed several sensor systems based on Zigbee for monitoring and control in building Automation.

The "Wireless Space Count System" is a system used for parking management [58]. This application was developed by NEDAP in January 2007. It can count the number of free places in the parking, thanks to a wireless sensor network distributed on every parking place.

In 2006, a mote network was designed, implemented, deployed and tested on the Golden Gate Bridge in San Francisco by some researchers of the Centre for Information Technology Research of the University of California [59]. Their objective was to monitor its structural condition. Sixty-four motes were distributed over the main span and southern tower, comprising the largest wireless vibration sensor network ever installed for structural health monitoring purposes. The spatially dense array resulted in an increase in effective signal-to-noise ratio compared to single, isolated, sensors. The array also allowed the values obtained from both vertical and torsional, to be analyzed easily and accurately.

Another interesting sensor application has been developed in the Loch Rannoch by BP. This project produced an efficient automated data collection system for machine monitoring and predictive maintenance that eliminated many of the manual processes by using handheld devices. It is equipped with 160 wireless motes [60].

There are applications that although they could be included in some aforementioned fields, they are also applications by themselves, e.g., intelligent nurseries personalized publicity, management of urban parking to know if there are empty places and where they are, and notification of the status of traffic at a city to know what way is the best-suited to get to a destination or to inform of the situation of the traffic at places where cameras do not cover up.

Finally, like a curious application, the interactive art is presented. Asholk Sukumaran presented the “Park View Hotel” in [61]. Using specially-built pointing devices, audiences in the park can access interior hotel spaces, by "pinging" them optically. Passers-by can emit infrared rays from several points of the street, and active sensors inside the building can illuminate rooms and change the colours. The color of the interior propagates leaking out of the building skin, jumping across the street, and entering some street-lights in the park below. The park enjoys a certain neighbourly access to the hotel, inverting the usual character of the relationship.

There are also some sensors deployments used to monitor the environment without battery that generate their energy from the environment. An example is the one developed by Enocean [62]. These sensors can produce 50 miliwatts, with variations of 3 degrees. Embedded EnOcean radios transmit around 300 meters in a free field or 30 meters in indoor (with walls and obstacles). Fig. 3 shows the schematic of free wireless sensors using ambient energy

There are other implementations published, but they appear in informative articles, so the technical depth is very low. On the other hand, there are motes developers, which products can be used in many types of applications.

D. Real deployments for military applications

The need of WSNs emerged during the Cold-War, especially for submarine surveillance and battlefield monitoring. But it was not available for the public use until the 1980s, when the first commercial distributed sensor network was developed by the Defense Advanced Research Projects Agency (DARPA). This WSN is called first-generation.

We can find several applications based on WSNs for service-based C4ISR (Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance) [63]. They can help or can be part of a military system due to some their embedded features like their rapid deployment, self-organization, and fault tolerance [64]. Wireless sensor networks can be an integral part of defense/offensive military systems [65].

Typical military applications of sensor networks are:

- Monitoring forces: reconnaissance of opposing forces and terrain and battlefield surveillance
- Monitoring of equipment and ammunition
- Targeting
- Battle damage assessment
- Nuclear, biological and chemical attack detection and reconnaissance
- "fog of war" clearing
- Space exploration
- Undersea monitoring

Rockwell Scientific has been working with the U.S. Marine Corps and U.S. Army to test and refine WSN performance in desert, forest, and urban terrain. For the urban terrain, WSNs are expected to improve troop safety as they clear and monitor intersections, buildings, and rooftops by providing continuous vigilance for unknown troop and vehicle activity [66].

E. General Purpose Indoor Location and Location tracking systems

Several position detection and location tracking systems based in wireless sensor networks exist. On the other hand, the growth of wireless networks caused multiple possibilities to calculate position. In indoor location, the main issue is that the system has to consider walls losses, interferences, multipath effect, humidity, temperature variations, etc.

One of the main applications of the Indoor Location and Location tracking systems is to locate people with Alzheimer or to locate disabled people with very little motion. An example of this kind of systems using the Zigbee technology was presented by Li-wei Chan et al. in [67].

MIT’s Cricket [68] is an indoor location system which uses ultrasounds to determinate user’s location that are moving in indoors, using mobile and fix nodes. Based on this system, interactive maps, personal location and access control to certain resources have been developed.

Jaime Lloret et al. [69] have analyzed two approaches where wireless sensors could find their position using WLAN technology inside a floor of a building. The first approach uses a training session and the position is based on a heuristic system using the training measurements. The second approach uses triangulation model with some fixed access points, but taking into account wall losses and signal variations. Both approaches are based on the Received Signal Strength Indicator (RSSI). Then, the authors compare their deployments with other deployments in [70].

Moreover, the same authors of this paper developed a new stochastic approach [71], which is based on a combination of deductive and inductive methods. In this system, wireless sensors can find their position using WLAN technology inside a floor of a building. Fundamental advantage of this system is that reduces the training phase in an indoor environment; but, without losing precision.

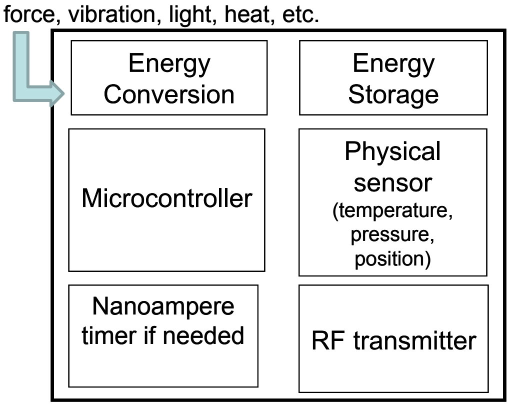

force, vibration, light, heat, etc.
Energy Conversion
Energy Storage
Microcontroller
Physical sensor (temperature, pressure, position)
Nanoampere timer if needed
RF transmitter

Figure 3. Schematic of free wireless sensors using ambient energy

V. Comparative Study

A comparative of the practical deployments presented in this paper is shown in table II. Not all the works presented have been added because some of them have been published without any technical depth and some main information was missing. The S letter means that the network has a small (few meters), M means medium size (from few meters to less than 1 Km), L means large size (more than 1 Km). In the communications type column we have used C when the information is sent to a server in order to process it and D when the information is distributed in the sensors. We added the “Single use” column in order to show if the deployment could be used in other environment or only in the environment where it has been created. When some information is not provided by the authors we used a dash “-“.

In order to show our conclusions from this comparative, we provide the following graphs, in order to show the global view of the state of the art of the practical deployments.

Fig. 4 shows that most of the deployments are medium, but there is not so much difference with the number of large and small size networks. Although many researchers study the scalability of the sensor networks, they are most used to cover less than 1 Km.

When wireless technologies are compared (see Fig. 5), we can observe that the most used technology has been Wi-Fi and the second one has been ZigBee. There are many implementations that use several types of technologies in their deployments.

Fig. 6 shows that there are more decentralized implementations than centralized implementations, but the difference is quite low.

Almost all the implementations have been performed for its specific purpose without the possibility of being able to be used in other implementations. Fig. 7 shows the measurements obtained.

Fig. 8 shows the comparison of the domains of the studied WSNs. We can see that WSNs are most used for health. The second domain has been agriculture and environment. This comparative gives us a view of where WSNs are most applied.

Finally, table III shows which sensor, software and hardware are used by some of the real deployments aforementioned. Not all the deployments appear because not all of them provide this information. In table III, N/A means that the information is not available.

VI. Limitations and Challenges

Generally, wireless sensor networks are deployed to work in hostile, remote and changeable environments, so it is necessary to incorporate security mechanisms when they are being designed. However, several limitations should be taken into account.

First, nodes can't be protected physically in these environments. For example, in a battlefield any enemy can capture them and analyze the information gathered, so, fundamental information could be extracted. The challenge for researchers and developers is to design resilient protocols or others solutions to provide security to these networks, even if one or several sensor nodes are compromised. Their goal is to ensure that, if a node is captured, sensitive information stored on it cannot be taken off easily.

Another issue to be considered is the random distribution of sensor nodes. When a sensor network is deployed in a hostile environment, it is done through a random distribution. So, in this case, to gather several encryption keys on nodes, in order to establish encryption among a group of neighbours, isn't possible, because the neighbourhood cannot be known a priori. So, the challenge is to design key agreement protocols that do not require neighbourhood between nodes, and also do not require storing encryption keys on sensors before the deployment.

On the other hand, the small size of the sensor nodes and the lack of wires is another important limitation of WSNs. Although these characteristics are fundamental for many applications, it involves limited resources, such as the energy, computational power, and storage resources. Besides, a large WSN contains hundreds to thousands nodes working by batteries, so, it is difficult both to replace and recharge them in some environments.

TABLE II. A COMPARATIVE OF REAL DEPLOYMENTS PRESENTED.

System	Network size	Technology used	Communication type	Single use	Domain
W. D. Hunt et al. [23]	S	RFID	C	No	Health (Early Cancer Detection)
C. Mc Caffrey, et al. [24]	S	RF (ISM-band)	D	No	Health (Gastrointestinal exploration)
SmartPill [26]	S	Wi-Fi	C	Yes	Health (diagnose of stomach disorders)
L. Schwiebert et al. [27]	S	Wi-Fi	D	Yes	Health (retina prosthesis)
Huan-Bang Li et al. [28]	S	Zigbee and Bluetooth	C	Yes	Health
B. P. Jarochowski et al. [29]	S	Zigbee	C	No	Health (Rehabilitation management system)
B. S. Hwang et al. [30]	M	Bluetooth	D	No	Health (monitoring system)
H. Kim et al. [32]	M	Zigbee	D	No	Health (monitoring system)
A. Wood et al. [33]	M	Zigbee	D	No	Health (Assisted-Living and Residential Monitoring)
Y.-J. Lin et al. [34]	M	RFID	C	No	Health (Assisted-Living and Elderly Nursing Home Monitoring)
M. Skubic [35]	S	Infrared	C	No	Health (In-Home Monitoring System)
F. Brunetti et al. [36]	M	Zigbee	D	No	Health (help for disabled or elderly people)
R. W. Gunderson et al. [38]	S	Ultrasonic signal	C	No	Health (wheelchair)
N. Firat et al. [40]	M	Wi-Fi and Zigbee	C	No	Health (monitoring system for assisting patients with cognitive impairments)
Chris Otto et al. [41]	S	Zigbee or Bluetooth and GPRS/3G data network connection	D	No	Health
A. Mainwaring et al. [42]	L	Zigbee	D	No	Environment (monitoring microclimates)
NICTOR [43]	M	Several	D	No	Environment (disasters detection)
D. Hughes et al. [44]	L	Zigbee	D	No	Agriculture (save water)
Irrigating When the Leaves Get Hot [45]	M	Infrared	C	Yes	Agriculture
S. A. Summers et al. [46]	L or M	Wi-Fi	D	No	Complicated situations (Fire-fighting and catastrophes)
J. Lloret et al. [47]	L	Wi-Fi	C	No	Environment (fire detection)
J. Lloret et al. [48]	L	Wi-Fi	C	No	Environment (fire detection)
ALERT [49]	L or M	Several	D	No	Environment (flood detection)
COUGAR [50]	M or S	Wi-Fi	C	No	Environment (snapshot and long-running queries)
EOFS [51]	L	Wi-Fi	D	No	Environment (forecast and pollution)
H. Song et al. [53]	L	GPS	D	No	Vehicle Anti-Theft System
C. Sharp et al. [54]	M	GPS	D	No	Vehicle Tracking and Autonomous Interception
Jenna Burrell et al. [55]	M	Wi-Fi	D	No	Industry (ethnographic research)
MESHNETICS [57]	M	Zigbee	C	No	Industry (parking)
Nedap's Wireless Space Count System [58]	L	Wi-Fi	C	No	Industry (parking)
S. Kim [59]	S	-	C	No	Industry
T. Kevan [60]	L	Zigbee and Wi-Fi	C	No	Industry (Shipboard Machine Monitoring)
A. Sukumaran [61]	S	Wi-Fi	C	Yes	Art
C4ISR [63]	L or M	GSM	D	No	Military (several applications)
M. Winkler [64]	L	GPS	C	No	Military (several applications)
D. N. Ngo [65]	L	Several	C	No	Military
K. Sohraby at al. [66]	L	-	D	No	Military
Li-wei Chan [67]	M	Wi-Fi	D	No	Indoor Location System
N. Bodhi [68]	M	RF (ISM-band)	D	No	Indoor Location System
M. Garcia et al. [69]	M	Wi-Fi	D	No	Indoor Location System
M. Garcia et al. [70]	M	Wi-Fi	D	No	Indoor Location system
J. Lloret et al. [71]	M	Wi-Fi	D	No	Indoor Location system

M: medium, L: large, S: small, D: decentralized, C: centralized, -: Information not provided

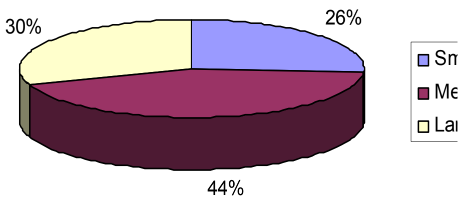


Figure 4. Network size comparison

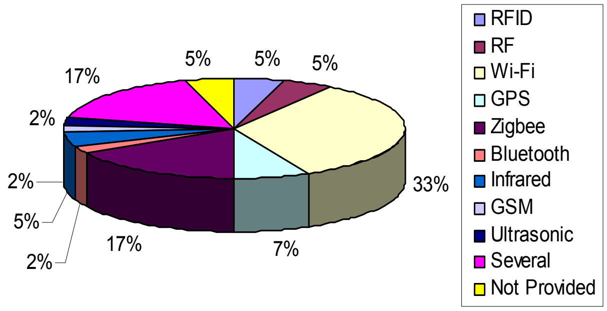


Figure 5. Technology comparison

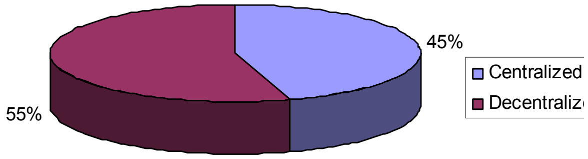


Figure 6. Communication type comparison

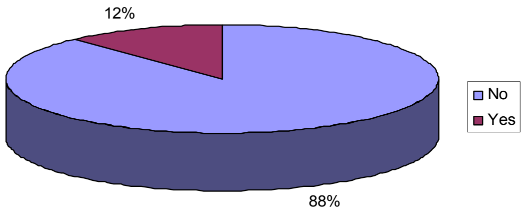


Figure 7. Single use comparison

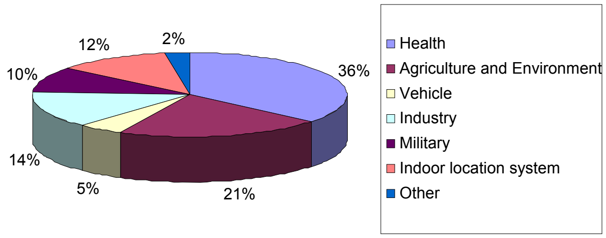


Figure 8. Domain comparison

TABLE III. SOFTWARE AND HARDWARE OF APPLICATIONS.

System	SENSOR	Software	Hardware
W. D. Hunt et al. [23]	Surface Acoustic Wave Sensor (SAW sensor)	Their detection mechanism is a mechanical, or acoustic, wave. So, it hasn't any software.	A SAW RFID tag or sensor has an antenna for receiving and propagating an RF signal, an input/output IDT electrically connected to the antenna and a dual track reflective IDT having a first track and a second track located adjacent and acoustically coupled to the input/output IDT.
L. Schwiebert et al. [27] Artificial retina	Smart sensor chip	N/A	An integrated circuit and an array of sensors. The integrated circuit is a multiplexing chip, operating at 40KHz, with on-chip switches and pads to support 10x10 connections. The circuit has both transmitted and receive capabilities.
B. S. Hwang et al. [30]	IR sensor	N/A	N/A
A. Wood et al. [33]	MicaZ	TinyOS	Microcontroller: ATMEGA 128 Wireless Transceiver or Wireless Technology: Transceptor 802.15.4/ZigBee Internal Memory: 4K RAM External Memory: 128K Flash

Y.-J. Lin et al. [34]	Bio-sensor	N/A	Contains electronic sensing devices with output electronic signal, a process unit which partially processes digitalized data and transmits required data, and a communication unit to communicate.
F. Brunetti et al. [36]	Inertial sensors	N/A	N/A
N. Firat et al. [40]	MicaZ	TinyOS	Microcontroller: ATMEGA 128 Wireless Transceiver or Wireless Technology: Transceptor 802.15.4/ZigBee Internal Memory: 4K RAM External Memory: 128K Flash
Chris Otto et al. [41]	Wireless pulse oximeter sensors, wireless ECG sensors, and triaxial accelerometer motion sensors.	N/A	N/A
A. Mainwaring et al. [42]	Mica2	TinyOS	Microcontroller: ATMEGA 128L Wireless Transceiver or Wireless Technology: Chipcon 868/916 MHz Internal Memory: 4K RAM External Memory: 128K Flash
NICTOR [43]	Nictor	Nictor Core	Sensor and Actuation Hardware: 4 Digital Inputs 5V or 24V logic levels individually selectable 2 Digital Outputs Switched relay outputs (0-240V, 30W) 2 Analog Inputs Configurable as 0-5V,0-10V, 4-20mA Common Mode Voltage rejection (in excess of +/- 30V)
D. Hughes et al. [44]	Mica2	TinyOS	Microcontroller: ATMEGA 128L Wireless Transceiver or Wireless Technology: Chipcon 868/916 MHz Internal Memory: 4K RAM External Memory: 128K Flash
S. A. Summers et al. [46]	Mica2	TinyOS	Microcontroller: ATMEGA 128L Wireless Transceiver or Wireless Technology: Chipcon 868/916 MHz Internal Memory: 4K RAM External Memory: 128K Flash
J. Lloret et al. [48]	Linksys WRT54GL	Linux	Microcontroller: Broadcom BCM5352E at 200 Mhz processor Wireless Transceiver or Wireless Technology: IEEE 802.11g and IEEE 802.11b Internal Memory: 16MB RAM External Memory: 4MB Flash
COUGAR [50]	Mica2	TinyOS	Microcontroller: ATMEGA 128L Wireless Transceiver or Wireless Technology: Chipcon 868/916 MHz Internal Memory: 4K RAM External Memory: 128K Flash
H. Song et al. [53]	Mica2	TinyOS	Microcontroller: ATMEGA 128L Wireless Transceiver or Wireless Technology: Chipcon 868/916 MHz Internal Memory: 4K RAM External Memory: 128K Flash
C. Sharp et al. [54]	Mica2	TinyOS	Microcontroller: ATMEGA 128L Wireless Transceiver or Wireless Technology: Chipcon 868/916 MHz Internal Memory: 4K RAM External Memory: 128K Flash
Jenna Burrell et al. [55]	Mica2	TinyOS	Microcontroller: ATMEGA 128L Wireless Transceiver or Wireless Technology: Chipcon 868/916 MHz Internal Memory: 4K RAM External Memory: 128K Flash
S. Kim [59]	Mica2	TinyOS	Microcontroller: ATMEGA 128L Wireless Transceiver or Wireless Technology: Chipcon 868/916 MHz Internal Memory: 4K RAM External Memory: 128K Flash
T. Kevan [60]	Intel motes	N/A	N/A

N/A: Information not available, OS: Operating System

The communication between sensor nodes consumes most part of the sensor energy, much more than sensing and computation. Any encryption algorithm to guarantee a secure communication introduces a communication overhead between nodes, because it is necessary to exchange more messages, for example for authentication, initialization and encryption data. So, the adoption of strong and computationally expensive cryptographic algorithms, such as the RSA public key algorithm, is not an efficient idea. Instead, symmetric encryption algorithms are used because they don't require as computational power as asymmetric encryption. However, providing security with symmetric encryption is limited. Therefore, another challenge for researchers is to design other security solutions to cover the weaknesses of symmetric encryption and so, establishing a secure communication among the nodes participating in a communication.

The limited capability of sensor nodes for storage is another problem for security algorithms. Each sensor node needs to store as different keys as sensor nodes network has. However, when a network is composed by a large number of sensor nodes, it requires a lot of memory, which probably cannot be provided. If a single encryption key common to all nodes is used, an enemy could compromise the whole network by compromising only a single node. So, the challenge to make up for storage restrictions is to design security protocols that minimize the number of keys must be used to provide adequate protection to the network. There many theoretical works that propose security systems, but we have not found any wireless sensor network deployment with security published.

VII. Conclusion

WSNs are needed to improve the quality of life of the disabled people. They are also a basic piece of the future medical applications. These reasons make WSNs a main research area for many research groups. WSNs help the humans to control, examine and survey places that they are not able to do because it is very difficult to achieve the place or because the human is not able to measure by itself.

Actually, more research is needed and developments of new sensors in order to get cheaper products and make them affordable. Lower prizes will help to introduce WSNs in our life.

All surveys on WSNs found in the literature show simulations and network tests in reduced and controlled environments, but this survey shows real implementations and practical deployments. There is not any survey published like the one presented in this paper.

We have compared the real deployments studied in this paper in terms of network size, technology used, the communication type, if it has being built for a single use and the domain where it is applied.

Acknowledgment

Authors want to thank Professor Petre Dini from Concordia University, Canada, because of his valuable recommendations and support to improve this paper.

References

[1] Diana Bri, Miguel Garcia, Jaime Lloret, and Petre Dini, "Real Deployments of Wireless Sensor Networks", Sensorcomm 2009, Third International Conference on Sensor Technologies and Applications, pp.415-423. Greece, 18-23 June 2009.

[2] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A Survey on Sensor Networks", IEEE Communications Magazine, vol. 40, issue: 8, pp.102-114. August 2002.

[3] Kemal Akkaya and Mohamed Younis, "A survey on routing protocols for wireless sensor networks", Ad-Hoc Networks, vol. 3, issue: 3, pp.325-349. May 2005.

[4] N. Xu, "A Survey of Sensor Network Applications", University of Southern California, technical report. New York, 2002.

[5] C. F. García-Hernández, P. H. Ibargüengoytia-González, J. García-Hernández, and J. A. Pérez-Díaz, "Wireless Sensor Networks and Applications: a Survey", International Journal of Computer Science and Network Security, vol.7, issue: 3. March 2007.

[6] I. Khemapech, I. Duncan, and A. Miller, "A survey of wireless sensor networks technology", 6th Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking & Broadcasting. Liverpool. UK, June 2005.

[7] J. Beutel, M. Dyer, M. Hinz, L. Meier, and M. Ringwald, "Next-generation prototyping of sensor networks". 2nd International Conference on Embedded Networked Sensor Systems. Baltimore, MD, USA, November 03-05, 2004.

[8] Crossbow website, at http://www.xbow.com/ [June 15, 2010]

[9] Moteiv website, at http://www.sentilla.com/moteiv-transition.html/ [June 15, 2010]

[10] Tiny Node website, at http://www.tinynode.com/ [June 15, 2010]

[11] J. Lifton, D. Seetharam, M. Broxton, and J. Paradiso, "Pushpin Computing System Overview: A

Platform for Distributed, Embedded, Ubiquitous Sensor Networks", Pervasive 2002, Proceedings of the First International Conference on Pervasive Computing, pp.139 – 151. Zurich, Switzerland, August 26-28 2002.

[12] R. Barr, J. C. Bicket, D. S. Dantas, B. Du, T. W. D. Kim, B. Zhou, and E. G. Sirer, "On the need for system-level support for ad hoc and sensor networks", Operating Systems Review, vol. 36, pp. 1–5. April 2002.

[13] Qing Cao, T. Abdelzaher, J. Stankovic, and Tian He, "The LiteOS Operating System: Towards Unix-Like Abstractions for Wireless Sensor Networks", IPSN 2008, Proceedings of the 7th international conference on Information processing in sensor networks table of contents, pp.233-244. St. Louis, Missouri, USA, 22-24 April 2008.

[14] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. "TinyOS: An operating system for wireless sensor networks", Ambient Intelligence, Springer, pp.115-148. New York 2004.

[15] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesC language: A holistic approach to networked embedded systems", SIGPLAN, Special Interest Group on Programming Languages, vol. 38, issue: 5, pp. 1-11. May. 2003.

[16] A. Dunkels, O. Schmidt, T. Voigt, and M. Ali, "Protothreads: simplifying event-driven programming of memory-constrained embedded systems", Sensys'06, Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, pp.29-42. Colorado, USA, October 31 - November 03, 2006.

[17] T. Henzinger, B. Horowitz, and C. M. Kirsch, "Giotto: A time-triggered language for embedded programming", Proceedings IEEE, vol. 91. January 2003.

[18] Wei Ye, John Heidemann, and Deborah Estrin, "An Energy-Efficient MAC protocol for Wireless Sensor Networks", Proceedings of the IEEE Infocom, pp. 1567-1576. New York, USA, June 2002.

[19] J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile, "IEEE 802.15.4: A Developing Standard for Low-Power Low-Cost Wireless Personal Area Networks", IEEE Network, Vol. 15, pp. 12-19. September/October 2001.

[20] ZigBee Alliance, "Zigbee specification", Technical Report Document 053474r06, Version 1.0. June 2005.

[21] M. J. Tierney, J. A. Tamada, R. O. Potts, R. C. Eastman, K. Pitzer, N. R. Ackerman, and S. J. Fermi, "The GlucoWatch® biographer: a frequent automatic and noninvasive glucose monitor", Annals of Medicine, vol. 32, Issue: 9, pp.632-641. 2000.

[22] J. C. Puyana, B. R. Soller, S. Zhang, et al. "Continuous measurement of gut pH with near infrared spectroscopy during hemorrhagic shock", The Journal of Trauma: Injury, Infection, and Critical Care, vol. 46, Issue: 1, pp.9-15. January 1999.

[23] W. Soofi, "Nanoscale Surface Acoustic Wave Sensors for Early Cancer Detection", National Nanotechnology Infrastructure Network Research Experience for Undergraduates. California 2005.

[24] C. Mc Caffrey, O. Chevalerias, C. O'Mathuna, and K. Twomey, "Swallowable-Capsule Technology". IEEE Pervasive Computing, vol. 7, issue: 1, pp.23-29. January-March 2008.

[25] Y. Zhang, C. K.C. Lui, and B.L. Lu., "Wireless Biomedical Sensing", Wiley Encyclopedia of Biomedical Engineering. John Wiley & Sons, Inc.

[26] The SmartPill GI Monitoring System, at http://www.smartpillcorp.com/ [June 15, 2010]

[27] L. Schwiebert, S. K. S. Gupta, and J. Weinmann, "Research challenges in wireless networks of biomedical sensors", MOBICOM'01, Proceedings of the 7th annual international conference on Mobile computing and networking, pp.151-165. Rome, Italy, July 2001.

[28] Huan-Bang Li, Ken-ichi Takizawa, Bin Zhen, and Ryuji Kohno, "Body Area Network and Its Standardization at IEEE 802.15.MBAN", 16th IST Mobile and Wireless Communications Summit, pp.1-5. Budapest, Hungary, July 2007.

[29] Bartosz P. Jarochowski, SeungJung Shin, DaeHyun Ryu, and HyungJun Kim, "Ubiquitous Rehabilitation Center: An Implementation of a Wireless Sensor Network Based Rehabilitation Management System", ICCIT'07, Proceedings of the International Conference on Convergence Information Technology, pp.2349-2358. Gyeongju, Republic of Korea, 21-23 November 2007.

[30] B. S. Hwang, J. M. Choi, and K. S. Park, "A novel method for unobtrusive measurement of indoor activities using sensor-based monitoring system", ITAB 2006, Proceedings of the International Special Topic Conference on Information Technology in Biomedicine. Ioannina - Epirus, Greece, 26-28 October 2006.

[31] Bluetooth CIG, "Specification of the Bluetooth system", Version 1.1, at http://www.bluetooth.com [June 15, 2010]

[32] H. Kim, B. Jarochowski, and D. Ryu, "A Proposal for a Home-Based Health Monitoring System for the Elderly or Disabled", Lecture Notes in Computer Science, Springer Berlin, vol. 4061/2006, pp.473-479. 2006.

[33] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, and J. Stankovic, “ALARM-NET: Wireless Sensor Networks for Assisted-Living and Residential Monitoring”. Technical Report CS-2006-11, Department of Computer Science, University of Virginia. 2006.

[34] Yaw-Jen Lin, Mei-Ju Su, Sao-Jie Chen, Suh-Chin Wang, Chiu-I Lin, and Heng-Shuen Chen, “A Study of Ubiquitous Monitor with RFID in an Elderly Nursing Home”, MUE’07, Proceedings of the International Conference on Multimedia and Ubiquitous Engineering, pp.336-340. Seoul, Korea. April 2007.

[35] M. Skubic, “Assessing Mobility and Cognitive Problems in Elders”, AAAI 2005 Fall Symposium, Workshop on Caring Machines: AI in Eldercare. Arlington, Virginia, 4-6 November 2005.

[36] F. Brunetti, J.C. Moreno, A.F. Ruiz, E. Rocon and J.L. Pons. “A new platform based on IEEE802.15.4 wireless inertial sensors for motion caption and assessment”, EMBS’06, Proceedings of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp.6497-6500. New York, USA, August 30-September 3, 2006.

[37] Holly A. Yanco, Anna Hazel, Alison Peacock, Suzanna Smith, and Harriet Wintermute, “Initial Report on Wheelesley: A Robotic Wheelchair System”, Proceedingss of the Workshop on Developing AI Applications for the disabled, held at the International Joint Conference on Artificial Intelligence. Montreal, Canada, August 1995.

[38] R. W. Gunderson, S. J. Smith, and B. A. Abbott, “Applications of virtual reality technology to wheelchair remote steering systems”, ECDVRAT’96, Proceedings of the 1st European Conference on Disability, Virtual Reality and Associated Technologies. Maidenhead, UK, 8-10 July 1996.

[39] V. Santos, P. Bartolomeu, J. Fonseca, and A. Mota, “B-Live-A Home Automation System for Disabled and Elderly People”, SIES’07, Proceedings of the International Symposium on Industrial Embedded Systems. Lisbon, Portugal, July 2007.

[40] N. F. Ince, C. Min, A. H. Tewfik, and David Vanderpool, “Detection of Early Morning Daily Activities with Static Home and Wearable Wireless Sensors”, EURASIP Journal on Advances in Signal Processing, vol. 2008. Hindawi, 2008.

[41] Chris Otto, Aleksandar Milenković, Corey Sanders, and Emil Jovanov, “System architecture of a wireless body area sensor network for ubiquitous health monitoring”, Journal of Mobile Multimedia, vol. 1, issue: 4, pp.307-326. 2006.

[42] A. Mainwaring, R. Szewczyk, J. Anderson, and J. Polastre, “Habitat Monitoring on Great Duck Island”, WSNA 2002, Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications. Atlanta, Georgia, USA, September 2002.

[43] NICTOR, National ICT Australia, at http://nicta.com.au/__data/assets/pdf_file/0003/2919/090527NICTOR1.pdf [June 15, 2010]

[44] D. Hughes, P. Greenwood, G. Blair, G. Coulson, F. Pappenberger, P. Smith, and K. Beven, “An intelligent and adaptable grid-based flood monitoring and warning system”, Proceedings of the UK eScience All Hands Meeting. Nottingham, UK, 10-13 September 2006.

[45] Irrigating When the Leaves Get Hot, at http://www.activefarming.org/irrigating-when-the-leaves-get-hot [June 15, 2010]

[46] S. A. Summers, “Wireless Sensor Networks for Fire fighting and Fire Investigation”, UCCS University of Colorado at Colorado Springs, CS526 Project. 2006.

[47] Jaime Lloret, Diana Bri, Miguel Garcia, and Pedro V. Mauri, “A Content Distribution Network Deployment over WLANs for Fire Detection in Rural Environments”, HPDC 2008, Proceedings of the International Symposium on High Performance Distributed Computing. Boston, MA, USA, 23-27 June 2008.

[48] Jaime Lloret, Miguel Garcia, Diana Bri and Sandra Sendra, “A Wireless Sensor Network Deployment for Rural and Forest Fire Detection and Verification”, Sensors journal, vol. 9, issue: 11, pp.8722-8747. October 2009.

[49] ALERT systems, at: http://www.alertsystems.org/ [June 15, 2010]

[50] Cougar: The Sensor Network is the Database, at http://www.cs.cornell.edu/boom/2002sp/extproj/www.cs.cornell.edu/database/cougar/default.htm [June 15, 2010]

[51] D. C. Steere, A. M. Baptista, D. McNamee, C. Pu, and J. Walpole, “Research Challenges in Environmental Observation and Forecasting Systems”, Proceedings of the 6th International Conference on Mobile Computing and Networking, pp.292-299. Boston, Massachusetts, 2000.

[52] G. Manes, R. Fantacci, F. Chiti, M. Ciabatti, G. Collodi, D. Di Palma, and A. Manes, “Enhanced System Design Solutions for Wireless Sensor Networks applied to Distributed Environmental Monitoring”, Proceedings of the 32nd IEEE Conference on Local Computer Networks, pp.807–814. 15-18 October 2007.

[53] Hui Song, Sencun Zhu, and Guohong Cao, “SVATS: A Sensor-Network-Based Vehicle Anti-Theft System”, Proceedings of the 27th Conference on

Computer Communications, pp.2128-2136. 13-18 April 2008.

[54] C. Sharp, S. Schaffert, A. Woo, N. Sastry, C. Karlof, S. Sastry, and D. Culler, “Design and implementation of a sensor network system for vehicle tracking and autonomous interception”, Proceedings of the 2nd European Workshop on Wireless Sensor Networks, pp.93-107. 2005.

[55] Jenna Burrell, Tim Brooke, and Richard Beckwith, “Vineyard Computing: Sensor Networks in Agricultural Production”. IEEE Pervasive Computing, vol. 3, issue: 1, pp.38-45. January-March 2004.

[56] Heemin Park, Jeff Burke, and Mani B. Srivastava, “Design and implementation of a wireless sensor network for intelligent light control”, IPSN/SPOTS 2007, Proceedings of the 6th international conference on Information processing in sensor networks, pp.370–379. Cambridge, Massachusetts, USA, 2007.

[57] Meshnetics, easy wireless for things, at http://www.meshnetics.com/zigbee-applications/ [June 15, 2010]

[58] Nedap’s Wireless Space Count System, at https://fjallfoss.fcc.gov/oetcf/eas/reports/ViewExhibitReport.cfm?mode=Exhibits&RequestTimeout=500&calledFromFrame=N&application_id=318778&fcc_id=%27CGDSENSNODE%27 [June 15, 2010]

[59] Kim, S., Pakzad, S., Culler, D.E., Demmel, J., Fenves, G., Glaser, S., and Turon, M., “Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks”, IPSN 2007, Proceedings of the 6th International Conference on Information Processing in Sensor Networks. Cambridge, MA, April 2007.

[60] Tom Kevan, “Shipboard Machine Monitoring for Predictive Maintenance”, at http://www.sensorsmag.com/sensors-mag/shipboard-machine-monitoring-predictive-maintenance-715 [June 15, 2010]

[61] Ashok Sukumaran, Park View Hotel, Ars Electronica, at http://artintelligence.net/review/?p=189 [June 15, 2010]

[62] Intelligent Wireless Sensors, at http://www.enocean.com/fileadmin/redaktion/pdf/white_paper/wp_energyforfree_en.pdf [June 15, 2010]

[63] Ericsson, “C4ISR for Network Oriented Defense”, White Paper. October 2006.

[64] Michael Winkler, Klaus-Dieter Tuchs, Kester Hughes, and Graeme Barclay, “Theoretical and practical aspects of military wireless sensor networks”, JTIT, Journal of Telecommunications and Information Technolog, vol. 2. 2008.

[65] Damian N. Ngo., “Deployment of 802.15.4 Sensor Networks for C4ISR Operations”, Master's thesis, Naval Postgraduate School Monterey, CA. June 2006.

[66] Kazem Sohraby, Daniel Minoli, and Znati Taieb, “Wireless sensor networks, Technology, Protocols, and Applications”, Wiley Interscience publication, 2007.

[67] Li-wei Chan, Ji-rung Chiang, Yi-chao Chen, Chia-nan Ke, Jane Hsu, and Hao-hua Chu, “Collaborative localization: enhancing wifi-based position estimation with neighbourhood links in clusters”, Pervasive 2006, Proceedings of the 4th International Conference on Pervasive Computing. Dublin, Ireland, May 2006.

[68] The Cricket Indoor Location System, at http://people.csail.mit.edu/rudolph/Teaching/Articles/bodhi-thesis.pdf [June 15, 2010]

[69] Miguel Garcia, Carlos Martinez, Jesus Tomas, and Jaime Lloret, “Wireless Sensors self-location in an Indoor WLAN environment”, SENSORCOMM 2007, Proceedings of the 1st International Conference on Sensor Technologies and Applications. Valencia, Spain, October 2007.

[70] Jaime Lloret, Miguel Garcia, Fernando Boronat, and Jesus Tomás, “The Development of Two Systems for Indoor Wireless Sensors Self-location”, Ad Hoc & Sensor Wireless Networks: An International Journal, vol. 8, issue 3-4, pp.235-258. June 2009.

[71] Jaime Lloret, Jesus Tomás, Miguel Garcia, and Alejandro Cánovas, “Hybrid Stochastic Approach for Wireless Sensors Self-Location in Indoor Environments”, Sensors journal, vol. 9, issue: 5, pp.3695-3712. May 2009.

Underwater Wireless Sensor Networks: Efficient Localization Schemes using SemiDefinite Programming

Bo Dong and Ahmed M. Mahdy
Department of Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX USA
bdong@tamucc.edu and ahmed.mahdy@tamucc.edu

***Abstract*— Location awareness and distance estimation cannot be underestimated in wireless sensor networks especially for the marine environment. State-of-the-art localization research and approaches have been thoroughly reviewed to illustrate the challenges imposed by inaccurate distance measurements and noisy backgrounds. Compared to existing localization approaches, the SemiDefinite Programming approach delivers accurate distance measurements even in hostile backgrounds. In this paper, two localization schemes, namely Fixed-Position, and Magnified-Range, are proposed. The main idea behind the Fixed-Position scheme is to improve connectivity of the whole system. Magnified-Range is mainly based on the fact that anchors are special nodes that suffer less from energy constraints. Therefore, it is possible to magnify their radio range without impacting the energy consumption of the network. Magnified-Range differentiates between the radio range of anchors and regular sensor nodes. Performance evaluation and simulation results have shown that the proposed schemes offer robust localization. In fact, the Magnified-Range scheme improves localization accuracy by 20%. Future research will focus on three main challenges, namely 3-D simulations, hybrid localization systems, and real-life demonstrations.**

Keywords-marine wireless sensor networks; underwater communication; localization; GPS; SemiDefinite Programming.

I. INTRODUCTION

The majority of the earth's surface is covered with water. As more research is conducted on underwater systems, data collection and environment monitoring become major components. This raises the need for an effective way to collect data and monitor the environment. Underwater wireless sensor networking offers an unmatched option. The characteristics of the underwater environment present researchers with many challenges, especially developing effective sensor communication and localization techniques. In terrestrial wireless sensor networks, the nodes use Radio Frequency (RF) to establish the communication infrastructure. In underwater environments, due to water absorption, RF does not deliver the same performance.

Compared to radio waves, sound has superior propagation characteristics in water, making it the preferred technology for underwater communications. However, since GPS may not work in underwater environments, acoustic signals bring many challenges to underwater sensor applications that require effective localization. Hence, there is a need to develop novel localization schemes that work well in the marine environment.

For the past few years, localization has become an indispensable factor of wireless sensor networks especially in tracking systems and environment monitoring. For many WSN applications, such as habitat monitoring, it is necessary to describe where the critical events occur. To obtain the location of sensor nodes, one can equip lightweight GPS receivers on all sensors but this significantly increases network cost and may defeat the whole objective of a wireless sensor network. A possible solution is to equip a few sensors with GPS and design efficient algorithms to estimate the position of other sensors using range and bearing information between neighboring nodes [2] [3] [4]. However, this option will not work in such environments since GPS receivers cannot reliably operate in such conditions. Moreover, GPS is a costly solution for WSN localization systems. Therefore, the need for a GPS-free localization system that can efficiently satisfy the underwater requirements cannot be underestimated.

This paper is organized as follows. In Section II, an introduction to acoustic underwater sensor networks is presented. In Section III, GPS-free localization in underwater WSN is discussed. Section IV describes the architecture of the network. Section V presents the proposed localization schemes and the performance evaluation. Section VI discusses future research directions. The paper is concluded in Section VII.

II. ACOUSTIC UUNDERWATER WIRELESS SENSOR NETWORKS

Underwater acoustic propagation depends on many factors that make designing an underwater wireless sensor network challenging. In the following, we present major factors that may impact the design process.

a. Bandwidth: The acoustic band underwater is limited due to absorption; most acoustic systems operate below 30 kHz. According to [5], no research or commercial system can exceed 40km × kb/s as the maximum attainable range×rate product.

b. Propagation Delay: The speed of RF is 3×10^8 m/s while the acoustic signal propagation speed in an underwater acoustic channel is about 1.5×10^3 m/s. The propagation delay in underwater is five orders of magnitude higher than the case with RF. The relatively low speed of sound causes multi-path propagation to stretch over time delay. It may greatly affect certain applications that require critical real-time communications.

c. Shadow Zones: Salinity, density and temperature variations of the water can influence acoustic communication. Temporary loss of connectivity is a major impact. This is evident in per the following formula [6]:

$$C = 1449.2 + 4.6T - 0.055T^2 + 0.00029T^3 + (1.34 - 0.01T)(S - 35) + 0.016Z \quad (1)$$

where, C speed of sound (m/s)
T temperature (deg C)
S salinity (practical salinity units “psu” equivalent to parts per thousand)
Z depth (m)

d. Energy: Power is a major issue for underwater environment due to the extreme difficulties in recharging such batteries. Unlike terrestrial WSN, UWSN cannot use solar energy to regenerate the power of the batteries.

e. Failure: Underwater sensors are more likely to suffer failure due to corrosion and other natural phenomena.

f. Attenuation: It is the reduction in amplitude and intensity of a signal. Attenuation at distance x is given as [7]

$$A(x) = x^k a^x \qquad (2)$$

where k is a spreading factor
a is frequency dependent term obtained as

$$a = 10^{(.\alpha(f))} \qquad (3)$$

where $\alpha(f)$ is absorption coefficient given by Thorp’s expression. The formula illustrates that attenuation is dependent on the frequency as well as distance.

A. Underwater vs.Terrestrial Wireless Sensor Networks

Although WSN and Underwater Wireless Sensor Networks (UWSN) are different, mainly due to the unique characteristics of water, certain aspects of WSN still apply to UWSN. In the following, we highlight major differences that affect the use of WSN techniques and algorithms in the marine environment:

a. UWSN primarily use acoustic signals while RF is the choice for WSN.

b. While terrestrial sensor nodes are expected to become increasingly inexpensive, underwater-ready equipment tend to be more expensive. This is due in large to transceivers complexity and the increased protection required for the hardware.

c. UWSN generally require more power. This is because UWSN use acoustic signals and usually cover larger geographical areas. Compared to acoustic signals, RF systems consume less power.

d. The connection of an acoustic signal may be interrupted by special underwater situations like shadow zones. Due to this fact, underwater systems may need to compensate using more complicated recovery techniques. This will lead to overhead.

e. Density: In terrestrial sensor applications, like tracking systems, sensors can be deployed densely. While an underwater sensor is more expensive than its terrestrial counterpart, it costs more to deploy UWSN densely. Not only that but it is usually more challenging to deploy a dense underwater network.

In fact, the aforementioned differences present clues on the development of new generation of UWSN. It is clear that there is a need for new types of cost-effective sensors. For example, research interests on developing nano sensors have been growing. Moreover, the deployed network ought to be highly reliable, so as to avoid failure of monitoring missions due to failure of single or multiple sensors. New power control algorithms for UWSN are needed. Despite the fact that there is many power control algorithms for wireless terrestrial networks [8][9]. However, these algorithms are not suitable for UWSN due to the underwater channel characteristics and significant propagation delays. Additionally, novel network protocols are vitally important in reducing power consumption and providing reliable connections using sparse underwater sensors.

B. Classes of Underwater Wireless Sensor Networks

According to [10], UWSN can be roughly classified into two broad categories:

a. Long-Term Non-Time-Critical Aquatic Monitoring

This class of UWSN is intended for long-time deployment where the collected data by the sensors is not real time. In this case, energy consumption is critical.

b. Short-Term Time-Critical Aquatic Exploration

Compared to long-term non-time-critical UWSN, this class of UWSN focuses on real-time data. Therefore, how to ensure efficient data transfer is a major issue when designing protocols for this type of networks. Unlike the long-term class, power consumption is not as critical.

Reference [11] classifies UWSN into three types: a) Static two-dimensional underwater acoustic sensor networks (UW-ASNs) which are most suitable for deep (i.e. bottom of the ocean) monitoring, b) Static three-dimensional UW-ASNs which are most suitable for ocean-column monitoring, and c) Three dimensional networks of Autonomous Underwater Vehicles (AUVs).

The main difference between the two classifications is the mobility aspect. In [10], long-term non-time-critical and short-term time-critical UWSN assume the ability of the sensors to move. Moreover, long-term and short-term do not distinguish between 2D and 3D. Obviously, there are some differences in protocol design.

In static two-dimensional underwater, all the sensors are fixed to the bottom of the ocean. The underwater sensor nodes are interconnected to one or more underwater sinks using acoustic signals. 2D UW-ASNs are commonly recommended for environmental monitoring.

Designing protocols for static three-dimensional underwater sensor networks, compared to the two-dimensional one, is relatively complex. The speed and propagation delay of acoustic signals differ according to water depth. This results in different power consumption levels depending on the depth of the sensor node. Clearly, it also complicates the design of efficient routing. Another challenge for this class of networks is to maintain the respective depth of the different sensors.

The three-dimensional networks of autonomous underwater vehicles may overlap with the long-term or short-term classes. “And one vital important design objective is to make them rely on local intelligence and less dependent on communications from online shores.” [11]

C. *Major Research Challenges in Uderwater Wireless Sensor Networks*

Power Consumption

As mentioned, underwater sensors, unlike terrestrial sensors, cannot use solar energy to recharge their batteries. It is also challenging to physically replace these sensors. A straightforward approach to resolve this problem is to self-generate energy using the sensors. Research on generating energy using current movement has been conducted in the last few years. Nonetheless, efficient routing protocols and novel communication technologies are greatly needed.

Communication Link

Underwater networks are dominated by acoustic signals for the aforementioned reasons. Acoustic signals bring a lot of challenges to the research arena, especially propagation delay and high error rates. Alternatives are sincerely sought. The search for better options is ongoing. According to [12], optical signals have been used successfully for sensors communications. This opens the door for future possibilities. Yet, optical signals have their own challenges especially with power consumption and how they compare to acoustic signals.

Distributed Localization and Time Synchronization

Location-awareness has become an essential characteristic for many of the underwater applications. For these systems, data without associated location information might be useless. Among many of the large-scale terrestrial WSN applications, GPS can be used to provide accurate locations and time synchronization. In GPS-free terrestrial applications, other techniques are used to calculate the distance. Once the distance is known, the position information can be calculated using algorithms such as SemiDefinite Programming (SDP) [13]. In UWSN, the position information can be calculated in the same way. However, it is very challenging to determine the distance between two sensors. GPS cannot be used since the satellite signal is weak for underwater. Further discussion is found in the next Section.

Routing Protocols

In UWSN, protocol design is tied to energy efficiency especially for long-term monitoring applications. Actually, there are numerous terrestrial WSN energy-efficient protocols already developed. However, due to signal nature and environment factors, most of them are not feasible underwater. Further discussion can be found in [14].

III. Localization in Underwater Wireless Sensor Netorks

A. *GPS-free Localization Schemes*

Localization in wireless sensor networks can be formulated into a graph realization problem. Given a graph G = (V, E) and sets of non-negative weights, $\{d_{ij} : (i,j) \in E\}$, the goal is to compute a realization of G in the Euclidean space R^d for a given low dimension d; to place the vertices of G in R^d such that the Euclidean distance between every pair of adjacent vertices(i, j) equals (or bounded) by the prescribed weight $d_{ij} \in E$.

GPS may not be an optimal choice for underwater systems. Therefore, a GPS-free scheme is needed for such systems. The characteristics of an underwater environment represent the significant difference between Underwater Wireless Sensor Networks (UWSN) and its counterpart. The schemes and network protocols of terrestrial Wireless Sensor Networks (WSN) cannot be used directly on UWSN. Therefore, GPS-free schemes used for terrestrial WSN localization applications cannot be used directly.

Normally, localization schemes can be classified into two categories: range-based schemes and range-free schemes. Range-based schemes use range measurements to calculate position information. Time of Arrival (TOA) is used widely as a method of distance estimation using propagation time of different kinds of signals. "GPS is a most basic localization system of TOA" [15]. The disadvantage of GPS is that it is costly and may not be suitable for some applications. To use a GPS system, the GPS receivers require the installation of expensive and energy-demanding hardware to rigorously synchronize with at least four satellites. Nonetheless, GPS systems do not work in certain settings, such as indoor and underwater environments. This is due to the fact that satellite signals are not able to go through buildings and seawater.

Another widely used range-based scheme is Time Difference of Arrival (TDOA). "TDOA measures range information using time difference between two kinds of signals, such as Radio Frequency (RF) and ultrasound" [16]. Cricket [17] is one of the existing commercial TDOA localization systems. Compared to TOA, TDOA does not need to precisely time synchronize all nodes. However, like TOA, it still needs expensive and power consuming hardware to emit two different kinds of signals. Hence, it may not be suitable for low-power sensor network applications. In addition, the average propagation distance of ultrasound, which is around 15-25 feet, is too short to satisfy large-scale applications. This brings many limitations to deploy a real TDOA-based application.

To complement the TDOA and TOA schemes, Angle of Arrival (AOA) has been developed to provide the two schemes with more accurate range information [3]. The idea of AOA is to allow the sensor nodes to estimate the angles between neighbors. However, like TOA and TDOA, AOA still requires costly hardware to estimate the angles. It is clear that the majority of range-based schemes rely on accurate range measurements. These measurements are easily affected by background noise. Therefore, how to conquer the impacts caused by noise is currently an active area of research in WSN localization.

Compared to range-based, the range-free schemes are more suitable for cost-effective situations. "Range-free estimates the location of sensor nodes either by exploiting the radio connectivity information among neighboring nodes, or by exploiting the sensing capabilities that each sensor node possesses" [18]. In [19], a centroid algorithm is proposed. Based on the number of received beacons broadcasted by anchors, a centroid model was established to calculate the positions of target sensor nodes.

Another outstanding solution for range-free is DV-HOP [3]. This work uses hop-count, the average distance per hop at each node, to compute the approximate position for the sensor nodes. Other algorithms use offline hop-distance estimations and neighbor information exchange to improve the accuracy of the position results.

Compared to the range-based scheme, the cost of equipment is cheaper and the physical factors have less impact on these algorithms. However, the results may be crude not reflecting the exact position information. Therefore, such schemes will only fit applications that do not require critically precise accuracy.

B. SemiDefinite Programming and Underwater Localization

The trace of a given matrix A, denoted by Trace (A), is the sum of the entries on the main diagonal of A. We use I, e and 0 to denote the identity matrix, the vector of all ones and the vector of all zeros, whose dimensions will be clear in the context. The inner product of two vectors p and q is denoted by $\langle p,\ q\rangle$. The 2-norm of a vector x, denoted by $\|x\|$, is defined by $\sqrt{\langle x,\ x\rangle}$. A positive SemiDefinite matrix X is represented by $X \succeq 0$.

In [20], the mathematical model of the sensor localization problem is described as follows. "There are n distinct sensor points in R^d whose localizations are to be determined, and other m fixed points (called anchor points) whose localizations are known as a_1, a_2,..., a_m. The Euclidean distance d_{ij} between the *i*th and *j*th sensor points is known if (*i, j*) ∈ N_x, and the distance $\bar{d}_{ik}$ between the *i*th sensor and *k*th anchor point is known if (i, k) ∈ N_a. Usually, $N_x = \{(i,j) : \|x_i - x_j\| = d_{ij} \le rd\}$ and $N_a = \{(i,k) : \|x_i - a_k\| = \bar{d}_{ik} \le rd\}$, where rd is a fixed parameter called radio range. The network localization problem is to find vector $x_i \in R^d$ for all *i* = 1, 2,..., *n* such that

$$\|x_i - x_j\|^2 = d_{ij}^2 \qquad \forall (i,j) \in N_x$$
$$\|x_i - a_k\|^2 = \bar{d}_{ik}^2 \qquad \forall (i,k) \in N_a \qquad [20]$$

Unfortunately, this problem is, in general, hard to solve even for d = 1. In 2004, the relaxation model of this math model was represented by a standard Full SemiDefinite Programming (FSDP) model, which is shown in the following equation. For simplicity, d is set to 2.

$$\text{(SDP)} \quad \text{minimize } 0 \cdot Z$$
$$\text{subject to} \quad Z_{(1,2),(1,2)} = I_2$$
$$(0; e_i - e_j)(0; e_i - e_j)^T \bullet Z = d_{ij}^2, \quad \forall (i,j) \in N_x$$
$$(-a_k; e_i)(-a_k; e_i)^T \bullet Z = \bar{d}_{ik}^2, \qquad \forall (i,k) \in N_a \qquad [20]$$
$$Z \succeq 0.$$

Here I_2 is the 2-dimensional identity matrix, 0 is a vector matrix of all zeros, and e_i is the vector of all zeros except a

1 at the *i*-th position. If a solution $Z = \begin{pmatrix} I_2 & X \\ X^T & Y \end{pmatrix}$ is of rank 2, or equivalently, $Y = X^T X$, then $X = \left[x_1,\ ...,\ x_n\right] \in R^{2\times n}$ is a solution to the sensor network localization problem. Here, Z is a (n + 2) symmetric matrix.

But this model does not deal with the noise factor; it just gets the feasible solutions for the sensor localization problem. A noise factor based SDP model is proposed in [20].

$$\begin{aligned}
&\min \sum e_{ij} + f_{ik} \\
s.t.\ \ & e_{ij} \geq (0;\ e_i - e_j)(0;\ e_i - e_j)^T \bullet Z - d_{ij}^2, && \forall (i,\ j) \in N_x \\
& e_{ij} \geq d_{ij}^2 - (0;\ e_i - e_j)(0;\ e_i - e_j)^T \bullet Z, && \forall (i,\ j) \in N_x \\
& f_{ik} \geq (-a_k;\ e_i)(-a_k;\ e_i)^T \bullet Z - \bar{d}_{ik}^2, && \forall (i,\ k) \in N_a \\
& f_{ik} \geq \bar{d}_{ik}^2 - (-a_k;\ e_i)(-a_k;\ e_i)^T \bullet Z, && \forall (i,\ k) \in N_a \\
& Z_{(1:2,1:2)} = I_2 \\
& Z\ \underline{\mathrm{f}}\ 0, && \forall (i,j) \in N_x
\end{aligned}$$

Using this model, even with its worst case scenario of inaccurate results affected by noise factor, this mathematical model still generates relatively accurate distance estimations even in a noisy background.

However, the FSDP model requires very long time to resolve large-scale position problems. In this case, tracking applications and real-time systems are impossible to use this approach. Therefore, in 2006, Edge-based SemiDefinite Programming (ESDP) model has been developed to reduce time complexity of FSDP [20].

$$\begin{aligned}
&\text{(SDP)}\ \ \text{minimize}\quad 0 \cdot Z \\
&\qquad\qquad\qquad \text{subject to}\quad Z_{(1,2),(1,2)} = I_2
\end{aligned}$$

$$\begin{aligned}
&(0;\ e_i - e_j)(0;\ e_i - e_j)^T \bullet Z = d_{ij}^2, && \forall (i,\ j) \in N_x \\
&(-a_k;\ e_i)(-a_k;\ e_i)^T \bullet Z = \bar{d}_{ik}^2, && \forall (i,\ k) \in N_a \\
&Z_{(1,2,i,j),(1,2,i,j)}\ \underline{\mathrm{f}}\ 0, && \forall (i,\ j) \in N_x
\end{aligned}$$

The only difference between FSDP and ESDP is the last constraint. The last constraint of ESDP only needs every four by four matrix to be a SemiDefinite matrix. On the other hand, FSDP works with a (n + 2) by (n + 2) matrix. ESDP will be much faster in resolving large-scale wireless sensor networks. Simulation results have proved this conclusion [21].

To use SDP, two types of data need to be collected. They are the accurate positions of anchors and the partial pair-wise distance measurements between some of the sensor nodes and the distances between sensor and anchor nodes.

SDP uses these two types of data to compute the position of every sensor node in the WSN. In practice, the number of anchors is at least three; however, more anchors can generate more accurate position results. Nonetheless, more anchors imply more cost. Therefore, how to deploy a limited number of anchors to generate accurate position results is a challenging issue in designing this localization system. According to [20], if all sensor nodes can connect to any anchor nodes, directly or indirectly, the solution for SDP must be bounded. Therefore, appropriate radio range should be adjusted depending on different situations in order to make sure all sensor nodes can connect to at least one anchor directly or indirectly.

In [20], it is mentioned that various techniques have been developed to address measurement uncertainties. Most of these methods are based on minimizing some global error functions, which can be different depending on the model of uncertainty. According to the type of optimization model being formulated, the characteristics and computation complexity vary. “Existing algorithms have limited success, even for small problem sizes” [22]. Moreover, in the real world, all the distance measurements inevitably have noises. Using this noisy distance information, we cannot get satisfactory results. However, SDP had been proved that it could generate accurate results even in extremely noisy backgrounds. More importantly, there is no special node(s) in the SDP approach. Hence, even if some of the nodes were destroyed, the localization system will still work and continue to generate relatively accurate results.

In [21], two future relaxations of SDP, NSDP and ESDP, have been proposed and tested. The performance of SDP approach and ESDP approach can be found in [13][21]. Further details and complete discussions on SDP can be found in [13][20][21].

C. Motivation

Recently, marine biology research has increasing needs of using the position information, especially in migration and distribution research such as red king crabs research. Most research is based on ship data collection. Biologists need to be sailing across the area utilizing a data receiver to collect data. This is very expensive and has many limitations, such as meteorological factors. Therefore, it is more convenient, if the biologists can get the data onshore, significantly reducing the cost of research. Migration and distribution research belong to a long-term non-time-critical research, normally at least one year. The design for this kind of system needs to focus on low energy consumption and stable communication. In this paper we propose a SemiDefinite Programming underwater localization system, which provides biologists with a convenient and cheaper way to collect data in marine research.

IV. Network Architecture

It is needed to collect anchors' position coordinates and the partial pair-wise distance measurements between the nodes for the proper operation of SDP-based localization systems. Partial pair-wise distances include the distances between sensors, and distances between anchors and sensors. SDP uses the collected data to compute the position of every sensor node. According to [13], three factors impact the accuracy of results from the SDP approach. The factors are the number of anchors, connectivity (number of reachable sensors in a sensor acoustic range), and background noise. More anchors, large connectivity and less noise lead to better accuracy and accurate position results. Among these three factors, it is possible to control the first two factors, but it is challenging to deal with the noise factor. Despite the performance gain, increasing the number of anchors and improving connectivity increase the cost of the system. Therefore, a balance between the result accuracy and cost needs to be weighed carefully.

The design of UWSN architecture needs to address three key problems to satisfy the requirements of SDP. First, how do anchor nodes get their position coordinates? This issue is a common challenge to a wide array of UWSN applications. Second is how to calculate pair-wise accurate distances between sensor nodes. Third is how to improve the connectivity of the system. The solution to the first problem is floating sensors. These nodes float on the water surface and have the ability to communicate using radio frequency (RF) and acoustic signals. These sensors may be utilized as anchor nodes. RF can be used for above-water communications including exchanging GPS information with other floating/anchor nodes and onshore base stations. The acoustic signal is used for communicating to the underwater sensor nodes. Therefore, floating sensors may act as sink nodes.

To resolve the second problem, round trip propagation of data packets is used to measure the distance between two sensor nodes. When underwater sensors are activated, they send distance-measure packets. If a neighbor within acoustic signal range gets the data packet, this neighbor sends a reply packet to the original sender node. The reply includes an arrival time stamp for the distance-measure packet. Also, it includes the depth, temperature and salinity data of the neighbor nodes. Then the data package is sent by the original sensor node to floating sensor nodes. During this process, the original sensor node adds its own depth, temperature and salinity data to the package. Finally, the collected data can be relayed to a base station for further distance calculations. The depth, temperature and salinity information is used to calculate the speed of the acoustic signal. Round-Trip Time is the total time from sending the distance-measure packet to receiving a reply. Therefore, the distance between a sensor node and its neighbor can be calculated as *speed of sound * Round-Trip-Time / 2*. However, there are underwater factors that can affect the round trip measurement consistency. Therefore, multiple group measurements are to be collected before transmitting to the floating sensor node. Standard deviation should be used to justify whether those samples in the group are accurate. If not, next group measurements will be tested until the smallest standard deviation group is found.

As stated earlier, connectivity is a major factor in achieving high accuracy. However, in certain applications such as tracking king crabs, there is no guarantee for enough sensors surrounding a target node to provide strong connectivity. Therefore, another type of sensors may be used for such systems; *Fixed-Position scheme*. The main idea is to improve connectivity of the whole system. Compared to other sensors, fixed-position nodes will not move with the target. Although the name is "fixed," it does not mean they are anchored at a certain position. In real applications, they can move within a certain area which will be defined in advance. They are deployed to evenly cover the monitored area and measure temperature, depth and salinity for calculating the speed of the acoustic signal. However, if the applications need to monitor the area in addition to tracking the target, those transmitter nodes are equipped with necessary sensors to measure other environment phenomena. The design of Fixed-Position scheme provides other benefits. Most importantly, it leads to a reduction in power consumption for all sensor nodes. This is because node connectivity must be above a certain level to guarantee the accuracy of position results. If the system does not have these transmitters, it should increase the power of acoustic signal to improve coverage range. In this case, all sensors will be subject to extra power consumption in order to complete the task.

The straightforward solution, to increase connectivity, is to increase all sensors' acoustic range. However, as mentioned above, this will lead to highly increased power consumption. Thus, it will reduce the system's lifetime. Since the power of floating anchors can be recharged, it is possible to only magnify the anchors' acoustic signal range (i.e. *Magnified-Range Scheme*). In this case, all regular sensor nodes will have the opportunity to connect to anchors directly or indirectly. Hence, magnified anchors' signal range will improve the probability to achieve more accurate results. A network architecture is illustrated in Figure 1.

V. Simulation and Performance Analysis

All simulations are based on a 2-D domain and all tests are solved by SeDuMi 1.1 of MatLab2007b on a MacBook Pro laptop with 2.16GHz Intel Core 2 Duo CPU, 2 GB 667 Mhz DDR2 Memory. In this simulation, we randomly generate the true position of all sensor locations in a square of 1 by 1.

To prove that the three factors, namely anchor number, connectivity and noise factor, significantly affect position results reported by the SDP approach, numerous simulations have been performed in [13][20][21][23]. The purpose of our simulation is to show that our proposed schemes are effective and achieve accurate localization. Moreover, issues in regard to how to deploy anchors in real applications will be discussed.

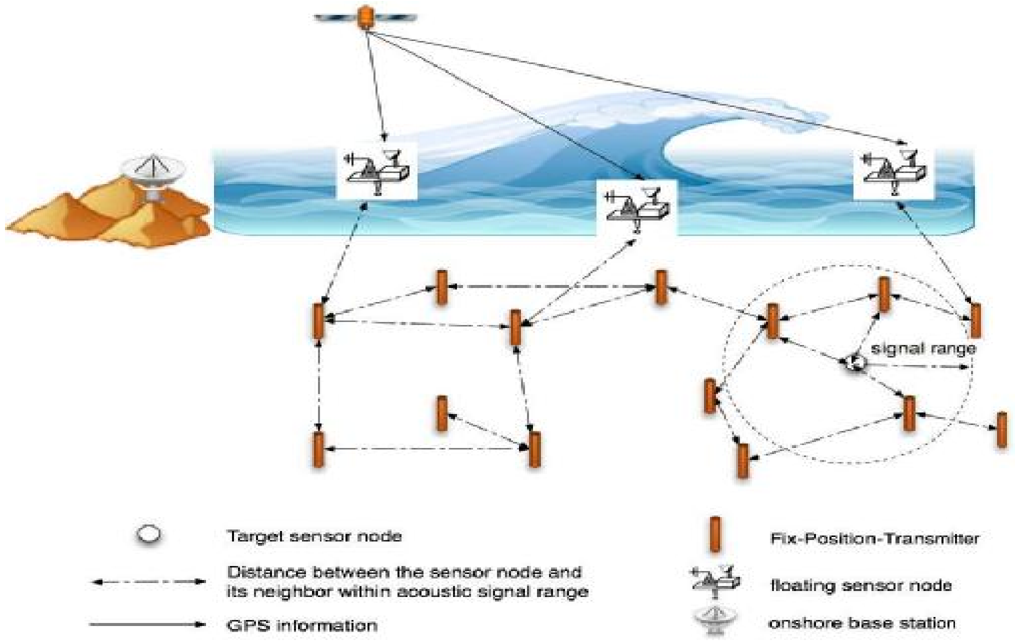


Figure 1. Network Architecture.

A. Error Estimations

In these simulations, three methods are used to estimate errors namely a) Error, b) Root Mean Square Deviation (RMSD) and c) Individual Trace. These errors often occur in real applications either due to the lack of information or the presence of noise, and are often difficult to detect since the true locations of sensors are unknown.

- **Error:** It is the distance between each pair of true position and estimate position. In these simulation settings and unlike real-life applications, it is possible to compare the true and estimate positions.

- **Root Mean Square Deviation (RMSD):** It measures the accuracy of the estimated positions {x_i : I = 1, …, n}. The formula is shown below. This method cannot be used in real-life applications since the true positions of the sensor nodes are unknown. This metric is only useful for simulation purposes. Lower RMSD means the global error is smaller.

$$RMSD = \frac{1}{\sqrt{n}}\left(\sum_{i=1}^{n}\left\|\overline{x}_i - x_i\right\|^2\right)^{1/2} \quad (4)$$

where, n is number of target sensor nodes
x_i is the true position of ith sensor nodes
x_i is the estimated position of ith sensor nodes.

- **Individual Trace:** When relaxing to the SDP model, change $Y = X^T X$ to $Y \; \underline{\mathrm{f}} \; X^T X$. This is equivalent to $Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} \underline{\mathrm{f}} \; 0$. Thus, $Y - X^T X$ represents the co-variance matrix of x_i, i = 1, …, n. Individual trace $Y_{ii} - \|x_i\|^2$,Which is also the variance of $\|x_i\|$. This helps detecting possible distance measure errors, and defect sensors in real application [13].

In the following, two kinds of chart are illustrated; position simulation charts and error estimation charts, respectively. In position simulation charts, the **red squares** indicate the position of anchors. The **green circles** present the true position of target sensor nodes. The **blue asterisks** show the estimated position of the target sensor nodes. The **blue lines** between green circles and blue asterisks represent the distance between true positions and estimated positions. In error estimation charts, the **red squares** represent the individual trace. The **blue circles** refer to errors.

B. Fixed-Position Scheme Simulation

Purpose: To test the Fixed-Position theory.

Analysis: Figure 2 shows the estimated results using 3 anchors, 30 sensor nodes and signal range of 0.3. It is clear that the results are very inaccurate. Instead of using a larger signal range to increase connectivity, we deploy 10 more sensor nodes in the area. The results are shown in Figure 3. Compared to Figure 2, the estimated results are much more accurate. The results in Figure 3 show a reduction in the global error from 0.162 to 0.039.

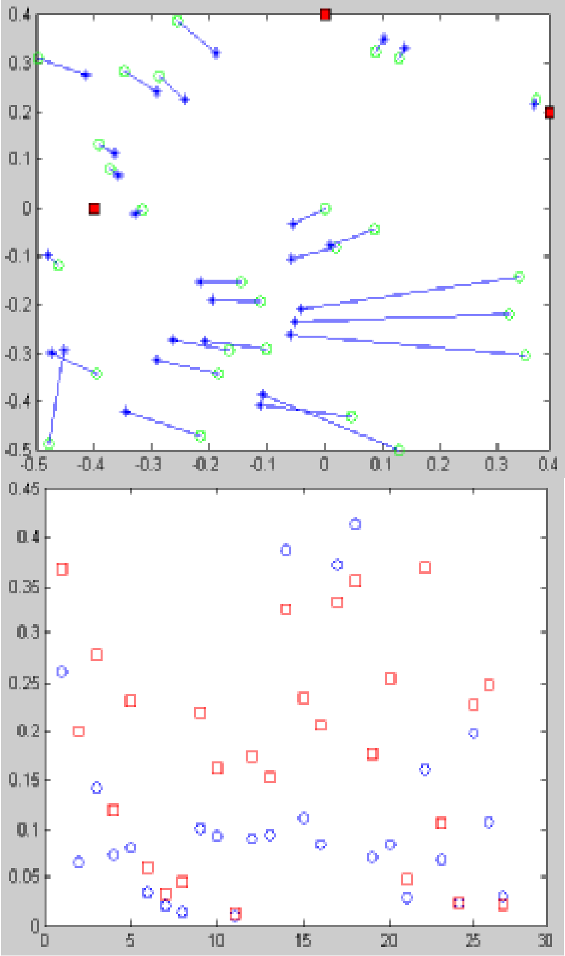


Figure 2. Fixed-Position Simulation 1 (RMSD = 0.162)
Condition: Anchor Number = 3; Sensor Number = 30; Acoustic Range = 0.3

C. *Magnified-Range Scheme Simulation*

Purpose: to test the performance of magnified-range theory.

Analysis: The difference between this simulation and the previous one is that the noise factor is set to 0.05. This is why the overall accuracy is lower than the results in Figure 3. The simulation in Figure 4 uses the same signal range for both; sensor nodes and anchors. While the simulation in Figure 5 uses 0.3 for the sensor signal range and 0.5 as anchor signal range. The simulation results show that the configuration in Figure 5 with magnified signal range improves the accuracy by around 20% compared to Figure 4.

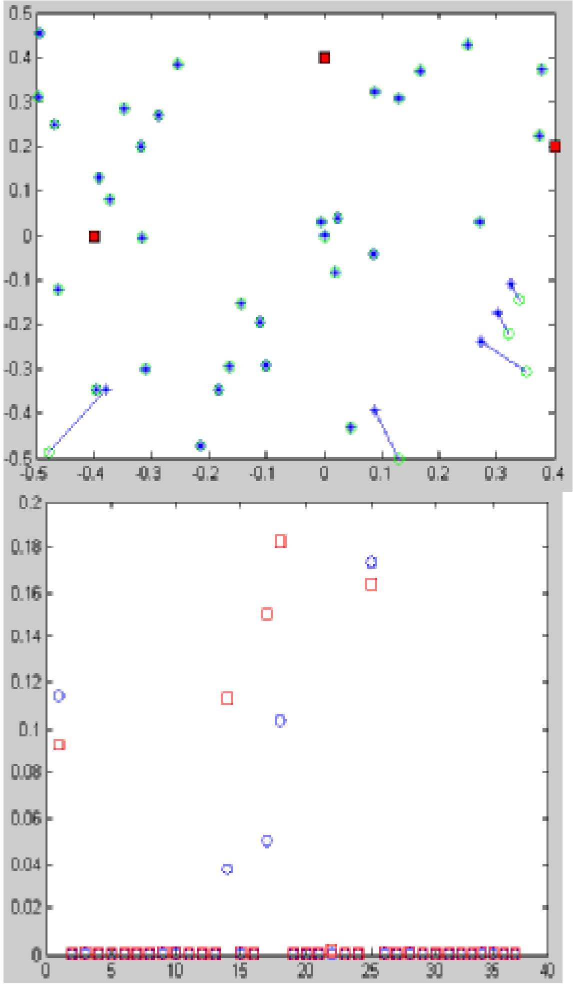


Figure 3. Fixed-Position Simulation 2 (RMSD = 0.039)
Condition: Anchor Number = 3; Sensor Number = 40; Acoustic Range = 0.3

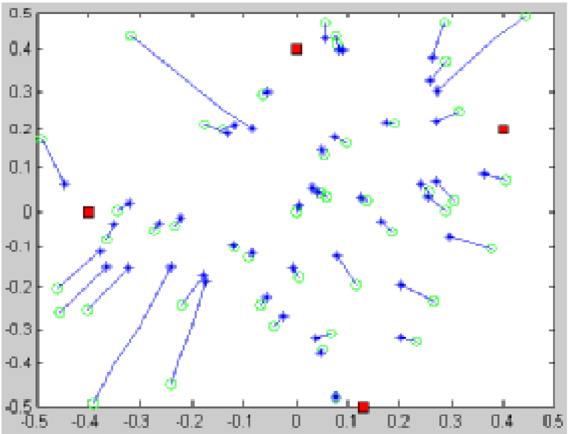


Figure 4. Magnified-Range Simulation (RMSD = 0.124)
Condition: Anchor Number= 4; Senor Number = 50; Noise Factor = 0.05; Senor Signal Range = 0.3

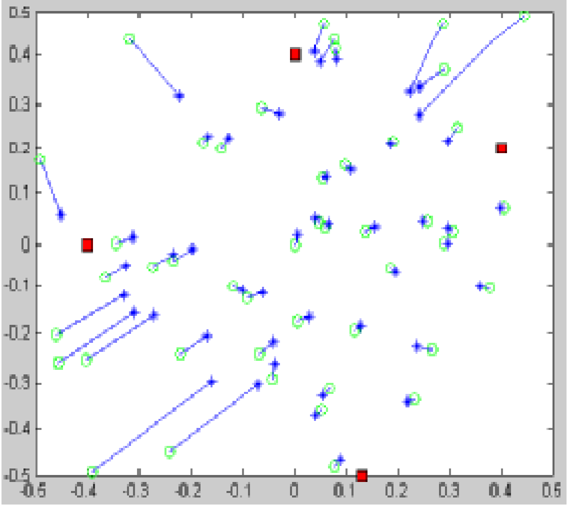


Figure 5. Magnified-Range Simulation (RMSD = 0.098)
Condition: Anchor Number = 4; Senor Number = 50; Noise Factor = 0.05;
Anchor Signal Range = 0.3 and 0.5

VI. Future Work

The simulations are currently based on 2-D environment to test the performance of our proposed schemes. Real applications may require 3-D environments. Therefore, our next research step is to focus on 3-D simulations. Moreover, the following challenges will be further investigated.

First, how to get more accurate distances between sensor nodes? Due to the limitations of acoustic signals and unique characteristics of underwater environments, it is very hard to compute the accurate distance. However, if it is possible to provide more accurate distance using other measurement schemes, it guarantees a better performance for the SDP approach. Moreover, this will allow the use of other localization schemes, especially some terrestrial range-based localization scheme.

Second, energy-efficient routing protocols are needed in such settings. Underwater routing protocols are relatively a new field. The unique features of acoustic propagation and noise make most of terrestrial WSN routing protocols not suitable for UWSN. Therefore, we plan to investigate the design of energy-efficient routing protocols that fit the marine environment especially in discovering the neighboring nodes within signal range.

Third is combining other localization approaches with SDP. There are many localization schemes that may lead to a better performance if combined with SDP. Therefore, we plan to research existing approaches to find a possible way to combine with our SDP-based schemes. This will probably lead to a hybrid localization scheme. The new system should generate better results and should address more marine applications and research problems.

Fourth, real-life experiments will be conducted. We plan to use commercial underwater transmitters to test our architecture design and schemes. Real-life demonstrations are most likely to reveal insights on improving the design and performance of the system. In fact, we are gathering more information on water current, attacks from sea creatures and other natural factors. This should help design a more robust underwater localization system.

VII. Conclusions

Compared to other localization algorithms, the SemiDefinite Programming (SDP) approach is proved to deliver accurate results even in noisy backgrounds. In this paper, we propose a design for underwater localization systems using SemiDefinite Programming. The objective is to provide marine scientists with a cost-effective and efficient way for conducting relevant research. Two designs, Fixed-Position and Magnified-Range, are proposed. Both schemes have shown better accuracy and performance, according to simulation results, over conventional networks. Future research directions are discussed in this paper.

Acknowledgment

This work was supported, in part, by an NSF CRI grant and a Texas Research Development Fund grant.

References

[1] B. Dong, and A.M. Mahdy, “GPS-Free Localization Schemes for Underwater Wireless Sensor Networks,” Proceedings of the Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, 2009, pp. 244-248.

[2] Y. Shang, W. Ruml, Y. Zhang, and M.P.J. Fromherz, “Localization from Mere Connectivity,” Proc. of ACM MobiHoc, Annapolis, MD, 2003, pp. 201- 212.

[3] D. Niculescu, and B. Nath, “Ad Hoc Positioning System (APS) Using AOA,” Proceedings of IEEE INFOCOM, San Francisco, CA, USA, 2003.

[4] A. Savvides, C.C. Han, and M. Srivastava, “Dynamic fine-grained localization in ad hoc networks of sensors,” Proceedings of MobiCom, 2001.

[5] D.B. Kifoyle, and A.B. Baggeroer, “The State of the Art in Underwater Acoustic Telemetry,” IEEE Journal on Oceanic Engineering, OE-25, No. 5, 2000, pp. 4-27.

[6] P.C. Etter, “Underwater Acoustic Modeling, Principles, Techniques and Application,.” 2nd edition, E&FN Spon, 1996.

[7] L. Berkhovskikh, and Y. Lysanov, “Fundamentals of Ocean Acoustics,” Springer, 1982.

[8] J. Proakis, J. Rice, E. Sozer, and M. Stojanovic, “Shallow water acoustic networks,” Encyclopedia of Telecommunications, John Wiley and Sons, 2003.

[9] V.O. Knudsen, R.S. Alford, J.W. Emling, “Digital Communications,” Journal of Marine Research, (7-12):410, 1948.

[10] J.H., Cui, J. Kong, M. Gerla, and S. Zhou, “The Challenges of Building Scalable Mobile Underwater Wireless Sensor Networks for Aquatic Applications,” IEEE Network, (0890-8044), 2006, pp. 12-17.

[11] I.F. Akyildiz, D. Pompili, and T. Melodia, “Underwater Acoustic Sensor Networks: Research Challenges,” Elsevier Ad Hoc Networks, Vol. 3, No. 3, 2005, pp. 257-279.

[12] I. Vasilescu, K. Kotay, and D. Rus, “Krill: An Exploration in Underwater Sensor Networks,” Proceedings of the Second IEEE Workshop on Embedded Networked Sensors, 2005, pp. 151-152.

[13] P. Biswas, and Y.Y. Ye, “Semidefinite programming for ad hoc wireless sensor network localization,” Proceedings of the third international symposium on Information Processing in Sensor Networks, 2004, pp. 46-54.

[14] B. Dong, and A.M. Mahdy, “A Survey of Underwater Wireless Sensor Networks,” Journal of Computing Sciences in Colleges, Electronic Issue, 2008.

[15] B.H. Wellenhoff, H. Lichtenegger, and J. Collins, “Global Positions System: Theory and Practice,” Fourth Edition. Springer Verlag, 1997.

[16] P. Bahl, and V. Padmanabhan, “RADAR: An In-Building RF-based User Location and Tracking System”, Proceedings of IEEE INFOCOM, 2000, pp. 775-784.

[17] CrossBow, Available from www.xbow.com (last visited June 2010).

[18] R. Stoleru, T. He, and J.A. Stankovic, “Secure Localization and Time synchronization for Wireless Sensor and Ad Hoc Networks,” Springer, 2007.

[19] N. Bulusu, and D. Estrin, “GPS-less Low Cost Outdoor Localization for Very Small Devices” IEEE Personal Communications Magazine, 7(5), pp. 28-34.

[20] P. Biswas, C. Liang, C. Wang, Y.Y. Ye, “Semidefinite programming based algorithms for sensor network localization”, ACM New York, NY, USA, May 2006, pp.188- 220.

[21] Z. Wang, S. Zheng, Y.Y. Ye, and S. Boyd, “Further Relaxations of the Semidefinite Programming Approach to Sensor Network Localization,” SIAM Journal on Optimization, Vol 19, Issue 2, 2006, pp. 655-673.

[22] J.J. More, and Z. Wu, “Global continuation for distance geometry problems,” SIAM Journal on Optimization, Vol 7, Issue 3, 1997, pp. 814-836.

[23] A.M.-C. So and Y. Ye, “Theory of semidefinite programming relaxation for sensor network localization”, Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms, 2005, pp. 405-414.

Editorial: Advances on Networks and Services

Special Issue on the 1st International Conference on Advances in Peer-to-Peer Systems (AP2PS 2009)

Prof Nick Antonopoulos
School of Computing
University Of Derby
UK

Prof Antonio Liotta
Department of Electrical Engineering
Eindhoven University of Technology
The Netherlands

Dr Takahiro Hara
Department of Multimedia Engineering
Osaka University
Japan

Peer-to-Peer (P2P) systems have considerably evolved since their original conception, in the 90's. The idea of distributing files using the user's terminal as a relay has now been widely extended to embrace virtually any form of resource (e.g., computational and storage resources), data (e.g. files and real-time streams) and service (e.g., IP telephony, IP TV, collaboration).

More complex systems, however, require more sophisticated management solutions, and in this context P2P can become an interesting issue, playing the hole of both the target and the enabler of new management systems. Therefore more intensive research is required to address the management of P2P applications as well as the use of P2P technologies as management tools for traditional and modern systems.

The First International Conference on Advances in Peer-to-Peer Systems (AP2PS 2009) that was held at Sliema, Malta in October 2009 built on the success of the First International Workshop on Computational P2P Networks (ComP2P 2008) but had a broader focus and aimed at capturing the latest developments, findings and proposals in the general area of P2P computing, networking, services, and applications. The conference was a resound success as it attracted over 100 paper submissions of which 34 were accepted after a double blind review process. This special issue includes the best five papers of the conference that have been suitably revised and extended to warrant a separate publication.

One issue that has received significant interest in P2P Network management is that of identifying and restricting free riding behaviour. Indeed free riding has been shown to significantly degrade the fault tolerance and content availability of a typical P2P network. In the paper, *A Taxonomy of Incentive Mechanisms in Peer-to-Peer Systems: Design Requirements and Classification*, Zhang et al., present seven design requirements for a free riding detection and prevention system based on incentives. The paper further classifies existing incentive mechanisms according to their ability to satisfy the aforementioned design criteria. The authors conclude, through a qualitative analysis, that reciprocity-based incentive systems represent the most promising solution to free riding.

The reliability of P2P systems depends on their ability to handle network churn or, in other words, unexpected peer failure and leaving. Due to their distributed nature, P2P networks tend to exhibit much higher churn rates compared to more traditional systems. In the paper, *Constructing a Stable Virtual Peer from Multiple Unstable Peers for Fault-tolerant P2P* Systems, Shikano et al present a novel method for constructing a stable virtual peer from a set of unstable physical peers. The authors explain how this mechanism improve the reliability of the P2P system and address various design challenges including how to replace failed peers and how to ensure communication between dynamic virtual peers. The paper concludes by providing an implementation of the proposed technique on an experimental P2P platform.

Another method for increasing the reliability of P2P systems, especially those that provide a distributed file storage service, is to use replication and random placement of the resulting replicas in order to overcome the unpredictable behaviour of peers. In the paper, *Replica Placement Algorithm based on Peer Availability for P2P Storage* Systems, Song et al present a new algorithm for determining replica placement based the availability pattern of every peer. Through a set of experiments, the authors show that their method substantially outperforms random replica placement in terms of data availability with only a moderate overhead in terms of memory consumption and processing time. The paper also demonstrates the use of this algorithm as an analysis tool for P2P networks in general.

Video streaming is emerging as a major application of the P2P paradigm. The issue of reliability of the underlying P2P network presents a substantial challenge in ensuring that video streaming users experience an appropriate level of quality of service. In the paper, *Resilient P2P Streaming*, Alhaisoni et al discuss stream redundancy, multi-source streaming and locality-awareness as methods for increasing the resilience of P2P streaming. The authors propose a new hybrid technique based on the above methods and prove through extensive simulations that their approach improves network utilisation, delay and throughput.

Wireless infrastructures have experienced a high increase in energy requirements in order to support more demanding end user services such as video streaming and multimedia applications. Mobile peer networking is a suitable platform for such service provision with the added benefit of requiring less transmission power compared to more traditional approaches by using appropriate peer connectivity heuristics. In the paper, *Increasing Energy Efficiency in Mobile Peer Networks by Exploiting Traffic Sampling Techniques*, Buhagiar and Debono present a novel algorithm that uses traffic mapping and neural networks to achieve a more efficient power distribution. The authors provide an optimisation of the base algorithm and through experimentation show that their approach presents a significant power efficiency improvement compared to traditional P2P networks.

In closing, we would like to thank the authors of the five selected papers for submitting their work to this special issue, the reviewers for their insightful and constructive comments and the IARIA administration team for their support in producing this publication. We hope you will find the material in this special issue stimulating and useful.

Guest Editors Biographies

Prof. Nick Antonopoulos is currently the Head of School of Computing and Assistant Dean (Research) of the Faculty of Business, Computing & Law at the University of Derby. Prior to joining the University of Derby in March 2009 he was a Senior Lecturer (US Associate Professor) and Director of the MSc Degrees at the Department of Computing, University of Surrey, UK. He holds a BSc in Physics (1st class) from the University of Athens in 1993, an MSc in Information Technology from Aston University in 1994 and a PhD in Computer Science from the University of Surrey in 2000. Prior to joining the academia he has worked as a networks consultant and were the co-founder and director of a company developing Web-based management information systems. He has over 11 years of academic experience during which he has designed and has been managing advanced Masters Programmes in computer science at the University of Surrey. He has published over 80 articles in fully refereed journals and international conferences. He has received a number of best paper awards in conferences and graduated 6 PhD students. He is a Fellow of the UK Higher Education Academy and a Fellow of the British Computer Society. His research interests include emerging technologies such as large scale distributed systems and peer-to-peer networks, software agent architectures and security. Contact him at the School of Computing, University of Derby, Derby, DE22 1GB, United Kingdom, N.Antonopoulos@derby.ac.uk

Prof. Antonio Liotta holds the Chair of Communication Network Protocols at the Eindhoven University of Technology (The Netherlands) where he leads the Autonomic Networks team. He was previously a Reader at the University of Essex (UK), where he led the Pervasive Networks team. Antonio is a Fellow of the U.K. Higher Education Academy and serves the Peer Review College of the U.K. Engineering and Physical Sciences Research Council. For many years now he has been a member of the Advisory Board of Editors of the Journal of Network and System Management (Springer) and of the International Journal of Network Management (Wiley). He is an active member of the network and service management research community. He has co-organized and co-chaired several international conferences; has served the Technical Programme Committee of over 100 international conferences; and has also contributed as keynote and tutorial speaker. Antonio has over 100 publications to his credit in the areas of autonomic network management, telecommunication services and peer-to-peer networks. After co-editing 4 books, he has authored "Networks for Pervasive Applications" (Springer), to appear in 2011.

Dr. Takahiro Hara received the B.E, M.E, and Dr.E. degrees in Information Systems Engineering from Osaka University, Osaka, Japan, in 1995, 1997, and 2000, respectively. Currently, he is an Associate Professor of the Department of Multimedia Engineering, Osaka University. He has published more than 150 international Journal and conference papers in the areas of databases, mobile computing, peer-to-peer systems, WWW, and wireless networking. He served as a Program Chair of several international conferences such as International Conference on Mobile Data Management (MDM'06 and 10) and IEEE International Conference on Advanced Information Networking and Applications (AINA'09). He guest edited IEEE Journal on Selected Areas in Communications, Sp. Issues on Peer-to-Peer Communications and Applications. He served and is serving as PC member of more than 100 international conferences such as IEEE ICNP, WWW, DASFAA, ACM MobiHoc, and ACM SAC. His research interests include distributed databases, peer-to-peer systems, mobile networks, and mobile computing systems. He is IEEE Senior member and a member of four other learned societies including ACM.

A Taxonomy of Incentive Mechanisms in Peer-to-Peer Systems: Design Requirements and Classification

Kan Zhang Nick Antonopoulos Zaigham Mahmood
School of Computing
University of Derby
Derby, UK
{k.zhang, n.antonopoulos, z.mahmood}@derby.ac.uk

Abstract— **Free riders in the Peer-to-Peer systems are the nodes that only consume services but provide little or nothing in return. They seriously degrade the fault-tolerance, scalability and content availability of the Peer-to-Peer systems. The solution to this problem in Peer-to-Peer networks is to have incentive mechanisms that aim to improve the network utility by influencing the nodes to be more cooperative. This paper presents seven design requirements according to the characteristics of Peer-to-Peer systems, latest distributed computing development trends and implementation techniques. This paper also provides a classification of the existing incentive mechanisms for Peer-to-Peer systems. For each category, the paper outlines the principles, provides typical examples, applications and discusses limitations against proposed design requirements. Two approaches to evaluate the effectiveness of the incentive mechanism are also presented. The findings suggest that the reciprocity-based incentive mechanisms are the most promising solutions. It is suggested that future research direction could focus more on the internal factors that encourage cooperation.**

Keywords-Peer-to-Peer; Free riding; Incentive Mechanism; GameTheory; Simulator; Instrinsic motivation

I. INTRODUCTION

Peer-to-Peer (P2P) techniques have been widely applied in file sharing [28], media streaming [46], VOIP [45], search engines [48] and so forth. P2P architecture overcomes the scalability issues and the fault-tolerance problems of a centralized Client/Server architecture [30]. This is due to the fact that all the functions of traditional powerful servers are distributed to the nodes throughout the network. An abstract P2P transaction can be described as a three-stage process: 1) a node (Requestor) issues a query for a resource and propagates the query through the network, 2) the node that provides this resource (Provider) receives this query and notifies the requestor that the requested resource can be found there and 3) the requestor establishes a connection to the provider and consumes the resources. However, it should be noted that whether a query can be successfully answered, largely depends on whether the provider will voluntarily provide the resources.

Researchers have conducted extensive measurements in real P2P applications. They have observed serious free riding problems in the sense that, whereas a proportion of nodes only consume resources from the system, they do not contribute to others. In 2000, Adar et al. [4] revealed that about 66% of the nodes in a Gnutella network [17] were free riders and 63% of the nodes were sharing un-requested resources. Moreover, nearly 50% of the queries were served by about 1% of altruistic nodes. Saroiu et al. [45] measured Gnutella networks again in 2002 and found that 25% of the nodes shared nothing and 50% of nodes shared very little. Furthermore, about 7% of nodes provided more than 90% of the total resources. In 2005, a measurement study [22] stated that the percentage of free riders in Gnutella networks went up to more than 85%. Another research conducted in 2005 [51] indicated that Maze P2P network also had free riding problem as more than 35% of the nodes did not share any resources. A study [20] in 2006 observed that more than 70% of the nodes were free riders in eDonkey P2P networks.

Such serious free riding problems can significantly decrease the total network utility as the majority of the network requests are served by a small portion of altruistic nodes, thus weakening the fault-tolerance of P2P networks [26]. Moreover, queries may be rejected by these ‘hot spots’ because of their capability bottleneck, leading to the scalability problem. In addition, the content availability of the network becomes limited, as a majority of nodes in the networks do not contribution any resources or contribute only a little.

Many researches and measurements [15][36] indicate that the rationality and the absence of incentive mechanism are the two factors that lead to the free riding problem in P2P systems. The majority of the nodes in P2P systems are self-interested. Their ultimate goal is to maximize their own utilities. Without compensation for contribution, the rational nodes tend to free ride since the profit gained from sharing can not cover the cost of such altruistic behaviors.

Extensive Incentive Mechanisms [5][13][25][50] have been proposed in the past decade to prevent free riding in P2P systems. Nonetheless, they fail to significantly improve the system utility due to their impracticability or logistical shortcomings. The contribution of this paper is twofold: First, we propose the incentive mechanism design requirements. Second, we provide guidance for future research directions by classifying the existing incentive mechanisms and analyzing them against the proposed design requirements.

In the next Section, we present seven incentive design requirements. Section III presents a classification and detailed analysis of existing incentive mechanisms. Two main effectiveness evaluation approaches are presented in Section IV. Following that, the Section V concludes the paper by summarizing the remaining work for each category and the open questions related to the free riding problem.

II. Incentive Mechanism Design Requirements

One solution to resolve the free riding problem in P2P systems is to apply an incentive mechanism that influences nodes' behaviors in a certain manner to increase the utility of the whole system. The incentive mechanisms need to address the challenges that arise from the P2P characteristics, latest development trends of distributed computing and implementation techniques. Moreover, during the mechanism design phrase, one should also consider the effectiveness of the incentive mechanism and its psychological influences. We propose the following design requirements:

Requirement 1: Decentralization

As most of the P2P systems are decentralized, the incentive mechanisms also need to be self-managed, that is, no dedicated centralized entity should be involved in monitoring nodes' behaviors, assessing their contributions, storing data and so forth. In this way, the scalability and fault-tolerance properties of P2P systems are preserved.

Requirement 2: Adaptability

P2P networks can be classified into two categories: unstructured and structured. In the unstructured P2P networks, resources are distributed randomly throughout the network whereas the structured P2P networks place the resources to specific locations. Regardless of the architecture of the networks, the resource discovery is the most crucial issue in P2P networks. The deployment of the incentive mechanism should not affect the underlying P2P networks' bootstrap and search mechanisms. Moreover, there should not be any big practical and psychological burdens on participation after the deployment.

Requirement 3: Service Diversity

Recently, Cloud computing and Service-oriented computing has drawn increasing attention by both industry and academia. These emerging techniques allow the heterogeneous users to collaborate with each other to perform much more complex tasks than classic P2P applications such as file sharing and video streaming. P2P overlay networks have been widely applied for resource discovery in such new technologies [12]. To cope with the high demand of richer interaction and collaboration between system users, the incentive mechanisms should be able to function effectively in the environment with high service diversity.

Requirement 4: Reward

The most important principle of an incentive mechanism is to reward the nodes' contributions. To evaluate a node's contribution, one can collect information from many sources such as personal experience, trusted third parties, collective global history and so forth [32]. However, the aggregation of the collected information should be carefully considered as the trust relationship in distributed systems is not guaranteed and the update of such information can be very frequent. Moreover, the mechanism designer should also take the heterogeneity of the P2P systems into account since the nodes can have various capability and the services are from different contexts.

Requirement 5: Penalty

[36] classifies the uncooperative behaviors into three categories: 1) selfish behavior: a node deliberately fails to fulfill a task in order to increase its own utility; 2) malicious behavior: a node tries to harm either a specific group of members or the whole system - the malicious nodes do not gain profit, thus, the detection and prevention of malicious behavior is out of the scope of this paper; 3) venial behavior: a node fails to commit a task due to reasonable reasons such as storage shortage and connectivity problems.

In the P2P systems, there are four types of nodes: 1) altruistic nodes that are always cooperative; 2) rational nodes which try to increase their utility by choosing either behave selfishly or cooperatively; 3) pure free riders that simply do not provide resources in any case; 4) intelligent free rider that may free ride selectively according to the applied incentive mechanism.

The incentive mechanism should be able to detect and punish the selfish behaviors. However, the global recognition of free riders is not necessary. The venial behaviors should be distinguished with the selfish behaviors. The free riders should not be able to obtain complete requested services and either choose to leave the systems or be isolated by the cooperative nodes. The rational nodes should have opportunity to change their behaviors from selfish to cooperative and start gaining profits before actively getting disconnected or barred from the system or being isolated.

Requirement 6: Enforcement

The enforcement of the mechanism is highly crucial for ensuring the effectiveness of the incentive mechanism. One solution is to encode the rules in the client software as BitTorrent [13]. However, this is fragile as one can possibly tamper the official software or develop its own [29].

Most of the P2P networks use user-specified pseudonyms. This is not resilient to the whitewash and sybil attacks. The issue can be resolved by introducing strong system identities for the accountability. However, the identification of the nodes in a distributed un-trusted P2P environment is very challenging since the certificate authority (CA) is required which is usually centralized. Although, decentralized public key infrastructures have been proposed, they either lack the reliability or increase the complexity of the system [2]. Therefore, the incentive mechanism should not rely on strong system identities for accountability.

Requirement 7: Lightweight

As a result of the Requirements 1, 2 and 6, the implementation of the incentive mechanism should be lightweight. That is, the deployment of the incentive mechanism should not produce heavy overhead in terms of amount and size of the messages. The cost of extra processing carried out by the nodes should be kept lower than the utilities they can gain from the increased social welfare.

III. Incentive Mechanism Classification

In this section, we evaluate the existing incentive mechanisms for P2P systems against design requirements mentioned above. We classify the existing incentive mechanisms into three categories: Monetary-payment, Fixed contribution and Reciprocity-based schemes. For each category, we provide the principles with example models to analyze the advantages as well as limitations.

A. Monetary-Payment Scheme

Monetary-payment Scheme [18][40][50] follows the principle that nodes pay the resource providers for the resources they consume with either real money or virtual tokens [18]. Micropayment systems are the common implementation of this scheme. There are two generations of such systems: Token-based and Account-based.

In Token-based systems, a customer first checks the service at the merchant. Second, it can buy tokens from the broker. Third, it can use the tokens to purchase the service at the merchant. The merchant then provides the service to the customer and finally redeems the tokens from the broker. This process is illustrated in Figure 1. One example is the extensively cited token-based mechanism, called PPay [50], proposed by Yang in 2003, which introduces an internal transferable and self-managed currency called coins. In this case, the process is as follows. A node X purchases a digital raw coin from a broker, say, C = {X, sn} SKS, where sn is the serial number of the coin and SKS is the secret key of the server. The node X, as the owner of this coin C uses C to pay the services from another node, say, Y by sending it the assigned coin: CXY = {Y, seq1, C} SKX, where seq1 is a sequence number indicating the time of the assignment and SKX is the secret key of node X. The node Y now becomes the holder of the coin C and, therefore, cashes this assigned coin from the broker or uses it to pay for services from other nodes. In case the node Y wants to purchase services from node Z, node Y first sends a reassignment request to the coin's owner node X: RXYZ= {Z, CXY} SKY where SKY is the secret key of node Y. Node X then verifies the CXY and sends the new assigned coin: CXZ = {Z, seq2, C} SKX to nodes Y and Z. It is noticeable that the seq2 should be bigger than seq1 to indicate that this is a new assignment.

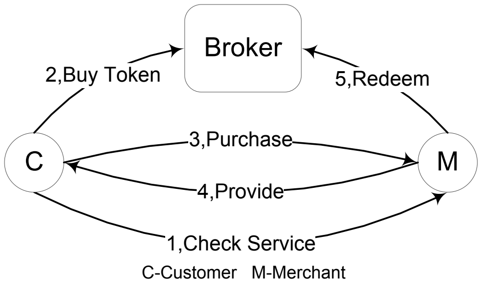


Figure 1. Abstract Transaction in Token-based Micropayment Systems.

The main limitation of the token-based micropayment system is that every transaction will generate a new token. The broker has to keep a record of all the tokens, which, in turn, leads to a scalability problem.

The second generation micropayment systems are account-based in which every customer has an account with the broker and authorizes the broker to transfer money from their accounts. To purchase a service, a customer first checks the service at the merchant. Second, it informs the broker of its interests. The broker then notifies the merchant that its services are of interest to a customer. The merchant then provides the service to the customer. After checking the quality of the service, the customer confirms with the broker that he is willing to pay. The broker then takes the money from the customer's account and pays the merchant. This process is illustrated in Figure 2.

There are several successful account-based real applications available for P2P networks, such as PayPal [40]. These are more scalable as compared to the token-based systems since the broker only manages nodes' accounts rather than all the tokens.

The monetary-payment scheme can work effectively in a service-oriented environment as the service providers are rewarded with either money or virtual currency and the free riding behavior can not gain nodes any profits. The deployment of such a scheme will not affect the resource discovery mechanism of the underlying P2P networks. However, to implement both the token-based and the account-based systems, dedicated centralized entities are required for token or account management. Although, it is fair that the price setting is up to the resource providers, some real world economic problems such as inflation and deflation need to be carefully considered [23]. The strong system identities are necessary for accountability so that the implementation is not trivial [15]. In addition, it could discourage the participation because of the heavy cost of identity creation.

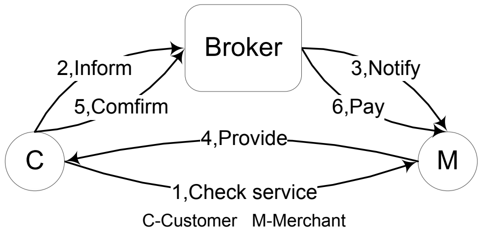


Figure 2. Abstract Transaction in Account-based Micropayment Systems.

B. Fixed-Contribution Scheme

Fixed-contribution scheme employs a simple rule: to participate in the networks, a node has to contribute a fixed amount of resources.

Direct Connect [14] is a typical example using fixed contribution incentive mechanism. It requires each node to contribute a minimum amount of files and make a minimum upload bandwidth available. This scheme normally requires a centralized entity to monitor the quality of contributions made by the nodes [8]; however, it is not suitable for a majority of decentralized P2P networks. Moreover, the centralized entity can only evaluate nodes' contributions in terms of validity and quantity. Consequently, this scheme is only suitable for single-service applications like file-sharing. The incentive for nodes to contribute is the access rights to network resources. However, once the nodes join the network, they do not have incentives to contribute any more. Therefore, the setting of the contribution threshold should keep the balance between the cost of contribution, and network resource diversity and availability. To adapt to the supply and demand trends of P2P network, a dynamic entrance threshold may be applied, causing unfairness, as some nodes may contribute less than others but have the same level of access rights to the network resources. The scheme needs to be deployed to the network from the bootstrapping phase rather than established networks. Since the priority gained from the contribution is the same for everyone, there is no incentive for nodes to spoof or collude. The user-specified pseudonyms can fulfill the requirement for accountability and thus the scheme is relatively lightweight.

C. Reciprocity-Based Scheme

In reciprocity-based schemes [3][5][13][25][27][43], a provider allocates its resources to requestors in proportion of their contributions. Based on the way a node's contributions are calculated, the mechanisms in this scheme can be categorized into two approaches [15]: non-real-time approach and real-time approach. In non-real-time approach, the nodes assemble information about the other nodes from multiple sources such as local information, trusted third party, neighbors and so forth [32] while in real-time approach, the transaction partners evaluate each other only during the transaction.

1) Non-Real-Time Reciprocity-based Approach

The non-real-time reciprocity-based incentive mechanisms can also be referred to as reputation-based systems. These aim to calculate a reputation value for every node to indicate their performance in the past. Nodes can use these values to predict others' behaviors in transactions.

In some systems, the nodes only use private information, which is resulting from direct transactions, to compute the reputation value for others. For examples, the EigenTrust [25] calculate a global trust value for all the nodes in the network. The algorithm is based on transitive trust, that is, if a node trusts its friends, it will also trust its friends' friends. All the nodes maintain a normalized local trust values in a vector c. A normalized local trust value matrix C combines all the local trust value vectors. Trust value vector $t = (C^T)c_i$ contains the results that node i asks its friends and weights them based on its own personal experience with them. To get a broader view of the network, the nodes can ask their friends' friends and continue in this way so that their trust value vector can be denote as $t = (C^T)^n c_i$. If n is large enough, the trust value vector for every node will converge to the same, which is the left principal eigenvector of C. To obtain C in a distributed manner, the algorithm employs a Distributed Hash Table (DHT) to assign every node a set of trust score managers. The trust score managers are responsible for submitting their children nodes' local ratings to ratees' trust score managers, aggregate ratings from raters' trust score managers and computing the global trust values. However, since this algorithm uses DHT techniques, it is not suitable for networks with high network churn due to their heavy maintenance overheads. The level of data redundancy is high since the entire set of nodes act as trust value managers have to compute the global trust value vector.

PowerTrust [43] is an improved version of EigenTrust. It proposes a Look-ahead Random Walk algorithm in which all the nodes calculate the trust values based on their neighbors and their own ratings $t = (C^T)c_i$. Therefore, the enhanced trust value matrix can be obtained and denoted as $E = C^2$. Using E instead of C to compute the global trust value vector significantly improves the convergence rate of the calculation. The paper also identifies that in P2P reputation systems, nodes feedback distribution is power-law. That is, the majority of nodes receive very few ratings while a small number of nodes receive a large number of ratings. The paper presumes that the trust values distribution has the similar distribution. Similar to Eigentrust, the DHT with locality preserving hashing mechanism can be applied to assign every node a trust value manager, which collects all the ratings about this node and submit these ratings to the ratees' trust value manager. Instead of asking all the trust managers in the network to calculate the global trust values, PowerTrust propose a distributed ranking mechanism to find a number of most trustworthy nodes to perform the calculation and store the trust values. The Hash values of nodes' trust values are inserted to their successor in the DHT. The nodes with fewest children nodes maintain the most trustworthy nodes' trust values because of the locality preserving hashing and the power-law trust score distribution. PowerTrust significantly reduces the calculation complexity and the data redundancy problem of Eigentrust.

PeerTrust [27] propose another trust metric to compute trust values of nodes by considering the following five factors. First, the ratings a node receives reflect its performance in past transactions. Second, the total number of transactions a node participates in, can be used to normalize the ratings it receives. Third, the credibility of the raters should be carefully considered. To weight the raters, a node computes the rating similarity between the rater and itself. A node is more likely to trust another node if they have similar opinions on a same set of nodes. Four, context of each transaction could be different. Therefore, the model also weights the ratings with the transaction factors such as the size of the transactions. Lastly, community context can be used as compensation to the aggregated trust values. For example, nodes that provide their ratings to others can receive a reward. This model also assigns every node a trust value manager that is responsible for rating submissions and trust value calculation. Any DHT techniques can be used for trust manager assignment. However, it has the same maintenance problem as all the structured P2P networks. Moreover, to retrieve these five factors may introduce heavy overheads.

The incentive in most reputation-based systems is that the requestors always choose the providers with highest trust values. Some other reputation systems use the trust values to present nodes' contributions to the system in the past. The providers can differentiate or schedule their services or resources according to the requestors' contributions. The incentives in these systems may include access rights [11], and differentiated quality of service [19].

Ma et al. [31] propose five resource (bandwidth) distribution mechanisms: 1) Even Sharing Mechanism (ESM), where the provider evenly divides its resource to all the requestors; 2) Resource Bidding Mechanism (RBM), where the requestors send a bidding message to the provider, indicating the upper limit of its download bandwidth. The provider then will not assign the bandwidth beyond their upper limits. This mechanism solves the bandwidth waste problem of the ESM; 3) Resource Bidding Mechanism with Incentive (RBM-I), where based on RBM, the requestors also include their contributions in the bidding messages. The provider can then assign its resources in proportion to the requestors' contributions; 4) Resource Bidding Mechanism with Utility Feature (RBM-U), where the provider differentiates its resources according to the requestors' bandwidth limits, providing them similar level of satisfaction and 5) Resource Bidding Mechanism with Incentive and Utility Feature (RBM-IU) where the provider assigns its resources based on both requestors' contributions and upper bandwidth limits. The authors suggest that the RBM-IU can achieve Pareto-optimal resource allocation. That is, the outcome of the allocation can not be improved without reducing at least one requestor's utility.

All the rating aggregation, trust value calculation and trust value manager assignment in the reputation systems are carried out in a distributed manner. The reputation systems are not sensitive to the changes in the nodes' behavior. That is, a node can accumulate high trust value and start free riding without being detected. Most of the reputation systems can be only applied for single-service applications. For example, a node may perform very well for sharing files. But it does not guarantee that it will also perform well for sharing computational resources.

Wang et al. [48] propose a Bayesian network trust model, where the trust values are calculated in a context-aware manner. However, the system complexity is significantly increased by the comparison and updating of Bayesian networks and the calculation of the trust values [24]. Moreover, to obtain a more convincible reputation values, the nodes are required to collect more information from strangers rather than its personal experience. Therefore, the strong system identities are required to account the credibility of the information source. Storage of these reputation values is another issue that needs to be carefully considered. Currently, most of the systems use DHT techniques to assign trust value managers, which generate extra maintenance overhead.

2) Real-Time Reciprocity-Based Approach

In real time reciprocity-based systems, transaction partners evaluate each other only during the transactions. Nodes are forced to make resources available when they are consuming resources from others. Such systems can also be referred to as exchange-based incentive systems.

BitTorrent [13] is an example of exchange based incentive systems. All the resources are divided into small segments with equal size. BitTorrent organizes nodes with the same interest into a group and enables them to download and upload resource segments among themselves.

The nodes with complete resources can create files with the .torrent extension and publish them. A .torrent file contains information about file length, name, hashing information and URL of a tracker. A tracker is a server that is responsible for helping the nodes to find other nodes with the same interest. To consume a resource, a node downloads the .torrent file and contacts the corresponding tracker to obtain a set of IP addresses of nodes that are currently downloading the resource (downloader). It can, then, establish connections to these downloaders and download file pieces from them. Once it completes the downloading of some file pieces, others can also start downloading these pieces from it. Figure 3 illustrates a transaction example in BitTorrent networks.

In the original BitTorrent, a centralized dedicated tracker is needed to organize nodes with the same interests. In some of its variations [16], the role of the tracker is assigned to the existing nodes, which is achieved by applying the DHT techniques. Both the original and its variations require that the resources shared with the network should be dividable. Therefore, they cannot be applied in applications with high service diversity. The incentive for nodes to make more contribution is the bandwidth. It can effectively degrade the QoS of the free riders since they are choked in most scenarios.

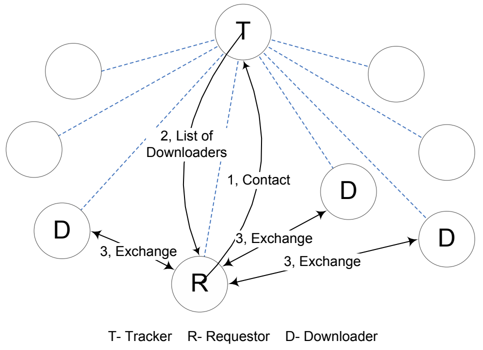
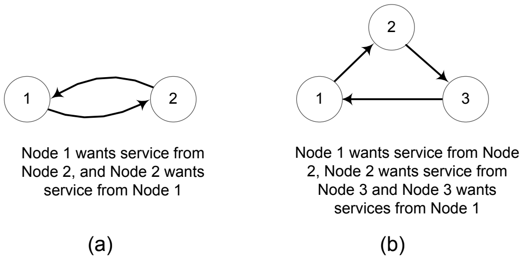


Figure 3. Transaction Example in BitTorrent Network.



Node 1 wants service from Node 2, and Node 2 wants service from Node 1

(a)

Node 1 wants service from Node 2, Node 2 wants service from Node 3 and Node 3 wants services from Node 1

(b)

Figure 4. Service Exchange Rings

In systems with higher service diversity, the standard mechanism involves direct exchange between two nodes with mutual interests [8]. Furthermore, some researchers [3], [5] have proposed a new concept of *service exchange ring*, which is illustrated in Figure 4. Each node in a ring provides a service to its successor and consumes a service from its predecessor. A node can participate in multiple rings. In a ring, the nodes always monitoring their predecessors and if free riding is detected, they can take proper actions to identify the free riders and isolate them.

Anagnostakis et al. [5] propose a mechanism to establish such rings. Every node has an *Incoming Request Queue* (IRQ) that maintains all the nodes, which request a local resource. Every node also has a *request tree*, which has the following structure: the root of the tree is the node; the first generation child nodes are the nodes in its IRQ; the second generation child nodes are the nodes registered in the first generation child nodes' IRQs. The process stops at the nth generation where n is limited to 5. A node keeps checking if any nodes in its request tree can satisfy its request. If so, it can form an n-way exchange ring. For example, as shown in Figure 5, the Node 0's IRQ consists of node 2 and node 4. Node 2' IRQ includes node 3 and node 1. Similarly, node 4' IRQ includes node 5, node 9 and node 6. If node 0 finds that one of its requests can be satisfied by node 5, then a service exchange ring that consists of node 0, 4 and 5 can be formed. The authors also propose, although very briefly, three possible free riding prevention mechanisms including: local blacklist, cooperative blacklist and synchronous block exchanging.

The current study is part of a bigger research project [52] that aims to build incentives for service oriented P2P networks. The service exchange ring establishment is achieved by modifying the query protocols of the underlying P2P protocols. The query messages can be used as the media where the nodes can publish their interests and provisions. When the query messages are propagated through the network, the nodes can self-form rings by checking the recorded services within the query messages. Therefore, no centralized entity needs to be involved. In general, the service-exchange-ring systems are designed for service-oriented applications. Node can specify various constraints on the resources or the operations on the resource within the service exchange rings to avoid deadlock [21]. For example, the initiator could restrict that all the resources shared in a certain service exchange ring should be dividable and the size should be more than 5MB. The incentive for nodes to provide promised services is the access right to the requested services.

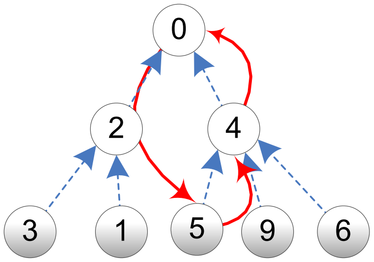


Figure 5. Service Exchange Rings

To prevent free riders from getting served, a simple rule can be applied: if a node is not receiving the promised service, it will stop providing service or decrease the quality of service to its successor. In this way, the service that the free rider consumes will also be influenced by the chain effect.

The authors also present a simple mechanism for free rider prediction and isolation in service exchange ring based incentive mechanism. Every node holds a *Suspect Table*, which records all the members of all the collapsed rings it has participated in. An entry in a suspect table consists of three elements: 1) a node ID, 2) a set of probabilities that this node is a free rider in the same collapsed rings that the table holder node participated in and 3) a Prediction value that is the average probability that the corresponding node is a free rider. The table holder also records the nodes with higher prediction values in a *Blacklist*. The nodes try to avoid joining the rings where there may be nodes sin a Blacklist.

To keep the suspect table lightweight, we employ a *First in First out* (FIFO) queue data structure to store the set of probabilities in table entries. If a queue is full and a new probability is about to be inserted, the oldest probability is deleted and the new probability inserted. In this way, a node's most recent activities are always recorded.

To calculate the prediction value, let ***n*** denote the size of the queue, ***p_i*** the probability where ***i*** indicates the position in the queue and **s** denote the size of the collapsed ring. The probability ***p*** being inversely proportional to the size of collapsed ring, each ring member of a collapsed ring is assigned an equal probability of ***1/s***. The prediction value for each node is now calculated according to following expression:

$$\text{Prediction value} = 1 - \prod_{0}^{n} (1 - P_i)$$

A node would get higher prediction value than others if it participated in more collapsed rings or if it appeared in smaller ones. It is important that opportunities are given to the rational nodes to change strategy and to the nodes to recover from connectivity problems. The nodes should periodically delete the oldest probability from each entry from their Suspect Tables. In this way the Suspect Tables are always kept updated so that the old information gradually fades away.

Since the search mechanism of the original P2P protocols are not affected, the extra overhead to the messages and the processing for the local blacklists should be relatively small compared with other incentive mechanisms.

This 'memory-less' scheme [7] does not require dedicated centralized entities and strong system identities for ensuring accountability. It is very adaptable as the resource discovery mechanism of the underlying P2P networks can also be used to detect service exchange rings. Surprisingly, there are not many service exchange ring based incentive mechanisms for P2P networks, though it has great potential to fulfill all the design requirements.

IV. Evaluation Methods

Real P2P systems can be scaled to a large number of nodes, the evaluation of the incentive mechanism in the real systems is impractical. Two approaches that are normally used to evaluate the effectiveness of the incentive mechanisms in P2P networks are: Game theory and simulators.

A. *Game Theory*

Game theory has been widely applied in economics, biology, engineering and so forth. It models competitions as games and tries to mathematically work out the best strategies for the game players.

The P2P systems consist of multiple nodes, each of which may carry out infinite number of transactions. In any transaction, a node can choose from finite number of actions. For simplicity, we assume that only two actions are available: serve requests and free ride. Most of the existing game theoretical analyses assume that all the nodes are rational, that is, they always try to maximize their utility and minimize the expense. Therefore, P2P systems can be represented as n-person repeated non-cooperative games that allow mixed strategies. According to Nash Folk Theorem [37], in such strategic games, there always exists at least one Mixed Strategy Nash Equilibrium (MSNE). The Nash Equilibrium is a strategy profile that consists of a strategy from every player who cannot increase its utility by choosing another strategy, given the strategies from the others.

Many researches [9] [10] have been conducted to find the MSNE, first, one needs to model the utility function of the players, which are assumed to have the same role, given the applied incentive mechanism. In general, a player utility is calculated as the profits it gains minus the cost of its contributions. Then, one can calculate the mixed strategy Nash equilibrium for every player (the calculation can be found in [38]).

Extensive game theoretical analyses have been conducted to prove that the non-real-time reciprocity-based incentive mechanisms can influence the nodes in the P2P systems to be more cooperative and hence increase the social welfare. However, these analytical models usually rely on some assumptions that simplify the systems，failing to estimate the real characteristics precisely. For example, although the rational nodes are in majority in the P2P systems, both the altruistic nodes and pure free riders can also be observed. In addition, the supernodes in the hybrid P2P systems take more responsibilities than the rest and thus their utility function are different to the others. The influence of such assumptions to stability and reliability of the Nash equilibrium needs to be further evaluated. Despite this, the models can still produce useful indications of nodes' behaviors.

B. *Simulator*

Another approach to compensate game theory for evaluating the effectiveness of the incentive mechanism is to simulate the system on a simulator. The functionality criteria that a P2P network simulator should meet, includes [34]: 1) allow the implementation of custom P2P protocols; 2) adapt to unstructured, structured and hybrid P2P architectures; 3) allow discrete event driven execution, that is, the events are scheduled by timers and appropriate handlers are called when the timers expire; 4) simulate underlying networks in either packet level or message level. Packet-level simulator simulates the data packets transfer through TCP connections while the message level ones only simulate the packets delay; 5) be scalable; 6) provide flexible measurement facilities and 7) provide user friendly interface and comprehensive API.

Naicken et al. [35] surveys extensive P2P researches and finds that most of them use either custom simulators or do not specify the details of the simulator that is used. Only a small portion of the papers surveyed refer to the use of open source simulators. Each of these can meet all the requirements stated above for P2P network simulation. For example, P2PSim [39] is a packet level discrete-event driven simulator only for structured P2P networks. Its scalability is proved to be low by real experiments. PeerSim [41] is another P2P simulator that supports discrete event scheduling. It can be used for both unstructured and structured P2P networks. However, it does not model the underlying network and hence loses accuracy in simulation results.

The lack of a standard for P2P simulators, results in much duplicated work and less convincible research findings. ProtoPeer [42] takes a step forward towards a P2P simulator standard. Its network model provides custom message design, packet delay and lost models and peer bandwidth assignment. This overlay model defines an abstract Peer Identifier, which allows the developers to implement custom peer IDs for both structured and unstructured P2P applications. The protopeer is discrete event driven and it supports concurrent execution of multiple handlers in different threads. Its measurement infrastructure allows data collection from all the peers and flexible operations on the collected data.

V. Conclusion and Futrue Research Directions

This paper has presented the negative effects that free riding problem brings to P2P systems, including reduced content availability, fault resilience and scalability. We have also proposed seven design requirements for the incentive mechanisms that can alleviate this problem. The existing incentive mechanisms are classified into three categories: monetary payment, fixed contribution and reciprocity based schemes. Taxonomy is shown in Figure 6. Their principles are explained along with the limitations against the proposed requirements, which are shown in Table I.

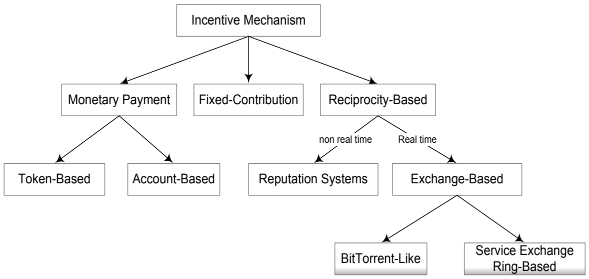


Figure 6. Taxonomy of Incetive Mehcanisms in P2P systems

The monetary payment based incentive mechanisms is applicable as long as the security related issues can be properly solved. A standard for micropayment systems is needed since the existing models are not inter-operational, limiting the system flexibility.

The fixed contribution based incentive mechanisms can not effectively prevent free riders in P2P systems. However, its principle can be incorporated in the real time reciprocity based scheme. That is when a node is consuming resources it is required to make a minimum amount of resources available to either the resource provider or the whole community [6]. Although, most reputation systems are impractical, this scheme has great potential to prevent free riding effectively. The complexity of the mechanisms can be reduced if fewer sources are contacted. The metrics for trust value calculation should be context-aware and behavior sensitive, that is, when misbehavior occurs, a node's trust value should drop quicker than if the behavior is good in which case the trust value should increase relatively slower.

BitTorrent like systems can only be applied to applications with resources that are dividable such as files, streaming and computational resources.

The service exchange ring based incentive mechanisms can meet most of the requirements. However, the feasibility of such mechanisms largely depends on the search mechanisms of the underlying P2P networks. For example, in pure P2P networks like Gnutella, which use blind search mechanisms, the gathering of the demand and supply information among nodes is very challenging. Moreover, although we believe that the overall request rate of nowadays service-oriented applications is very high, we should allow the altruistic nodes to help when a ring cannot be formed.

Existing incentive mechanisms focus more on the extrinsic motivations, more specifically, the external regulations. That is, one performs to satisfy the contingent external reward. Apart from external regulations, the extrinsic motivation regulations are of three other types [44]: 1) Introjected regulations, where one controls the behaviour according to contingent self-esteem; 2) Regulations through identification, where one controls the behaviour according to the extent she consciously value the outcome or importance of the given tasks and 3) Integrated regulations. The regulations are fully congruent to one's values and needs. Ryan [44] argues that as people internalize the regulations to their self, they may behave more effectively, voluntarily.

To facilitate the internalization, the relatedness meaning the feel of community is the most important factor. In addition, the feelings of competence and autonomy such as respect, choices and opportunities for self directions are also found to facilitate internalization. There are also the factors that facilitate the intrinsic motivations.

To incorporate these factors in the context of P2P systems, the future research could focus on the following questions: How to group nodes according to their interests, capability and provisions? How to enhance communication between nodes? How to design software interface? How to design tasks that have the appeal of challenge and novelty?

TABLE I. COMPARISON OF EXISTING INCENTIVE MECHANISMS FOR P2P SYSTEMS

			De	Ad		SD	Re		P		E	L
				To	SM		CE	Re	FRD	P		
MP			×	√	√	√	Local	Money Virtual Token	√	No Service	System Identities	×
FC			×	×	√	×	Third Party	Access Right	Partial	No Service	User-Specified pseudonyms	√
RB	NRT-RB		√	√	√	Partial	Global	Differentiated QoS	Partial	Differentiated QoS	System Identities	×
	RT-RB	BT	√	×	×	×	Local	Differentiated QoS	Partial	Differentiated QoS	User-Specified pseudonyms	√
		SER	√	√	√	√	Local	Access Right	TBS	No Service	User-Specified pseudonyms	TBS

MP: Monetary Payment; **FC**: Fixed Contribution; **RB**: Reciprocity Based; **NRT-RB**: Non Real Time Reciprocity Based; **RT-RB**: Real Time Reciprocity Based; **BT**: BitTorrent; **SER**: Service Exchange Ring; **De**: Decentralization; **Ad**: Adaptability; **To**: Topology; **SM**: Search Mechanism; **SD**: Service Diversity; **Re**: Rewards; **CE**: Contribution Evaluation; **P**: Penalty; **FRD**: Free Rider Detection; **E**: Enforcement; **L**: Lightweight; **TBS**: To Be Specified;

REFERENCES

[1] K. Zhang, N. Antonopoulos and Z. Mahmood, "A review of incentive mechanism in Peer-to-Peer systems," AP2PS 2009, First International Conference on Advances in P2P Systems 2009, pp.45-50.

[2] K. Aberer, A. Datta, and M. Hauswirth, "A decentralised public key infrastructure for customer-to-customer e-commerce," *International Journal of Business Process Integration and Management*, vol. 1, 2005, pp. 26 - 33.

[3] T. Ackemann, R. Gold, C. Mascolo, and W. Emmerich, "Incentives in peer-to-peer and grid networking," Research Note, 2002.

[4] E. Adar and B. Huberman, "Free riding on Gnutella," First Monday, vol. 5, 2000.

[5] K. Anagnostakis and M. Greenwald, "Exchange-based incentive mechanisms for peer-to-peer file sharing," Distributed Computing Systems, 2004. Proceedings. 24th International Conference on, 2004, pp. 524-533.

[6] P. Antoniadis and C. Courcoubetis, "Enforcing efficient resource provisioning in peer-to-peer file sharing systems," SIGOPS Oper. Syst. Rev., vol. 40, 2006, pp. 67-72.

[7] P. Antoniadis, C. Courcoubetis, and B. Strulo, "Incentives for content availability in memory-less peer-to-peer file sharing systems," *SIGecom Exch.*, vol. 5, 2005, pp. 11-20.

[8] P. Antoniadis and B. Le Grand, "Incentives for resource sharing in self-organized communities: From economics to social psychology," 2nd International Conference on Digital Information Management, 2007, pp. 756-761.

[9] T. Bedrax-weiss, C. Mcgann, and S. Ramakrishnan, "Formalizing resources for planning," In Proceedings of the ICAPS-03 Workshop on PDDL, 2003.

[10] T. Bocek, M. Shann, D. Hausheer, and B. Stiller, "Game theoretical analysis of incentives for large-scale, fully decentralized collaboration networks," Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on, 2008, pp. 1-8.

[11] C. Buragohain, D. Agrawal, and S. Suri, "A game theoretic framework for incentives in P2P systems," Peer-to-Peer Computing, 2003. (P2P 2003). Proceedings. Third International Conference on, 2003, pp. 48-56.

[12] R. Buyya, Chee Shin Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," *10th IEEE International Conference on High Performance Computing and Communications*, 2008, Dalian, pp. 5-13.

[13] B. Cohen, "Abstract Incentives Build Robustness in BitTorrent." Technical Report, 2003.

[14] DC++ website, http://dcplusplus.sourceforge.net/, Last Access: 06/25/2010.

[15] M. Feldman and J. Chuang, "Overcoming free-riding behavior in peer-to-peer systems," SIGecom Exch., vol. 5, 2005, pp. 41-50.

[16] C. Fry and M. Reiter, "Really truly trackerless bittorrent," Technical Report, 2006.

[17] Gnutella Protocol Specification v0.4, http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf, Last Access: 06/25/2010.

[18] P. Golle, K. Leyton-Brown, and I. Mironov, "Incentives for sharing in peer-to-peer networks," Proceedings of the 3rd ACM conference on Electronic Commerce, Tampa, Florida, USA: ACM, 2001, pp. 264-267.

[19] A. Habib and J. Chuang, "Service differentiated peer selection: an incentive mechanism for peer-to-peer media streaming," Multimedia, IEEE Transactions on, vol. 8, 2006, pp. 610-621.

[20] S.B. Handurukande, A. Kermarrec, F.L. Fessant, L. Massoulié, and S. Patarin, "Peer sharing behaviour in the eDonkey network, and implications for the design of server-less file sharing systems," SIGOPS Oper. Syst. Rev., vol. 40, 2006, pp. 359-371.

[21] R.C. Holt, "Some Deadlock Properties of Computer Systems," ACM Comput. Surv., vol. 4, 1972, pp. 179-196.

[22] D. Hughes, G. Coulson, and J. Walkerdine, "Free Riding on Gnutella Revisited: The Bell Tolls?," IEEE Distributed Systems Online, vol. 6, 2005, p. 1.

[23] D. Irwin, J. Chase, L. Grit, and A. Yumerefendi, "Self-recharging virtual currency," Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems, Philadelphia, Pennsylvania, USA: ACM, 2005, pp. 93-98.

[24] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, Mar. 2007, pp. 618-644.

[25] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, "The Eigentrust algorithm for reputation management in P2P networks," Proceedings of the 12th international conference on World Wide Web, Budapest, Hungary: ACM, 2003, pp. 640-651.

[26] M. Karakaya, İ. Körpeoğlu, and Ö. Ulusoy, "Counteracting free riding in Peer-to-Peer networks," Comput. Netw., vol. 52, 2008, pp. 675-694.

[27] X. Li and L. Liu, "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities," IEEE Trans. on Knowl. and Data Eng., vol. 16, 2004, pp. 843-857.

[28] LimeWire website, http://www.kazaa.com/, Last Access: 06/25/2010.

[29] T. Locher, P. Moor, S. Schmid and R.wattenhofer, "Free Riding in BitTorrent is Cheap," HotNet-V, 2006.

[30] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys & Tutorials,* vol. 7, 2005, pp. 72-93.

[31] R.T.B. Ma, S.C.M. Lee, J.C.S. Lui, and D.K.Y. Yau, "Incentive and service differentiation in P2P networks: a game theoretic approach," IEEE/ACM Trans. Netw., vol. 14, 2006, pp. 978-991.

[32] S. Marti and H. Garcia-Molina, "Taxonomy of trust: Categorizing P2P reputation systems," Computer Networks, vol. 50, Mar. 2006, pp. 472-484.

[33] D. Moore, C. Shannon, D.J. Brown, G.M. Voelker, and S. Savage, "Inferring Internet denial-of-service activity," ACM Trans. Comput. Syst., vol. 24, 2006, pp. 115-139.

[34] S. Naicken, A. Basu, B. Livingston, and S. Rodhetbhai, "A Survey of Peer-to-Peer Network Simulators," *Proceedings of the 7th Annual Postgraduate Symposium (PGNet '06)*, 2006.

[35] S. Naicken, B. Livingston, A. Basu, S. Rodhetbhai, I. Wakeman, and D. Chalmers, "The state of peer-to-peer simulators and simulations," *SIGCOMM Comput. Commun. Rev.*, vol. 37, 2007, pp. 95-98.

[36] P. Obreiter and J. Nimis, "A Taxonomy of Incentive Patterns," *Agents and Peer-to-Peer Computing*, 2005, pp. 89-100.

[37] M.J. Osborne and A. Rubinstein, *A course in game theory*, MIT Press, 1994.

[38] G. Owen, *Game theory*, Academic Press, 1995.

[39] P2PSim, http://pdos.csail.mit.edu/p2psim/, Last Access: 06/25/2010.

[40] PayPal website, https://www.paypal.com/, Last Access: 06/25/2010.

[41] PeerSim, http://peersim.sourceforge.net/, Last Access: 06/25/2010.

[42] ProtoPeer, http://protopeer.epfl.ch/faq.html, Last Access: 06/25/2010.

[43] Z. Runfang and H. kai, "PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing," IEEE Trans. Parallel Distrib. Syst., vol. 18, 2007, pp. 460-473.

[44] R.M. Ryan and E.L. Deci, "Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being," *The American Psychologist*, vol. 55, Jan. 2000, pp. 68-78.

[45] S. Saroiu, P. K. Gummadi and S.D. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," Multimedia Computing and Networking, 2002, pp. 156-170.

[46] Skype website, http://www.skype.com/intl/en-gb/, Last Access: 06/25/2010.

[47] SopCast websit, http://www.sopcast.com/channel/, Last Access: 06/25/2010.

[48] Y. Wang and J. Vassileva, "Bayesian Network Trust Model in Peer-to-Peer Networks," *Agents and Peer-to-Peer Computing*, 2005, pp. 23-34.

[49] YaCy, http://yacy.net/, Last Access: 06/25/2010.

[50] B. Yang and H. Garcia-Molina, "PPay: micropayments for peer-to-peer systems," Proceedings of the 10th ACM conference on Computer and communications security, Washington D.C., USA: ACM, 2003, pp. 300-310.

[51] M. Yang, Z. Zhang, X. Li, and Y. Dai, "An Empirical Study of Free-Riding Behavior in the Maze P2P File-Sharing System," Peer-to-Peer Systems IV, 2005, pp. 182-192.

[52] K. Zhang, N. Antonopoulos, "Towards a Cluster Based Incentive Mechanism for P2P Systems," CCGRID 2009, Shanghai, China, 2009.

Constructing a Stable Virtual Peer from Multiple Unstable Peers for Fault-tolerant P2P Systems

Masanori Shikano, Kota Abe, Tatsuya Ueda, Hayato Ishibashi and Toshio Matsuura
Graduate School for Creative Cities, Osaka City University
Osaka, Japan
Email: {shikano,k-abe,tueda,ishibashi,matsuura}@sousei.gscc.osaka-cu.ac.jp

***Abstract*—P2P systems must handle unexpected peer failure and leaving, and thus it is more difficult to implement than server-client systems. In this paper, we propose a novel approach to implement P2P systems by using *virtual peers*. A virtual peer consists of multiple unstable peers. A virtual peer is a stable entity; application programs run on a virtual peer are not compromised unless a majority of the peers fail within a short time duration. If a peer in a virtual peer fails, the failed peer is replaced by another (non-failed) one to restore the number of working peers.**

The primary contribution of this paper is to propose a method to form a stable virtual peer over multiple unstable peers. The major challenges are to maintain consistency between multiple peers, to replace a failed peer with another one, and to communicate with a virtual peer, whose member peers are dynamically changed. For the first issue, the Paxos consensus algorithm is used. For the second issue, the process migration technique is used to replicate and transfer a running process to a remote peer. For the last issue, communication protocols based on application level multicast are introduced. Furthermore, the relation between the reliability of a virtual peer and the number of peers assigned to a virtual peer is evaluated.

The proposed method is implemented in our *musasabi* P2P platform. An overview of *musasabi* and its implementation is also given.

***Keywords*—peer-to-peer systems; fault tolerance; Paxos consensus algorithm; process migration; strong mobility**

I. Introduction

In comparison to server-client systems, peer-to-peer (P2P) systems provide superior fault tolerance because P2P systems do not have servers, which are single points of failure. However, because peers in a P2P system are unstable (they fail or leave unexpectedly), P2P systems are more difficult to implement than server-client systems. (In this paper, peer leaving is considered a kind of peer failure and the two are not distinguished.) For example, the following are common measures to achieve fault tolerance in P2P systems: peer failure detection, data replication over multiple peers, and managing multiple pointers as a precaution against peer failure. While such measures are crucial for practical P2P applications to provide stable services, implementing such measures is delicate work and a troublesome burden for developers.

In this paper, we propose a novel approach to ensuring fault tolerance in P2P systems. In the proposed method, *virtual peers* are formed by grouping multiple peers on a P2P network. A virtual peer is an entity for running an application program. Similar to mirroring on file systems, each peer of a virtual peer serves as a part of a redundant system. The running application on a virtual peer is not compromised when some of the peers fail. In addition, when a peer fails, the number of peers is restored by replacing the failed peer with another (non-failed) one. Therefore, virtual peers rarely fail.

Using virtual peers in a P2P system is of great benefit. Developing P2P systems based on virtual peers (i.e., using virtual peers as a building block of P2P systems) is more straightforward than conventional P2P systems; because the possibility of virtual peer failure is negligible, P2P systems constructed over virtual peers do not have to implement measures against peer failures. Furthermore, virtual peers can be used for the following applications.

- Replacing central servers on hybrid P2P systems: One disadvantage of hybrid P2P systems is that central servers are single points of failure. This single point of failure can be eliminated by replacing the server with a virtual peer.
- Using virtual peer as a super peer: In super peer-based P2P systems, super peers have a more important role than normal peers and are expected to be stable. Using a virtual peer as a super peer reduces the possibility of super peer failure and thus the system can be more stable.

The proposed method has been implemented in the prototype of our P2P platform *musasabi*[1]. *Musasabi* is implemented in pure Java.

The rest of this paper is organized as follows. Section II presents the proposed method. Section III provides an overview of *musasabi* and an explanation of the implementation. Section IV describes communication protocols for virtual peers. Section V discusses the proposed method. Section VI gives related work, and Section VII summarizes our conclusion.

II. Proposed method

In the proposed method, multiple peers chosen from the P2P network are grouped to form a *virtual peer* (Figure 1). The peers that form a virtual peer are called *member peers*.

An application program running on a virtual peer is called a *virtual process*. A virtual process corresponds one-to-one with a virtual peer. A virtual process does not stop even if some of the member peers fail. We describe the details below.

[1] *Musasabi* is a type of flying squirrel found in Japan.

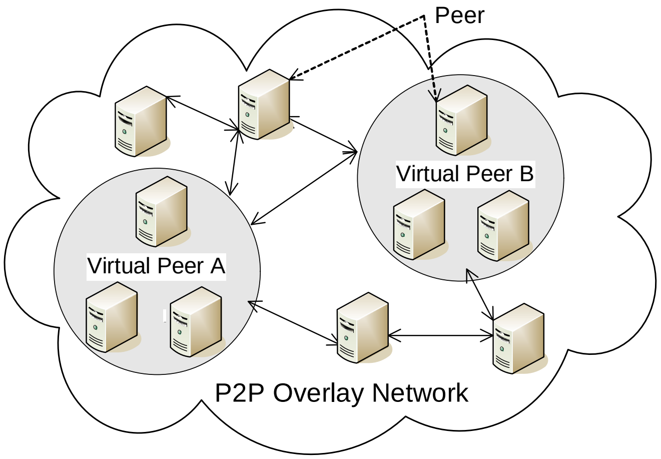


Figure. 1. Peers and Virtual Peers

A. Choosing member peers

In order to start a virtual peer, the initial member peers of the virtual peer must be chosen. Currently, we assume that no nodes in the P2P network decline to be a member peer of any virtual peer. A single peer is allowed to be a member of multiple different virtual peers.

To choose a *good* member peer, the following criteria must be considered: the peer's stability, network distances to the other member peers, network bandwidth, CPU speed and load average, memory size, etc. However, we choose member peers randomly for now.

Choosing a peer randomly can be implemented using a Skip Graph overlay network [3]. We assume that each peer has its own ID (peer ID). On initializing a peer, its peer ID is registered in the Skip Graph, which is shared among all peers. In order to randomly choose a peer, generate a random number r and search for a peer from the Skip Graph whose ID is closest to r.

B. Achieving fault-tolerance of virtual process

To make it possible for a virtual process to continue its execution even if some of the member peers fail, the state of a virtual process must be replicated over multiple member peers. In order to make it easy to develop application programs, we choose the following approach.

Each member peer of a virtual peer simultaneously and redundantly executes the same application program, as a *process*. Each of the processes has the complete replica of the virtual process state. This is depicted in Figure 2.

In order to maintain the state of each process identically, the process must be a state machine. The process state must be changed only by external input (messages) that the process receives. Moreover, the sequences of external input (messages) received by each process are controlled to be completely identical. This approach is known as *State Machine Replication* [4].

In this approach, application programs can be quite simple; just process the received messages in order. No commit protocol is required.

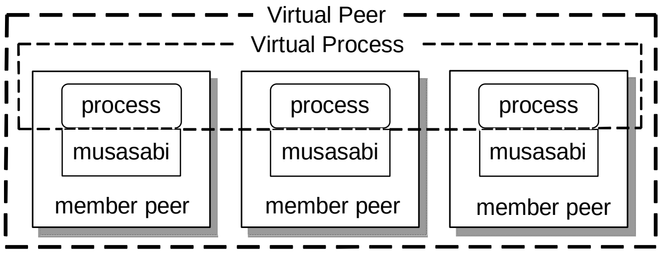


Figure. 2. Virtual peer and virtual process

C. Ensuring identical message sequences

When multiple nodes transmit messages to multiple receiving nodes via the Internet, the message sequences received among the receiving nodes are not necessarily identical. In order to ensure that multiple nodes receive messages in identical order (in other words, *atomic broadcast* or *total order broadcast*), we use the Paxos consensus algorithm because it is proven and well described in literature [4][5].

Paxos is a distributed algorithm to form a consensus between multiple participants (in this case, participants are peers) on an unreliable network. Paxos ensures that all participants (that have not failed) eventually choose a single value that one of the participants proposed. In Paxos, a consensus is reached when a majority of the participants accept the proposed value. Paxos can be extended to a series of values (called Multi-Paxos). In this paper, Paxos and Multi-Paxos are not distinguished.

In order for the Paxos protocol to progress, values must be proposed by only one peer (leader peer). The leader peer is elected from the member peers by a leader election algorithm. Note that the Paxos algorithm guarantees only one value is chosen even if multiple leaders are present. (This is called the *safety* property and is important because leader election may fail to elect a single leader.)

An outline of the Paxos protocol is as follows. The leader peer initially evaluates the current condition by broadcasting *Collect* messages. (Message names are based on the literature [5].) After the leader receives the *Last* message from a majority of the peers, the leader peer can propose a value by broadcasting *Begin* messages. A value is agreed upon when a majority of the peers reply with an *Accept* message. If a value is agreed upon, the leader broadcasts the *Success* message to announce the consensus. Hereafter, consensus about a sequence of values is reached by repeating the Begin–Accept–Success sequence. *Begin*, *Accept* and *Success* messages contain a sequence number to identify each proposal.

As we will describe in Section IV, when a message is sent to a virtual peer, its leader peer receives the message through ALM (Application Level Multicast). In order to make the message order received by each member peer identical, the leader peer assigns a sequential number (i) to the message and proposes the message as the i^{th} value with the Paxos protocol.

On each of the member peers, the agreed-upon message is passed to the process in sequence number order. Note that because the Paxos protocol runs asynchronously, consensus is not always reached in the sequence number order when multiple proposals have been made simultaneously. Therefore, if the sequence number of a received message is not continuous with the previous one (i.e., if there is a *gap* between them), the process must wait until the gap is filled. If the gap is not filled for a long time, a retransmission request is sent to the leader peer.

D. Handling peer failure

If a member peer fails, the remaining member peers must replace the failed peer with another one to keep the number of member peers constant. Otherwise, eventually the virtual peer will *evaporate*.

In order to detect member peer failure, each member peer carries out watchdog monitoring on the others. When a peer detects that another peer has failed, the failed peer is replaced by another one. As mentioned in Section II-A, a substitute peer is randomly chosen from the peers in the P2P network.

All of the member peers must maintain a consistent view of the member configuration. Paxos is also used to change the peer configuration in order to ensure consistent updating of this view. The procedure is as follows.

i) If the failed peer is the leader, a new leader is elected (as explained later in Section III-D).

ii) The leader peer chooses another peer p from the P2P network.

iii) The leader peer proposes a peer configuration change (swapping the failed peer with p) using Paxos.

iv) When the proposal has been agreed upon, the peer configuration is changed by all of the member peers. If the consensus fails, return to 2) after waiting briefly.

v) p must execute the same process, whose state must be identical to the ones on other member peers. For this reason, the *process migration technique* is used to replicate the running process on the leader peer to p (see Section III-A).

In order to reach consensus in Paxos, a majority of the peers must be functional while replacing a failed peer. Otherwise, the virtual peer fails. Reliability of virtual peers is discussed in Section V-A.

Note that some of the member peers might not be aware that the configuration has been changed, partly because the *Success* message is lost, and partly because only a majority of member peers are required to reach a consensus. If such a peer becomes the leader, consensus might not be reached because it may not be able to reach a majority of member peers. To avoid this situation, member configuration is periodically exchanged among the member peers. The details of this issue is out of the scope of this paper. We will describe this in another article.

E. Communication with a virtual peer

Communication protocols for virtual peers are separately discussed in Section IV because they require much space.

III. P2P PLATFORM *musasabi*

We implemented the proposed method in the prototype of our *musasabi* P2P platform. *Musasabi* is written in Java and is intended to be a platform for implementing P2P services. Each instance of *musasabi* can be regarded as a peer.

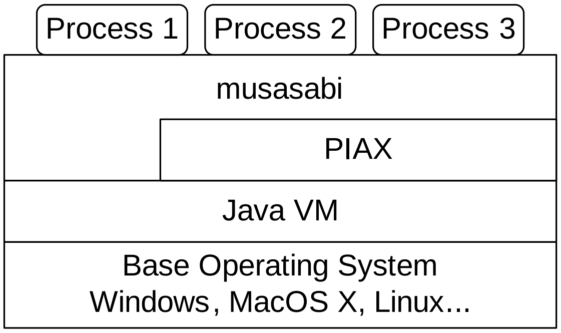


Figure. 3. Configuration of *musasabi*

On *musasabi*, an application program, also in Java, can be executed as a *process*. To implement a P2P service using *musasabi*, a process on *musasabi* communicates with other processes on remote *musasabi* peers.

Because a process may be transferred from a remote peer using process migration, measures against malicious programs are necessary in order to protect a local node. For this reason, processes are executed in the Java sandbox mechanism.

Musasabi uses another P2P platform, *PIAX* [6], for P2P networking. Among the functions provided by PIAX, the Skip Graph overlay network and the ALM service are important for *musasabi*. The configuration of *musasabi* is shown in Figure 3.

Musasabi supports the virtual peer mechanism described above; we describe the details below.

A. Implementation of process migration

In order to implement a virtual peer, *process migration*, a function to transfer a running process from a node to another node, is required.

Process migration techniques are classified into two classes: *weak mobility* and *strong mobility* [7]. In weak mobility, only program code and data fields (the values of global variables) are transferred, whereas in strong mobility, in addition to these, the execution context of threads (the contents of the thread stack and the value of its program counter) is also transferred. Strong mobility makes it possible to describe mobile programs in a natural form. *Musasabi* supports strong mobility, whereas the standard Java does not.

Musasabi provides the following APIs for process migration. An example of their use is shown in Figure 4.

go(peer)
The calling process is migrated to the specified peer.

fork(peer)
The calling process is replicated and the replicated process is transferred to the specified peer. This API resembles the *fork* system call in UNIX.

1) Implementation of strong mobility: Some research has been done on implementing strong mobility in Java. As stated above, in order to achieve strong mobility, program code, data fields, and execution contexts must be transferred. The program code can be transferred as class (or jar) files. The Java

```
// start on peer p0
PeerId p1 = ...;
go(p1); // move to peer p1
...; // run on p1
PeerId p2 = ...;
// duplicate the process and transfer to peer p2
if (fork(p2) == null) {
  ...; // run on p2
} else {
  ...; // run on p1
}
```

Figure. 4. A sample program to demonstrate process migration APIs of *musasabi*

```
class MyClass() implements Runnable {
  public void run() {
    block1;
    Continuation.suspend(); // suspend here
    block2;
  }
};
// start from MyClass#run().
// it will be suspended after executing block1 and
// the executing state is saved to c.
Continuation c
  = Continuation.startWith(new MyClass());
// resume from block2.
c = Continuation.continueWith(c);
```

Figure. 5. A sample program of Javaflow's Continuation

serialization mechanism can be used to transfer the data fields. The remaining problem is the method for transferring the execution context. The following methods are known [8]: (1) modifying the Java Virtual Machine (JVM) [9], (2) extending JVM using the Java Native Interface that is provided by JVM [10], (3) translating Java byte code [8][11], and (4) translating Java source code [12].

In a P2P network, because each peer is independently managed, it is difficult to use a non-standard JVM or native code. Thus, (1) and (2) are rejected. *Musasabi* adopts method (3), using *Javaflow*, a library for realizing *Continuation* in Java, because it is publicly available [13]. The idea of using Javaflow for implementing strong mobility is also described in the literature [8].

Javaflow allows Java programs to save a program execution context as an object called *continuation*, which contains information of the stack frame of a thread. Saved context can be restarted later (Figure 5). To make it possible to save and restore an execution context, Javaflow translates the byte code of the program based on certain rules. Both Javaflow and translated byte codes run on a standard JVM.

Because continuation of Javaflow is *Serializable*, it is possible to restart the saved context on a different node. Thus, strong mobility can be implemented using Javaflow. (Note that to serialize a continuation, all objects stored in the saved context also must be *Serializable*.)

2) Migration of a multi-threaded process: When a multi-threaded process is migrated by using the strong mobility model, the execution contexts for all of the threads must be stored and recovered. However, by using a method in which JVM is not modified (including the case of using Javaflow), the execution context cannot be forcibly acquired from outside of the thread; in order to acquire the execution context of a thread, the thread itself must call an API to capture the context. Therefore, it is difficult to migrate all the threads in a process when a thread requests a process migration.

Musasabi solves this problem by providing *Coroutines* instead of Java threads. Coroutines are a unit of parallel processing, similar to threads. While threads may be executed simultaneously (in parallel), coroutines are executed in turn. While context switching between multiple threads is preemptive, context switching between coroutines is voluntary. When a coroutine is running, other coroutines are not executed until the running coroutine itself yields the execution right (by executing yield() or some other API). Therefore, when a coroutine requests process migration, all other coroutines should have yielded, and thus the execution state of all the coroutines can be transferred to the destination peer. The APIs for coroutines in *musasabi* are similar to the standard Java thread APIs.

Javaflow is also used to implement coroutines in *musasabi*. Context switching between coroutines is done by saving a coroutine context and restoring another coroutine context. A simple coroutine scheduler is also implemented.

B. Application Level Multicast

As we will describe in Section IV, the proposed method uses ALM for sending messages to a virtual peer. *Musasabi* uses the ALM service of PIAX, which is built on top of Skip Graph. The implementation of ALM is simple. If a peer joins a multicast group whose ID is g, the peer registers g as a key in the Skip Graph. A message sent to the multicast group g is routed over the Skip Graph and distributed to all peers that registered the key g.

Note that communication within a virtual peer is directly performed on the IP layer.

C. Starting a virtual peer

A virtual peer (and the corresponding virtual process) is started as follows. First, a user starts a normal (non-virtual) process on a peer on which *musasabi* is running. This peer will be the initial leader. The process requests *musasabi* to become a virtual process. An ID for the virtual peer is assigned using a random number generator. Then, in order to form a virtual peer, *musasabi* randomly chooses the specified number of member peers from the P2P network. The initial leader peer joins the multicast group whose ID is the same as the ID of the virtual peer.

D. Leader election

If a leader peer fails, the Paxos protocol needs a new leader. In *musasabi*, a new leader is elected as follows.

Failure of a member peer is detected by timeout of the *Keep Alive* message, which each of the member peers periodically sends to others. Each member peer maintains the (*Alive* or *Dead*) status of the other member peers.

Each member peer has a unique number, *mnum* (member number). All member peers know the *mnum* of the others. If the number of the member peers is n, the *mnum* of each initial n peer will be $0, 1, \ldots, n-1$. When a peer replacement is agreed upon, the *mnum* of the new peer must be assigned without duplication. In order to guarantee the uniqueness of *mnum* even in a multiple leader situation, we use the Paxos sequence number which is used in the replacement proposal as the new *mnum*. (The initial Paxos sequence number is adjusted not to be duplicated with the initial *mnum*s.)

If a leader peer fails, the next leader is the member peer that is alive and whose *mnum* is the smallest. When a member peer detects that the leader peer has failed, the peer determines whether the peer should be the next leader (check whether the peer itself has the smallest *mnum* among all live member peers). If the peer believes that it is the next leader, the peer broadcasts *Collect* messages (Section II-C). If the peer receives *Last* messages from a majority of the peers, it joins in the multicast group of the virtual peer and broadcasts *IamLeader* messages to announce that it is the new leader. If the peer fails to collect enough *Last* messages (this could happen if another peer also believes that it should be the next leader), and no *IamLeader* message has been received from other member peers, the peer retries broadcasting *Collect* messages after waiting for some random period.

E. Replacing a failed peer

When a configuration change proposal (swap the failed peer with a substitute peer p) is agreed-upon, the leader peer uses the fork() API to replicate the process running on the leader peer onto peer p.

An outline of a replacement sequence for a failed peer is shown in Figure 6. The leader peer detects that Peer#2 has failed because of a *Keep Alive* timeout, and chooses Peer#3 as a substitute. The leader proposes replacing Peer#2 with Peer#3. When the proposal is agreed upon, the member configuration is changed in each member peer. In addition, the leader peer replicates (fork) the process onto Peer#3. Subsequently, the virtual peer (process) is served by peers #0, #1, and #3.

F. Interaction between application programs and Paxos

Interaction between application programs and Paxos is depicted in Figure 7. Messages that the leader peer receives via ALM are not automatically proposed by *musasabi*. All messages received via ALM are sent to the process on the leader to determine whether the message should be proposed or not. This is because messages that do not affect the process' status, such as simple read requests, may be processed without proposing. (However, not all read requests can be processed without proposing; if multiple leaders are present, a leader cannot guarantee that it has the latest state. Therefore, only messages that do not depend on the latest state can be processed without proposing.)

IV. Communication protocols for virtual peers

To implement services using virtual peers, virtual peers must be able to communicate with other peers (either virtual or non-virtual). Communication between virtual peers is a kind of group communication, which is a means for multi-point to multi-point communication. In this section, we describe communication protocols for virtual peers.

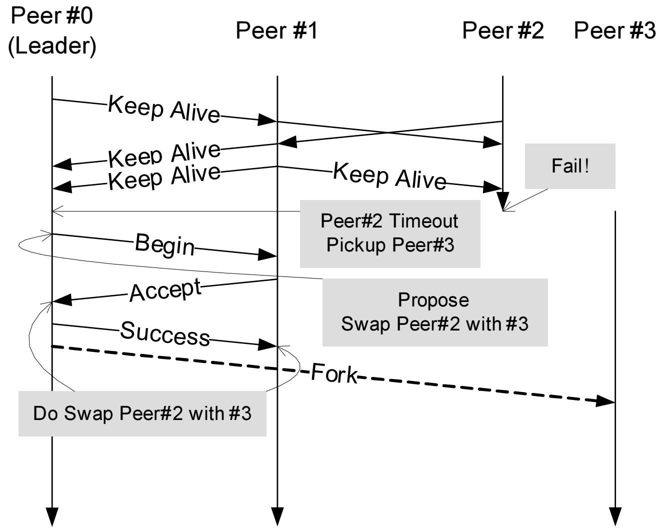


Figure. 6. Sequence for replacing a failed peer

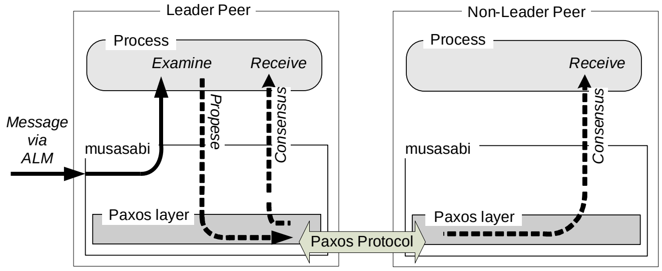


Figure. 7. Interaction between Applications and Paxos

To clearly distinguish the two, we call a peer that is not a virtual peer a **normal peer**. We assume that a peer (either normal or virtual) sends a request message to a virtual peer and receives a response message. We also assume that messages in the underlying network may be delayed or lost. In our protocols, communication channels between a sender and a receiver are not FIFO, i.e., if multiple messages are asynchronously sent, their order at the receiver may be changed.

The structure of our protocols is depicted in Figure 8. Note that basically each application in the figure is executed on a different node.

We describe the communication protocol between a normal peer and a virtual peer in Section IV-A, and between virtual peers in Section IV-B.

A. Communication between a normal peer and a virtual peer

First, we discuss the protocol in the case where a normal peer p sends a message m to a virtual peer v and v returns a response message q to p.

1) Delivery message to a virtual peer: As described in Section II-D, member peers of a virtual peer are not fixed. Thus, it is not a straightforward task to deliver messages to a virtual peer from outside of the virtual peer.

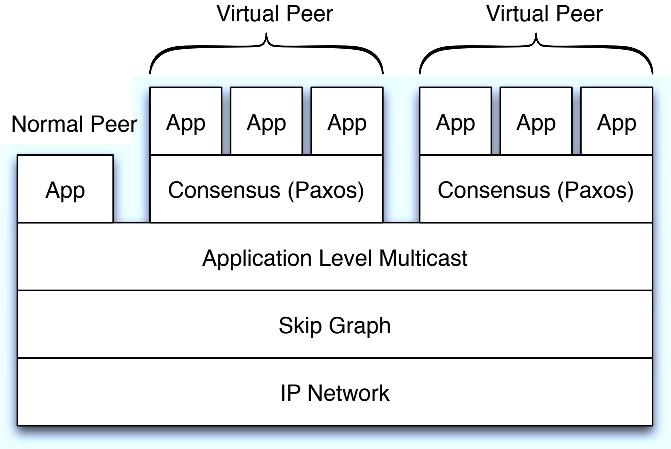


Figure. 8. Protocol Structure

This issue can be solved using ALM. All peers of a virtual peer join in a multicast group specified by the ID of the virtual peer. When a peer sends a message to a virtual peer, the peer multicasts the message over the multicast group of the virtual peer. The sender peer does not have to know the addresses of each member peer. (Note that multicast is required because there might be multiple leaders in a single virtual peer (See Section II-C).)

2) Retransmission of message: When the leader peer of v receives m through ALM, it proposes m with the Paxos protocol, as described in Section II-C. (All the other peers of v do not propose m). Each process of each member peer receives and processes the agreed-upon message in sequence number order. However, proposed messages might not be agreed upon because, for example, the message might be lost, or even if the message arrives at the leader, the leader might leave before proposing the message. For this reason, p has a response timer and periodically retransmits the message m until p receives q.

3) Message ID: As messages might be retransmitted, a receiver may receive duplicated messages. In order to detect such duplicated messages, a message ID is assigned to each message. To uniquely assign a message ID to a message, message IDs are generated by combining the peer ID of p and the sequence number of the message. Message IDs are also used on sender peers for finding a request message from a received response message.

4) Sending response: On receiving m on v, each process on v must call an API to send a response message q to p. Note that all the responses that each process produces must be identical in our execution model (see Section II-B). To reduce network traffic, only the leader peer actually sends the response.

5) Retransmission of response: Response messages from the leader peer also might be lost. In such a case, p retransmits m to v when the response timer expires. However, since m has already been processed by v in this case, v should not reprocess the retransmitted message and just resend the response message previously sent. For this reason, v keeps response messages that it has sent for a while.

6) Protocol: We propose the following protocol based on the discussion above.

Behavior of sender: The peer p generates a message ID for m and multicasts m to the multicast group of v. The peer p retransmits m if no response message for m is received within a fixed period of time.

Behavior of a receiver: The algorithm for a receiver virtual peer is implemented on each member peer.

On each member peer, a *history* table, which keeps the status of received messages, and a *response* table, which keeps response messages, are maintained. These are used for preventing the processing of the same message more than once. These tables can be implemented as hash tables whose key is a message ID. The entry of the *history* table is one of the following:

INITIAL
The message has not yet been received (default).

PASSED
The message has been passed on for processing.

REPLIED
The message has been passed on for processing and its response message has been sent.

The algorithm of a receiver is shown below.

i) On receiving m through ALM, only the leader peer executes the following steps. (all other peers just ignore m).
   a) Get the *status* of m from the *history* table.
   b) If the *status* is INITIAL, then propose m using Paxos.
   c) If the *status* is PASSED, then just ignore m.
   d) If the *status* is REPLIED, then send the response message recorded in the *response* table to p.

ii) On receiving the *Success* message of m (notification of agreement), all the member peers execute the following steps.
   a) Get the *status* of m from the *history* table.
   b) If the *status* is INITIAL, then (1) record the status of m as PASSED in the *history* table, and (2) pass m to the process on the peer.
   c) If the *status* is not INITIAL, then just ignore m.

iii) The process that received m must call an API to send a response message q to p. In the API, following steps are executed.
   a) Record the status of m as REPLIED in the *history* table.
   b) Record q in the *response* table.
   c) Only the leader peer sends q.

7) Examples of the message sequence: An example of message sequence in which no message retransmission occurs is shown in Figure 9, and another example in which message retransmission occurs is shown in Figure 10. The bold short dashed lines in the figures mean the period the process is handling the message.

B. Communication between virtual peers

Next, we discuss the protocol in the case where a virtual peer s sends a message m to another virtual peer r and r

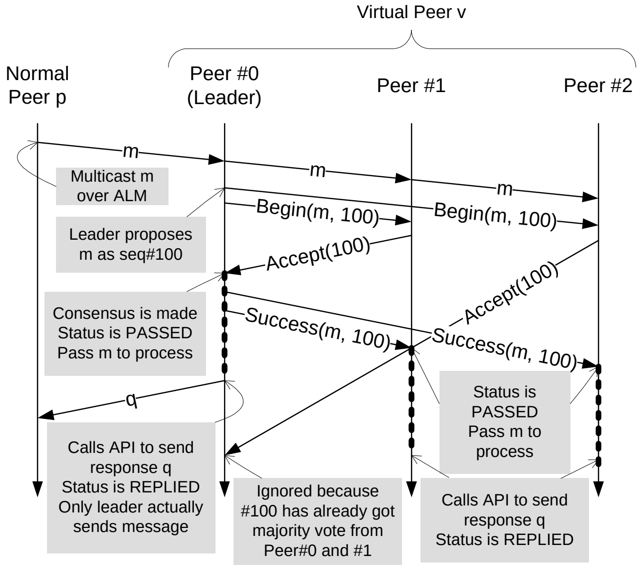

Virtual Peer v
Normal Peer p
Peer #0 (Leader)
Peer #1
Peer #2
m
m
m
Multicast m over ALM
Begin(m, 100)
Begin(m, 100)
Leader proposes m as seq#100
Accept(100)
Accept(100)
Consensus is made Status is PASSED Pass m to process
Success(m, 100)
Success(m, 100)
q
Status is PASSED Pass m to process
Calls API to send response q Status is REPLIED Only leader actually sends message
Ignored because #100 has already got majority vote from Peer#0 and #1
Calls API to send response q Status is REPLIED

Figure. 9. Example of message sequence between a normal peer and a virtual peer (without retransmission case)

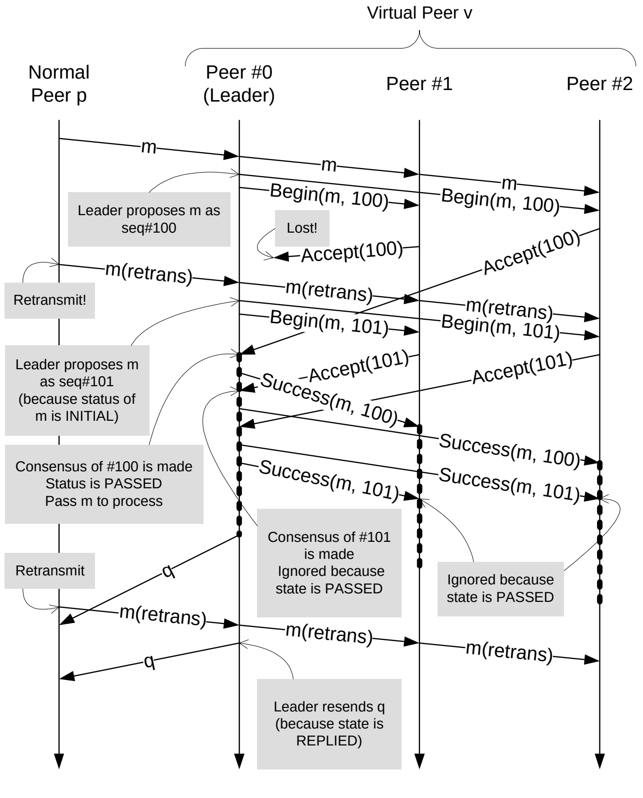


Figure. 10. Example of message sequence between a normal peer and a virtual peer (with retransmission case)

returns a response message q to s. Note that if s sends m to r, s must have received a *trigger* message M, which has been agreed upon, because virtual processes are executed in message-driven models. We assume that each process of s calls an API to send m to r by receiving M.

Communication between virtual peers is basically the same as that between a normal peer and a virtual peer. A leader peer of s plays the role of a sender peer p in Section IV-A. In addition to that, the following points should be considered.

1) Sending response: A response message from a virtual peer r also must be sent with ALM because member peers of s may be changed while s is communicating with r.

2) Agreement of response: Response messages also must be ordered identically at all of the member peers because message receiving order may affect the internal state of a virtual process. Therefore, the leader of s must propose every received response message with the Paxos protocol. A response message is processed after it is agreed upon.

3) Member peers change: Here we consider a case where the leader peer s_l of s leaves before receiving a response message from r, after M is agreed upon. Suppose the following scenarios.

- s_l leaves before sending m to r: this includes the two cases shown below.
  - (Case 1a) The next leader s'_l of s has processed M (i.e., M has been passed to the process on s'_l) before it becomes a leader.
    In this case, when s'_l called the API to send m to r, no messages were actually sent by s'_l because it was not a leader at that time. However, when the response timer expires, s'_l must actually send m this time. For this reason, all the peers should record their sending messages whether they are leaders or not.
  - (Case 1b) s'_l processes M after it becomes a leader. In this case, m is sent to r by s'_l when it processes M.
- s_l leaves after m is sent to r: it is possible to classify it into two groups as follows.
  - (Case 2a) s'_l has been elected as a leader when q arrives from r:
    In this case, s'_l simply proposes q with Paxos.
  - (Case 2b) s_l has already left but no leader has been elected when q arrives:
    In this case, because no peer in s proposes q and thus q is lost, the next leader (s'_l) retransmits m to get q as soon as it is elected.

4) Protocol: We propose the following protocol based on the discussion above.

Behavior of sender: The algorithm for a sender virtual peer is implemented on each member peer.

i) In the API to send m to r, the following steps are executed. (Note that all the processes in s call this API on receiving M.)
   a) In our execution model, all member peers must independently generate the same message m. Therefore, the message ID of m also must be identical. Thus, message IDs are generated based on not the ID of each member peer but the ID of virtual peer s.
   b) Only the leader peer multicasts m to the multicast group of s.
   c) Record m for retransmitting and start the response timer.
   d) When the response timer expires, only the leader peer retransmits m.

ii) On receiving q through ALM, s processes q according to the receiver protocol i) and ii) described in Section IV-A6. All the member peers of s stop the timer after step ii).

If the leader peer that sent m leaves and the new leader detects that no response message for m has been received, the new leader retransmits m, as described in Section IV-B3.

Behavior of receiver: The behavior of receiver r is almost the same as that of receiver v described in Section IV-A6, except that the leader peer of r multicasts the response message to the multicast group of s.

5) Example of message sequence: An example of a message sequence between virtual peers is shown in Figure 11.

V. Discussion

A. Reliability of virtual peers

In this section, we discuss the relation between the reliability of a virtual peer and the number of its member peers.

Let t be elapsed time since the start of a virtual peer, m the number of member peers, and T the maximum time required to swap a failed peer (from the moment of a peer failure until fork() is done). Because the Paxos protocol requires that a majority of member peers survive to reach a consensus, and because Paxos is used for changing a peer configuration, a virtual peer looses its functionality when $\lfloor (m+1)/2 \rfloor$ (denoted as n) peers have failed during T. Assuming that each peer fails independently and that the intervals of peer failure are exponentially distributed, let λ' be the peer failure rate per unit of time. λ' can be expressed using the duration of half of the peers failing (denoted as t_{half}), as $e^{\lambda' t_{\text{half}}} = 0.5$. The reliability of a single peer $R'(t)$ is expressed as $R'(t) = e^{-\lambda' t}$. Because the probability that n peers fail within time duration T is $(1-R'(T))^n$, the failure rate of a virtual peer λ is expressed as $\lambda = (1 - R'(T))^n/T$. Note that the reliability of a virtual peer $R(t)$ is given by $R(t) = e^{-\lambda t}$.

The value of T mainly depends on the *Keep Alive* message interval. We pessimistically assume that $T = 60$ (sec).

Now, we show two graphs (Figure 12 and Figure 13). Figure 12 shows the reliability of virtual peers for a different number of member peers versus t. Because the peer failure rate depends on the environment, here we assume $t_{\text{half}} = 1$ (hour). In this case, a virtual peer is practically stable if it consists of seven member peers.

Next, we vary the peer failure rate and show the resulting MTTF (Mean Time To Failure) of a virtual peer (expressed as λ^{-1}) in Figure 13. Note that the x-axis is expressed in t_{half}. The graph indicates that, even in the high peer failure rate environment, virtual peers can be practically stable by increasing the number of member peers. (We modestly assume that in the real environment $t_{\text{half}} > 10$ (min) because only 1.56% of peers remain after one hour if $t_{\text{half}} = 10$, which seems too pessimistic.)

B. Overhead of the proposed method

In this section, we discuss the overhead of the proposed method.

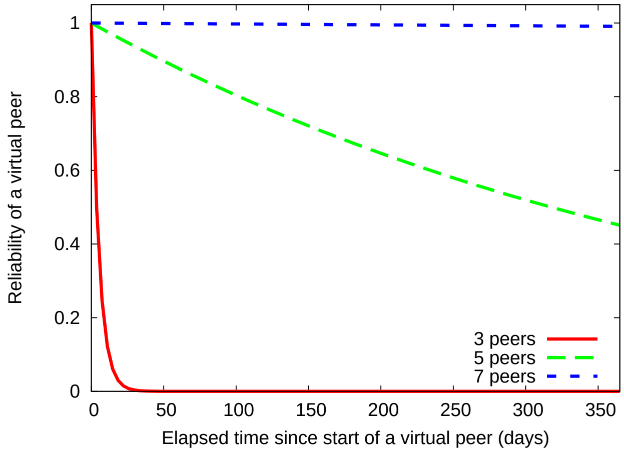


Figure. 12. Reliability of a virtual peer.

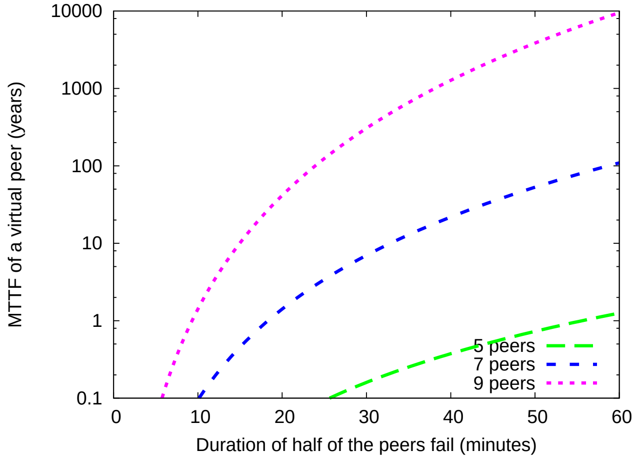


Figure. 13. MTTF of a virtual peer versus peer failure rate.

1) Overhead of the Paxos: The Paxos protocol imposes latency on processing a request sent to a virtual peer. The latency is mostly affected by the RTT (Round Trip Time) between the leader and other member peers. (Note that the latency is not proportionally increased with the number of member peers because a leader peer can send *Begin* messages to each member peer without waiting.)

2) Overhead of communication: The ALM, used for sending a message to a virtual peer, also imposes extra overhead. Because *musasabi* uses ALM built on Skip Graph, sending a message over ALM requires $O(\log n)$ hops where n is the number of virtual peers. This overhead is common in P2P systems but can be reduced if a peer subsequently communicates with the same virtual peer. A peer first uses ALM to send a message and receives the configuration of the virtual peer; subsequent messages can be sent directly to the leader peer (until it fails).

3) Overhead of watchdog monitoring: If one wants to make the maximum time required to swap a failed peer below 60 seconds (see Section V-A), the period for sending *Keep Alive* messages will be around 20 to 30 seconds. This overhead is small and will not be a problem, we believe.

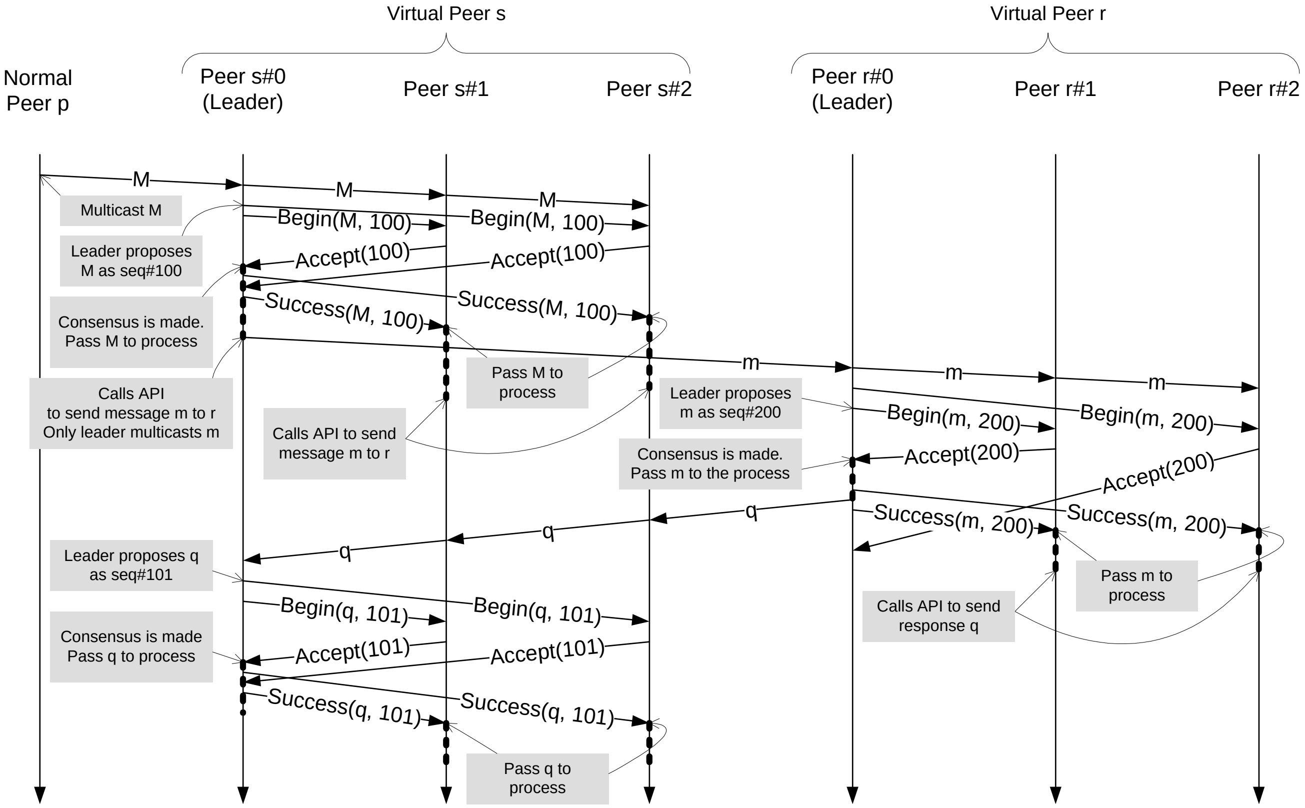


Figure. 11. Example of message sequence between virtual peers

4) Coroutines and byte code translation: The performance of coroutines is inferior to that of threads, especially on multi-CPU machines. In addition, the byte code translation technique used for achieving strong mobility may degrade the performance of application programs. We believe these will not be a problem unless virtual peers are used for computational purposes (such as in P2P-Grid).

C. Leader failure

If the leader peer fails, execution of the virtual process stops until the new leader is elected. Therefore, leader failure must be quickly detected. Multi-coordinated Paxos [14], which allows multiple leaders and thus improves the availability of the system, may relax this problem.

D. Network partitioning

If network partitioning occurs and no fragment of the network contains a majority of the member peers, the virtual peer looses its functionality. (We assume an odd number of member peers.) In this case, the virtual peer must wait until the network has recovered. Note that this situation happens only if the network is split into three or more fragments.

E. Scalability

Using a virtual peer does not contribute to scalability problems of P2P systems. Scalability can be obtained by using multiple virtual peers.

F. Security

A virtual peer might be compromised either if a malicious node is selected as the leader peer or if a member peer violates the Paxos protocol. The Byzantine Paxos protocol [15] may ameliorate this issue but this kind of protocol imposes big overhead. This is beyond the scope of this paper and will be investigated in the future.

VI. Related work

There are few research efforts that use the Paxos algorithm in P2P-related contexts. *Scalaris*, a distributed transactional key/value store written in *Erlang*, uses an adapted Paxos protocol to implement replication and ACID properties (atomicity, concurrency, isolation, durability) [16]. In the field of cloud computing, Google's *Chubby*, an implementation of distributed lock service, uses Paxos in order to make a fault-tolerant distributed database with multiple computers [17][18]. However, these systems do not provide fault-tolerance of general applications.

In the field of Grid computing, *Vigne*, a grid middleware for dynamic large scale grids, uses similar techniques (such as execution on redundant nodes, state machine replication using atomic multicasting over the Paxos protocol) to achieve fault-tolerance of its *Application Manager* [19]. The distinguished differences are that (1) *Vigne* uses the Pastry overlay network [20] for routing messages to the manager node (which corresponds a leader peer in *musasabi*), whereas *musasabi* uses ALM over Skip Graph, and that (2) replicas of the manager node are selected from the leaf set (peers whose ID's are close) of Pastry. The drawbacks of this approach are that there is less flexibility on choosing replica members and that it incurs reconfiguration of replica members not only in case of node failure but also in case of node addition (in the leaf set) [19].

In the literature [21], Mena et al. proposes a group communication scheme based on consensus. While their protocol stack is similar to ours, their discussion is generic and not specific for P2P network. The contribution of our paper is that we revealed the detail of how to implement such architecture in P2P network.

VII. Conclusion and Future Work

P2P systems must handle unexpected peer failure and leaving, and thus they are more difficult to implement than server-client systems.

In this paper, we proposed a method to construct a stable virtual peer from multiple unstable peers. An application program running on virtual peers is not compromised unless a majority of the underlying unstable peers fail within a short time duration. Moreover, application programs of this method can be quite simple. Virtual peers achieve superior fault-tolerance by integrating the Paxos consensus algorithm and process migration technique. In addition, we proposed communication protocols for virtual peers based on application level multicast, and analyzed the relation between the reliability of a virtual peer and the number of peers assigned for a virtual peer. The result indicates that our method appears promising.

The proposed method can be used for reducing development costs, and for improving stability, of P2P systems.

The proposed method has some overhead as described in Section V-B. Quantitative evaluation of each overhead is one of our future work. Other future work includes: (1) improving the method for choosing good member peers, (2) analyzing and improving security of virtual peers, and (3) evaluating the method on the Internet.

Acknowledgment

We would like to thank Dr. Mikio Yoshida, the developer of PIAX at BBR Inc., for giving helpful advice regarding PIAX. This research was partially supported by National Institute of Information and Communications Technology (NICT), Japan.

References

[1] Kota Abe, Tatsuya Ueda, Masanori Shikano, Hayato Ishibashi, and Toshio Matsuura. Toward Fault-tolerant P2P Systems: Constructing a Stable Virtual Peer from Multiple Unstable Peers. In *AP2PS '09: Proc. of 1st Intl. Conf. on Advances in P2P Systems*, pages 104–110. Information Processing Society of Japan, 2009.

[2] Masanori Shikano, Tatsuya Ueda, Kota Abe, Hayato Ishibashi, and Toshio Matsuura. Communication Methods for Virtual Peers on musasabi P2P Platform. Technical Report 2, IPSJ DPS-139, 2009 (in Japanese).

[3] James Aspnes and Gauri Shah. Skip graphs. *ACM Trans. on Algorithms*, 3(4):1–25, 2007.

[4] Leslie Lamport and Keith Marzullo. The part-time parliament. *ACM Trans. on Computer Systems*, 16:133–169, 1998.

[5] Roberto De Prisco and Butler Lampson. Revisiting the Paxos algorithm. In *Proc. of 11th Intl. Workshop on Distributed Algorithms (WDAG 97)*, pages 111–125. Springer-Verlag, 1997.

[6] Mikio Yoshida, Takeshi Okuda, Yuuichi Teranishi, Kaname Harumoto, and Shinji Shimojo. PIAX: A P2P Platform for Integration of Multi-overlay and Distributed Agent Mechanisms. *IPSJ Journal*, 49(1):402–413, 2008 (in Japanese).

[7] Alfonso Fuggetta, Gian Pietro Picco, and Giovanni Vigna. Understanding code mobility. *IEEE Trans. on Software Engineering*, 24:342–361, 1998.

[8] Jose Ortega-Ruiz, Torsen Curdt, and Joan Ametller-Esquerra. Continuation-based mobile agent migration. ⟨http://hacks-galore.org/jao/spasm.pdf⟩. (23-Jan-2009).

[9] Anurag Acharya, M. Ranganathan, and Joel Saltz. Sumatra: A language for resource-aware mobile programs. In *Mobile Object Systems Towards the Programmable Internet*, pages 111–130. Springer-Verlag, 1997.

[10] Wenzhang Zhu, Cho li Wang, and Francis C. M. Lau. JESSICA2: A distributed Java virtual machine with transparent thread migration support. In *IEEE 4th Intl. Conf. on Cluster Computing*, 2002.

[11] Takahiro Sakamoto, Tatsurou Sekiguchi, and Akinori Yonezawa. Byte-code transformation for portable thread migration in Java. In *Joint Symposium on Agent Systems and Applications/Mobile Agents*, pages 16–28, 2000.

[12] Arjav J. Chakravarti, Xiaojin Wang, Jason O. Hallstrom, and Gerald Baumgartner. Implementation of strong mobility for multi-threaded agents in Java. In *Proc. of Intl. Conf. on Parallel Processing*, pages 321–330. IEEE Computer Society, 2003.

[13] Apache Project. Javaflow. ⟨http://commons.apache.org/sandbox/javaflow/⟩. (23-Jan-2009).

[14] Lásaro Jonas Camargos, Rodrigo Malta Schmidt, and Fernando Pedone. Multicoordinated paxos. In *PODC'07: Proc. of the 26th annual ACM sympo. on Principles of distributed computing*, pages 316–317. ACM, 2007.

[15] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *3rd Sympo. on Operating Systems Design and Implementation (OSDI)*. USENIX Assoc., Co-sponsored by IEEE TCOS and ACM SIGOPS, 1999.

[16] Thorsten Schütt, Florian Schintke, and Alexander Reinefeld. Scalaris: reliable transactional p2p key/value store. In *ERLANG '08: Proc. of 7th ACM SIGPLAN workshop on ERLANG*, pages 41–48. ACM, 2008.

[17] Mike Burrows. The chubby lock service for loosely-coupled distributed systems. In *OSDI '06: Proc. of 7th sympo. on Operating systems design and implementation*, pages 335–350. USENIX Assoc., 2006.

[18] Tushar D. Chandra, Robert Griesemer, and Joshua Redstone. Paxos made live: an engineering perspective. In *PODC'07: Proc. of 26th annual ACM symposium on Principles of distributed computing*, pages 398–407. ACM Press, 2007.

[19] R. K. Nath. Fault tolerance of the application manager in Vigne. Master's thesis, University of Tennessee, 2008. Internship Report.

[20] Antony I. T. Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. *IFIP/ACM Intl. Conf. on Distributed Systems Platforms (Middleware)*, 2218:329–350, 2001.

[21] Sergio Mena, André Schiper, and Pawel Wojciechowski. A step towards a new generation of group communication systems. In *Proc. of the ACM/IFIP/USENIX 2003 Intl. Conf. on Middleware*, pages 414–432. Springer-Verlag, 2003.

Resilient P2P Streaming

Majed Alhaisoni, Mohammed Ghanbari
School of Computer Science and Electronic Engineering
University of Essex
Colchester, UK
malhai@essex.ac.uk
ghan@essex.ac.uk

Antonio Liotta
Electrical Engineering Department
& Mathematics and Computer Science Department
Technische Universiteit Eindhoven
The Netherlands
a.liotta@tue.nl

Abstract— **P2P streaming has shown an alternative way of broadcasting media to the end users. It is theoretically more scalable than its client-server based counterpart but suffers from other issues arising from the dynamic nature of the system. This is built on top of the internet by forming an overlay network. End-users (peers) are the main sources of the overlay network, sharing their bandwidth, storage and memory. Peers join and leave freely, which dramatically affects, both on QoS and QoE. Furthermore, the inter-connections among the peers are based on logical overlays, which are not harmonized with the physical underlay infrastructure. This article presents combinations of different techniques, namely stream redundancy, multi-source streaming and locality-awareness (network efficiency), in the context of live and video-on-demand broadcasting. A new technique is introduced to improve P2P performance and assess it via a comparative, simulation-based study. It is found that redundancy affects network utilization only marginally if traffic is kept at the edges via localization techniques; multi-source streaming improves throughput, delay, and minimizing the streaming time.**

Keywords- P2P; Multimedia; Redundancy; Multi-source; Locality-awareness

I. INTRODUCTION

Peer-to-Peer (P2P) streaming has evolved into one of the most popular Internet applications. P2P entails a highly attractive paradigm in a distributed fashion. It provides simple and efficient mechanisms to pool and share redeemable resources like CPU cycles, disk space or multimedia files. These advantages tolerate that any peer can join and leave without resulting in ill effect on the stream continuity and indexing, in contrast to the traditional client/server concept where a failure of the central server may affect the stream strongly. A P2P mode of operation, however, also has some downsides. P2P systems cause high traffic volumes, including extra traffic due to redundancy as well as signalling traffic to maintain the overlay topology. P2P network topologies reveal a high variability and P2P traffic patterns of P2P applications fluctuate strongly in time and space.

Overlay networks form the basis of the P2P networking applications where peers connect logically to form the virtual overlay. Peers can join and leave the overlay freely. In a P2P streaming application, multimedia contents are delivered to a large group of distributed users with low delay, high quality and high robustness [2]. P2P-based versions of IPTV, Video on Demand (VoD) and conferencing are, thus, becoming popular.

P2P streaming systems can sustain many hosts, possibly in excess of hundreds or even millions, with miscellaneous heterogeneity in bandwidth, capability, storage, network and mobility. An additional aim is to deliver the stream even under dynamic user churn, frequent host failures, unpredictable user behaviours, network traffic and congestion. To accomplish these goals, it is essential to address various challenges to achieve effective content delivery mechanisms, including routing and transport support.

This article is primarily directed towards finding effective ways to increase the resilience and scalability of the overlay network whilst at the same time minimizing the impact on the physical network (or underlay). It is found that P2P frameworks mostly fail to address the network locality which tends to cause severe network operational and management issues. In turn, this limits P2P scalability when traffic streams traverse, and thus congest, large portions of the network. Another issue is that existing P2P streaming systems are intrinsically best-effort. This fact, combined with their network unfriendly behaviour, often leads network operators to impair P2P traffic, with detrimental consequences for the resulting quality of service.

The key question we are addressing herein is whether and how it would be possible to increase the user quality of experience (QoE) in P2P streaming. Common techniques were developed, such as redundant streaming, i.e. to send multiple streams to the same user in order to reduce packet loss. The downside is that redundancy increases traffic, thus reducing network utilization and, hence, increasing congestion.

In order to retain the benefits of redundancy (QoE) and reduce its detrimental effects on the network (congestion and QoS degradation), in [1] we have studied the combination of two techniques, redundant-streaming and network locality. We found that by keeping traffic local among the peers and mainly at the edges of the network, the benefits of redundant-streaming outweigh their shortcomings.

Herein, we extend [1] considerably, including further evidence and carrying out an additional comparative analysis with another approach. We introduce multi-source streaming based on disjointness and without any type of redundancy. It is found that this approach further improves throughput, decrease delay and packet loss, and minimizes the overall streaming time. Both techniques are combined with network locality and computational load balancing, benchmarking against P2P TV system Joost.

Initial results indicate that both the multi-source and the redundant-streaming approaches lead to significant improvements, suggesting a promising direction in P2P streaming.

II. RELATED WORK

As we are dealing with network QoS, QoE, P2P locality awareness, stream redundancy and multi-source streaming we give first an overview of different studies that have looked at these topics individually. To the best of our knowledge, a comparative evaluation assessing combined strategies, as we do in this paper, has not been published before.

Ways to pursue efficiency between overlay and underlay have started to be investigated only recently. Authors in [3] propose a technique, where the peers on the overlay are chosen based on their mutual physical proximity, in order to keep traffic as localized as possible. A similar approach is described in [4], where they measure the latency distance between the nodes and appropriate Internet servers called landmarks. A rough estimation of awareness among the nodes is obtained to cluster them altogether, as in [5] [6].

On the other hand, another study in [7] proposes different techniques where the video stream is divided into different flows that are transmitted separately to increase parallelism and, hence, reduce transmission latency. The authors use the PSQA technique that gives an estimate of the quality perceived by the user. This study was concerned on how to influence and improve on quality (as measured by PSQA). They introduce three cases: sending a single stream between nodes; sending two duplicate streams via different paths; and sending two disjoint sub-streams whose union recreates the original one. In our work we look at the case of multiple redundant streams and multi-source streaming, looking at the effects that redundant streams have on both the network load and the user QoE. Also we see how multi-source streaming alleviates some of the shortcomings of redundant streams. However, we emphasize on techniques to choose the inter-communicating peers based on their mutual proximity, to keep traffic local and minimize the impact on the network load.

Overlay locality is also studied by [8], where the authors make use of network-layer information (e.g. low latency, low number of hops and high bandwidth). We use though a different distance metric, based on RTT (round trip time) estimations, to prioritize overlay transmissions. Additionally, we use a cluster management algorithm whereby intercommunicating peers are forced to periodically handover, in order to distribute computational load as well as network efficiency (as explained in [13] and [14]).

Hefeeda et al [10] have proposed a mechanism for P2P media streaming using Collectcast. Their work was based on downloading from different peers. They compare topology-aware and end-to-end selection based approaches.

The latter approach is also the subject of [9], which employs a simpler version of our RTT approach based on continuous pinging of peers. Similarly, we adopt clustering to limit the signalling overheads associated with this process and prevent bottlenecks.

Other studies such as [11], propose relevant methods to serve multiple clients based on utility functions or clustering. A dynamic overlay capable of operating over large physical networks is presented in [12]. In particular, they show how to maximize the throughput in divisible load applications.

Moreover, Thomas *et al* [15] proposed a distributed hash table, which is suitable for highly dynamic environments. Their work was designed to maintain fast lookup in terms of low delay and number of routing hops. In their work, the hop-count was the main metric used to determine locality-awareness. According to their work, neighbouring nodes are grouped together to form a clique. Nodes share the same ID in a clique; moreover, the data will be replicated on all the nodes on the clique to avoid data loss.

A clique has an upper and lower bound in terms of the number of nodes, such that cliques are forced to merge or split. Another aspect of their work is to assume that all the nodes are distributed uniformly in a two-dimensional Euclidean space. However, this may not work in a large network such as the Internet. The link structure is updated periodically in order to establish a structured network. On the other hand, their proposal is based on pining nodes to join the closet clique which will drastically introduce extra signalling overhead.

Another study similar to [15] was conducted by Shah Asaduzzaman *et al* [16]. Their proposal was built on top of [22] with some modifications, introducing stable nodes (super-nodes) and replicating the data among the stable nodes only. However, their proposal elects one or more stable nodes of highest available bandwidth in each cluster and assigns a special relaying role to them.

Their work is based on a combination of tree and mesh architectures where the nodes on the clique form a mesh and the stable nodes are connected in a tree structure. For each channel, a tree is formed between the stable nodes including only one stable node in each clique. However, stable nodes are elected based on their live session. So, in this case a clique may have more than a stable node. The downside of this approach is that the relaying nodes (super nodes) are forming a tree, so reconstructing them in case of failures and peers churn will be costly and can introduce some latency.

By contrast to the abovementioned two works, our proposal aims not only to retain the benefits of redundancy (QoE) but also to reduce its detrimental side effects on the network. We study the combination of two techniques, redundant-streaming and network locality, while on [15] and [16] they are mainly concerned with network locality. We prioritize the choice of sources based on their mutual distance from the destinations. In essence we adopt a previously published hierarchical RTT monitoring approach [17] to maintain a list of sources $\{S_i\}$, ranking their order based on their distances from the recipient (R). Periodically, a new set of sources $\{S_{inew}\}$ is chosen from this pot and handover is forced from $\{S_i\}$ sources to $\{S_{inew}\}$ sources.

On the other hand, effective streaming mechanisms make use of the multi-path nature of P2P networks to satisfy the bandwidth requirements of media applications by using network resources. In [18], authors establish a generic framework for multi-path streaming. Different advantages gained by the exploitation of multiple transmission paths for media broadcasting consist of accumulative network bandwidth and delay reduction. Another experimental work on multi-path streaming was conducted in [19], which offer some insights concerning the selection of content sources and streaming paths, based on the jointness/disjointness of network segments.

However these findings cannot be applied directly in P2P scenarios, especially due to the lack of coordination among the peers. Hence, our paper introduces a multiple-source (stream) from different peers and clusters. In this regard, receiver makes the selection of those peers according to the algorithm published in [14]. GnuStream [25] uses multiple senders to stream a video to the receiver. GnuStream is, however, not robust to the churn of peers. This problem is improved in [26] by introduction of a central power peer responsible for sender selection and switching when such occasion happens. In real-life scenarios, the assumption of an always-available central power node cannot be justified. In [27], the authors proposed an algorithm where the receiver has control of rate allocation and packet partitioning among the senders.

In this paper we aim to establish to point up to which we can increase redundancy without triggering network congestion and, also, what is the optimum number of peers needed for transmission in case of redundancy and multi-transmission sources. Our work not only introduces locality-awareness, redundant-stream and multi-source but also it load balances computing and network resources. We introduce a new way of calculating the packet loss ratio, and end-to-end delay, with the stream redundancy.

Furthermore, QoS and QoE are tested by transmitting a video to examine the scalability and resilience of this proposition.

Looking at previous studies, we can say that our main contributions are:

1) To study a new combination of existing techniques (cross-layer optimization, localization, forced handovers, redundant-stream, and multi-source-stream);
2) To take the perspective of the network operator, in trying to harmonize overlay and underlay networks;
3) To look for trade-offs between redundancy levels (to increase QoE) and network efficiency;
4) To make a comparison of two well know techniques redundant-stream, and multi-source-transmission accompanied by locality awareness and computational load distribution;
5) Introducing new techniques to gauge packet loss and end-to-end delay;
6) To compare against randomized approaches. (*mimicking the Joost application*).

III. PROPOSED APPROACHES

In this section two techniques are proposed and compared to each other. These techniques are: 1) redundant-streaming, whereby the stream redundancy is increasing from 1 to 5 sources (*more details will be given in the following section*); 2) and multi-source streaming, based on disjointness, whereby the stream is divided into flows and transmitted from 1 to 5 sources out of different clusters. Both techniques are combined with locality-awareness and computational load balancing. The emphasis of this paper is to examine the performance and compare each technique to a Joost-like system [20]. In essence, Joost chooses sources randomly and continuously handovers among sources to pursue computational load distribution.

A. Redundant- streaming approach

In this scenario the number of redundant streams is increased from 1 to 5 as shown in Figure 1. Sources {S1…S5} are chosen based on locality and are also continuously (periodically) forced to handover, choosing new sources from a pot of available sources. These are prioritized based on mutual inter-peer distances to ensure that traffic is kept as local as possible. On the other hand, forced handovers ensure that the important feature of computational load-balancing is maintained. This special aspect was discussed in previous publications [13] [14]. More details as to how network locality and computational load balancing are maintained from the receiver side can be found in [14].

Instead, herein we are mainly interested in understanding whether location-aware P2P techniques can actually reduce the detrimental effects of P2P redundant streaming. Under architectures other than P2P (unicast, multiple unicast and multicast), redundant streams increase QoE but have the side effect of increasing network congestion. An interesting proposition is that of finding the minimum redundancy level, which leads to the maximum QoE improvement. By contrast if we adopt a P2P approach that succeeds in keeping traffic away from the core network, we have a good chance that redundancy does not always result into network congestion. Our aim is to verify this hypothesis and better understand its implications, comparing it to localized multi-source based on disjointness but without introducing any type of redundancy.

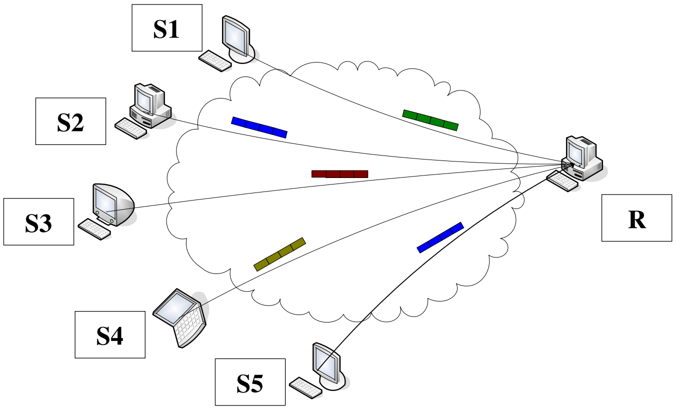


Figure 1 - Proposed architecture

Redundancy study

In order to study the effect of redundancy in relation to both QoS and QoE, we first measure relevant parameters (as detailed in section IV) for a Joost-like system [20] and use this as a benchmark. Redundancy is increased from 0 (1 source per destination) to 4 (5 sources per destination).

We then compare this with our proposed approach, in which sources are forced to handover continuously (as in Joost) - to ensure computational load balancing – but are not chosen randomly.

We prioritize the selection of sources based on their mutual distance from the destination. In essence we adopt a previously published hierarchical RTT (round trip time) monitoring approach [17] to maintain a list of sources $\{S_i\}$, ranked based on their distance from the recipient (*R*). Periodically, a new set of sources $\{S_i^{new}\}$ is chosen from this pot and handover is forced from $\{S_i\}$ sources to $\{S_i^{new}\}$ sources. Our hypothesis is that this forced handover strategy does not impact network congestion if traffic is kept away from the core network. We wish to establish, however, until which point we can increase redundancy without triggering network congestion. Details of how the peers are clustered and the inter-connecting and switching among the peers are maintained can be found in earlier publication [14].

B. Multi-source approach

Based on the derived findings of the redundant-stream results, it was found that locality-awareness and computational load balancing play a vital role in the scope of redundant-source-streaming. In the initial approach, it was shown how redundant-aware streaming has a positive impact over the quality of the received video. Now another scenario is proposed to improve the resilience of streaming without resorting to redundancy. This approach finds an effective way to improve the resilience and scalability of the overlay whilst at the same time not conflicting with the underlay network or overloading the network. Common techniques have been proposed to transmit the streams over multi-source or multiple paths. But a new combination of these two techniques (multi-source-path) complemented with locality-awareness and computational load balancing is examined here. This approach is compared to the former redundant-aware streaming.

Multi-source principle

Multi-sender methods are the best existing solutions for streaming video on P2P networks. However, in some cases multi-senders share a bottleneck and this impairs the throughput, increasing the delay and packet loss as well. In this method, multi-source streaming is combined with disjoint paths streaming, which ensure that peers don't share early bottleneck, which most likely happens over the access network. To illustrate and assess this method, the topology of figure 2 is used. Clique of peers are clustered together, giving the chance to alleviate the probability of having different peers over the same path (thus, creating a bottleneck or getting early congestion). In this method if a receiver R requests a certain video, a set of candidate senders (determined by the method in [14]) having the desired media are chosen from different clusters.

As in redundant-streaming, we prioritize the choice of sources based on their mutual distance from the destination. In essence we adopt a previously published hierarchical RTT monitoring approach [18] to maintain a list of sources $\{S_i\}$, ranked based on their distances from the recipient (*R*). Periodically, a new set of sources $\{S_i^{new}\}$ is chosen from this pot and handover is forced from $\{S_i\}$ sources to $\{S_i^{new}\}$ sources. Our hypothesis is that choosing senders from different clusters help in avoiding bottlenecks; furthermore, the forced handover strategy can be applied smoothly on the same cluster since peers in a clique share nearly the same RTT values [14]. This ensures that the proposed technique is applied efficiently in choosing the peers from different clusters and performing the enforced handover to maintain computational load balancing. The second part of the hypothesis is that transmitting the media file in this way will improve transmission time, increase throughput and decrease latency.

The video stream is divided into different flows which are sent from different sources, different clusters and over disjoint paths as shown in figure 1. This increases the parallelism and, hence, reduces transmission latency and also the transmitting time.

IV. Assessment Method

The effects of redundancy and multi-source streaming on QoS (Quality of Service) and QoE (Quality of Experience) are investigated, for each of the following two scenarios: 1) randomized scenarios (new sources are chosen randomly); and 2) locality-aware scenario (new sources are chosen based on minimal mutual distances from the recipient.

Sufficient Background traffic was added into the simulated network in order to simulate first lightly-congested networks and, then, heavily congested networks. Simulation scenarios, parameters and design are described below.

A. Simulation scenarios

Various simulation scenarios are considered. Each proposed scheme is benchmarked with a randomized behaviour. These scenarios are classified as follow:

- Localized-Redundant-Stream (L-R-S): in this scenario, redundant stream packets are streamed from 1 up to 5 sources, whereby every source is sending the same packets, to introduce redundancy. This technique is combined with locality-awareness and computational load balancing (*handover*).
- Randomized-Redundant-Stream (R-R-S): This scenario has the same characteristics of the L-R-S. The difference between the two is that L-R-S is locality-aware with load distribution mechanism, whereas R-R-S is purely based on random switchovers among peers (*this approach mimics a popular Video-On-Demand application known as Joost, used here as benchmarking for L-R-S*).
- Localized-Multi-Streaming (L-M-S): this approach is based on multi-source streaming but without creating any kind of redundancy. In this approach, packets are

scheduled and transmitted as flows from different sources starting from only 1 source and up to 5 sources. Packets are divided into flows and are scheduled to be transmitted from sources belonging to different clusters. This approach is combined with locality-awareness, computational load balancing and clustering. The most important factor besides multi-source streaming is clustering, which helps choosing the sources (peers) from different clusters to avoid the generation of hot spots.

- Randomized-Multi-Streaming (R-M-S): this approach is used as benchmarking for L-M-S. Connection and switching-over are built randomly though we transmit the stream from multiple-source (*This approach mimics a popular Video-On-Demand behaviour known as Joost, which is used here as benchmarking for L-M-S*).

B. Simulation setup

The proposed approaches were implemented and tested on the ns-2 network simulator (http://isi.edu/nsnam/ns/). A sample of the used topology is shown in figure 2. However, we have run our experiment with 200 nodes and found that the results are not changed owing to the nature of the proposed approach. In fact, the proposed approach always tries to keep the traffic at the edges of the network. Therefore, the traffic will not concentrate on particular part of the network although redundancy is introduced. The effect of redundancy is minimised as the traffic is kept at the edges and periodically connecting to new peers among the available peers with the aim of distributing the load. Furthermore, background traffic plays a vital role by running different sources and sinks over the network. Additionally, Although the used topology is fixed in the simulation, the proposed scenarios are treated in a different way. So, for the randomized scenarios, senders and receivers are chosen randomly for every run. On the localized scenarios, the senders are chosen based on locality and the receivers are selected randomly for every run. This gives the advantage of testing the L-R-S and L-M-R scenarios under different conditions over the used topology.

Moreover, various parameters were set on the used topology. First of all, each link has a bandwidth of 2 Mbps with equal latency (delay). However, the actual delay will be according to the nodes distance among each other; so, all the participants' peers have the same characteristics. IP is the network protocol and UDP is the transport protocol. For the redundant streaming simulation, video traffic of the "Paris" sequence of CIF resolution with 4:2:0 format was H.264/AVC coded and the same video packets were sent from one and, then, from multiple peers to the receiver.

On the other hand, for the multi-source video traffic simulation, the "Paris" video clips of CIF resolution with 4:2:0 format was H.264/AVC coded and the video flows were sent from one and multiple peers to the receiver. So, in this way, single video flows are transmitted from different peers according to a scheduling mechanism, whereby every flow has to go from different source to insure multiple sources and paths concurrently.

Secondly, in order to overload the network, it was imperative to set the CBR background traffic to vary the network load and enable us to study the various approaches under different loading conditions. The CBR traffic was setup from different sources to different destinations, with a 512 byte packet size. This background traffic operates during the whole duration of the simulations. This is added to the stream video on the running simulation.

For statistical significance, the proposed approaches were simulated independently and repeated 50 times. The presented results correspond to the average values of these simulations.

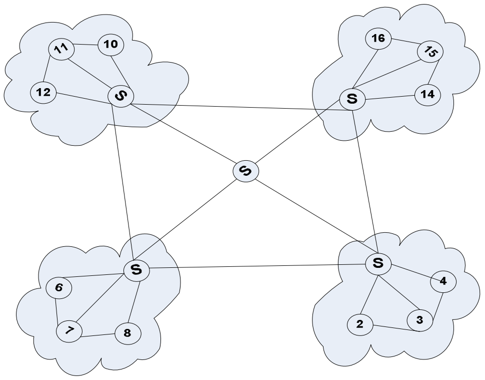


Figure 2 Simulation Topology

C. Evaluation metrics

Since we are dealing with two different approaches, redundant streaming and multiple-sources, it was important to devise suitable performance metrics. For the redundant stream, QoS factors such as packet loss and end-to-end delay is defined below in a new way. Next section shows how these metrics are defined in the scope of redundant streaming.

Throughput*:* is the average rate of successful delivery of the packets from the senders to the receiver. The throughput can be measured in different ways such as bit/s or packet/s.

Packet loss ratio*:* usually defined as the ratio of the dropped over the transmitted data packets. It gives an account of efficiency and of the ability of the network to discover routes. However, in P2P communication, a new way of calculating the packet loss ratio needs to be defined to study the particular issues relating to redundancy. In our case a packet is actually transmitted by several sources and is considered lost only if it is never received through any of the streams. This is formalized as follow:

Pi generic packet (i) sent by all source nodes
d_j sending or source nodes

$$X_{ij} = \begin{cases} 1 & \text{if } P_i \text{ sent by } d_j \text{ is lost} \\ 0 & P_i \text{ is received} \end{cases}$$

The decision as to whether a packet is lost or not will be according to the following Cartesian product:

$$PL_i = \prod_{j=1}^{d} x_{ij}$$

Therefore, if P is the total number of packets required to reconstruct a given stream, the packet loss ratio will be:

$$\mathbf{P}L = \frac{\sum_{i=1}^{p} PL_i}{P} \qquad \textbf{(1)}$$

Average end-to-end delay: is the average time span between transmission and arrival of data packets. In redundant P2P streams, the delay of each received packet is the minimum delay among all the received packets of the same type sent by all senders. This includes all possible delays introduced by the intermediate nodes for processing and querying of data. End-to-end delay has a detrimental effect on real-time IPTV.

Peak Signal to Noise Ratio: PSNR is an objective quality measure of the received video, taken as the user QoE. It is defined as the logarithm of the peak signal power over the mean squared difference between the original and the received captured video. This is formulized as in equation (2) and figure 3 shows how this is obtained through the simulation architecture.

$$\text{PSNR} = 10\log\frac{P^2}{E^2} \qquad \textbf{(2)}$$

Where p is the peak value for a given pixel resolution, e.g. for 8-bits $p = 255$

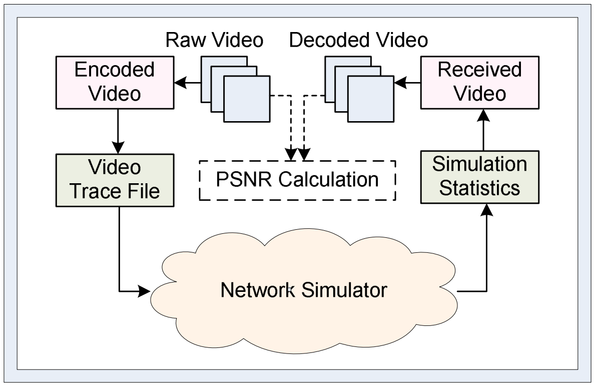


Figure 3 – Simulation Architecture

V. SIMULATION RESULTS

First of all, the redundant-streaming scenarios (L-R-S and R-R-S) will be compared to each other. Then, these results will be compared to the multi-source streaming scenarios (L-M-S and R-M-S).

A. Redundant-stream

Figures 4 and 5 show the packet loss ratio for the cases of lightly and heavily loaded networks, respectively. At light network load shown in Figure 4, both methods do not lead to any packet loss up to a redundancy level equal to 4. This means that for randomized redundant streaming (R-R-S), up to 4 sending peers, enough redundant packets can be received to compensate for any losses. However beyond 4, the added traffic creates congestion that leads to more losses, such that the backed up redundant packets may also be lost. With the localized-redundant-streaming (L-R-S), the figure shows that even going up to 5 redundant streams does not cause any packet loss. This is due to the fact that, no matter how much the network is congested, there is always enough number of redundant packets to be used by the receiver.

A network friendly behaviour of the L-R-S is even more apparent at higher network load, as shown in Figure 5. In this case, the network is brought close to congestion by the background traffic (not by the streams under scrutiny). The network is severely congested; then even one or two senders in action can lead to packet loss. By increasing the redundancy level (e.g., more senders), the packet loss is reduced in both methods. However, with the L-R-S, there is almost no packet loss after receiving from 3 senders. This is due to the fact that multiple copies of the same packets are now sent to the recipient.

By disparity, the randomized connection of Joost-like approach (R-R-S) cannot bring packet loss down to zero. In this case, even increasing the number of senders (more redundancy), the senders themselves create additional congestion such that, beyond 4 senders, congestion increases, as shown in figure 5.

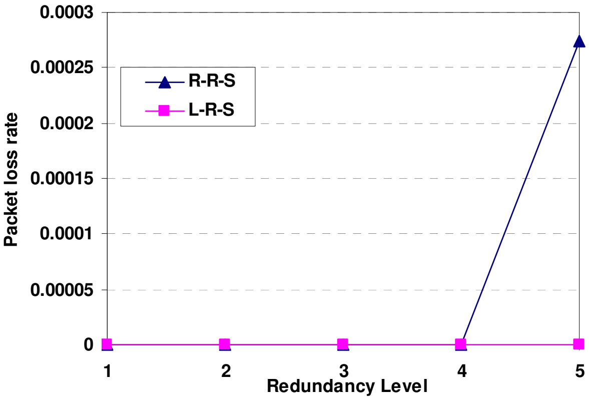


Figure 4 - Packets loss ratio – lightly loaded network

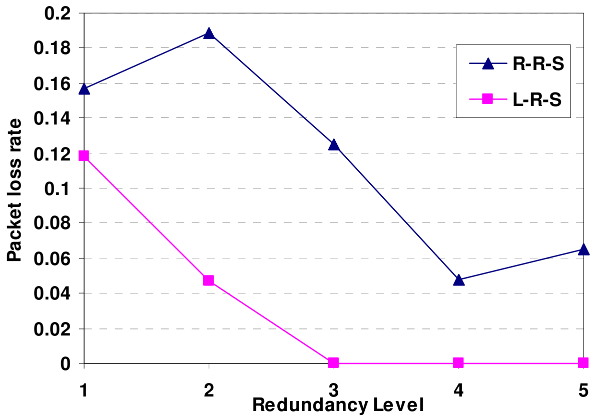


Figure 5 - Packets loss ratio – heavily loaded network

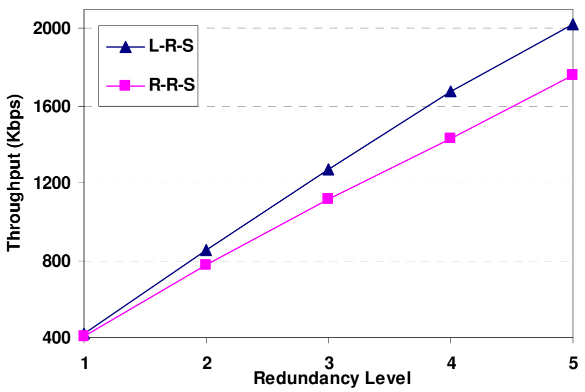


Figure 6 – Throughput

Figure 6 shows the effect of L-R-S and R-R-S the throughput. It can be noticed that the L-R-S, by managing the overlay, leads to considerable improvements. This approach reduces the average RTT among the inter-communicating nodes and, in turns, reduces the overall link utilization. In fact, the average throughput achieved with L-R-S is increasing with the number of sending peers.

Delay is another important quality of experience (QoE) parameter. Figures 7 and 8 show the average end-to-end network delay, for the lightly and heavily congested scenarios, respectively. It is important to note that at heavy network load, the end-to-end delay under localized connection has the least value at the redundancy level of 3 – 4 senders. Considering that packet loss rate is almost eradicated with just about 3 redundant senders (figure 5), it appears that the optimal redundancy level is comprised between 3 and 4.

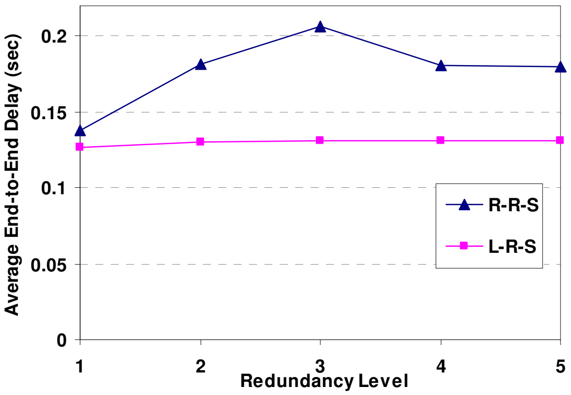


Figure 7 - Avg. E2E Delay – heavily loaded network

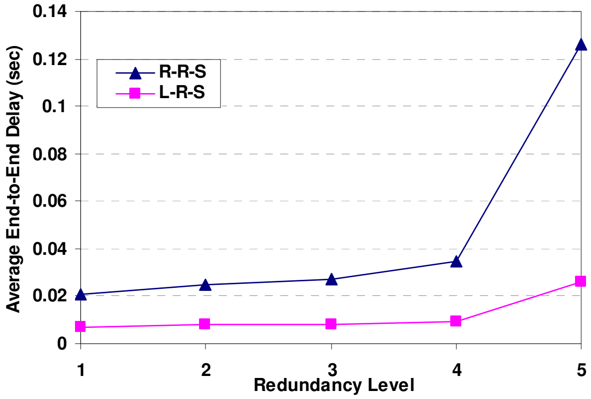


Figure 8 – Avg. E2E Delay – lightly loaded network

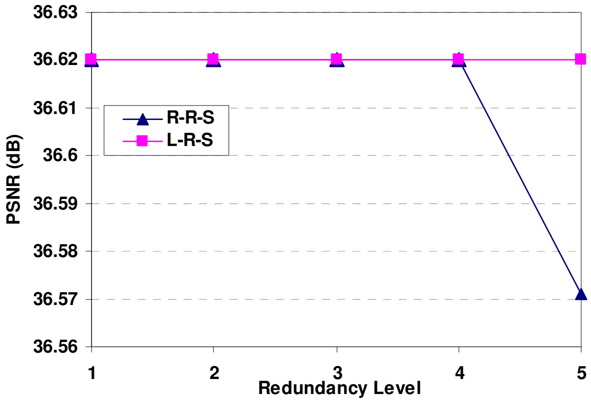


Figure 9 - PSNR – lightly loaded network

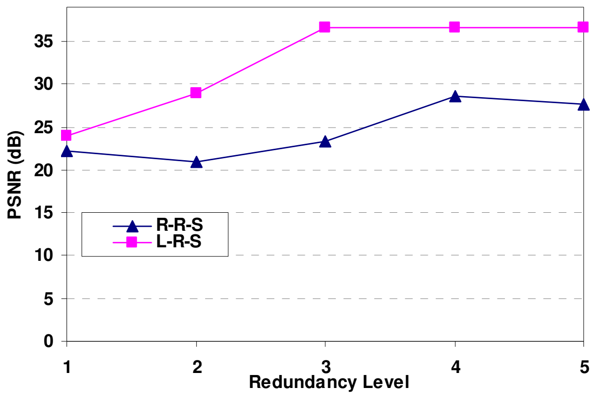


Figure 10 - PSNR – heavily loaded network

Finally, the subjective quality of the decoded video under both loading conditions is compared. The streams were coded and decoded with an H.264/AVC encoder/decoder of type JM15. Figures 9 and 10 show the objective video quality as measured by PSNR for lightly and heavily congested network, respectively. As expected they exhibit behaviour similar to that of Figures 4 and 5. When the network is lightly loaded, there is hardly any packet loss up to a redundancy of 4, at which point the randomized scenario generates congestions and, thus, PSNR drops.

The localized approach does not show any packet loss and maintains a constant level of QoE even with 5 injected redundant streams. Figure 10 is also consistent with this rationale. Noticeably, the localized approach improves PSNR steadily up to a redundancy level of 3, where the quality reaches its maximum theoretically achievable value (packet loss is zero at that point).

	Sending period (s)		*Throughput (Kbps)*	
Senders	**L-R-S**	**L-M-S**	**L-R-S**	**L-M-S**
1	35.59	35.59	424.16	409.98
2	35.59	17.88	850.78	829.69
3	35.59	12	1272.41	1243.39
4	35.59	9	1672.78	1645.92
5	35.59	7.32	2020.72	2041.21

Table 1 - Transmission Time with achieved Throughput

B. *Multi-source-streaming*

This section shows the results of multi-source-streaming compared to redundant-streaming (previous section). By examining the same parameters (*throughput, end-to-end delay, packet loss, and PSNR*), different pros and cons can be highlighted between the two approaches.

Table 1 shows the average data-rate over time when transmitting the Paris sequence over multiple sources and redundant-stream. Clearly, multi-streaming is able to achieve low sending time with high bandwidth. On the other hand, redundant-streaming is increasing the throughput but without reducing the sending period. Mean results were taken over 50 simulation runs to provide statistically significant results.

Figure 11 indicates the achieved throughput by every scenario; in the two proposed scenarios (L-M-S and L-R-S) the throughput is almost the same. However, L-M-S can be preferred over L-R-S since multi-streaming does not generate any redundancy load to the network and still achieving high throughput with the increase in number of sources. On the other hand, the randomized scenarios (R-M-S and R-R-S) are achieving less throughout for both cases and this is due to the pure randomness in selecting the sources, which leads to congesting different paths towards the receiving node.

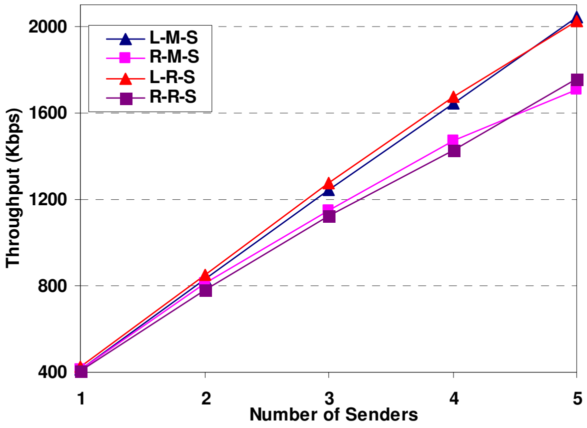


Figure 11 - Throughput

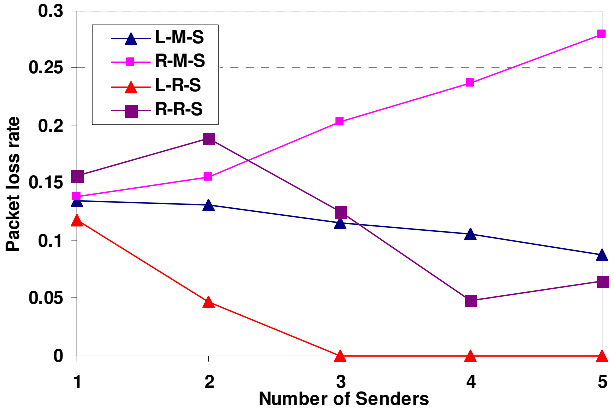


Figure 12 - Packet loss

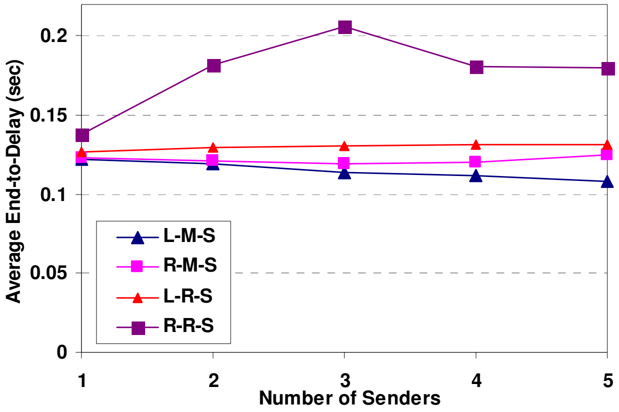


Figure 13 - End-to-End Delay

Looking now at packet loss, as shown in figure 12, it can be seen that L-R-S is performing the best due to the offered redundant-streaming by many sources. On one hand, this is positive but, on the other hand, this may at some point incur heavy load onto the network. L-M-S is not eradicating packet loss but with the increase in number of sources, packet loss is decreasing. However, the maximum occurred packet loss ratio is around 13%, which according to table 2, is acceptable as a threshold of QoE acceptability.

Another factor that is very important for streaming the video over P2P is end-to-end delay. This parameter plays a vital role with regards to the playback deadline mainly for real-time streaming. Any packet experiencing transmission latency will affect the video quality and, thus, the user QoE. In real time applications, packets missing their playback deadline are discarded and, hence, can be counted as lost packets. All these factors will diminish the continuity indexing for the video streaming.

Figure 13 depicts the end-to-end delay for the video packets. Looking at L-M-S and L-R-S, it can be seen that delay is slightly increasing with the increase in redundancy level. Whereas on the multi-source streaming the transmission latency is decreasing smoothly with the increase in number of transmission sources. In this regard, the value of dividing the streams into flows and transmitting that from different peers out of various clusters are obvious. On the other hand, for the randomized scenarios, in every case, it is clear that the most affected approach is the randomized scenarios with redundancy, which can be considered the worst.

Finally, the objective quality of the decoded video of all the scenarios is compared in Figure 14. The streams were coded and decoded with an H.264/AVC encoder/decoder of type JM15. Figure 14 shows the objective video quality as measured by PSNR for all the examined approaches. As expected, they exhibit behaviour similar to the packet loss findings of Figure 12. However, still the localized-redundant-streaming is performing the best. At the level of 3 redundant streams, it improves the packet loss ratio and, thus, the PSNR as received by the end-consumer.

Packet loss ratio [%]	QoE acceptability [%]	Video quality playback
0	84	Smooth
14	61	Brief interruptions
18	41	Frequent interruptions
25	31	Intolerable interruptions
31	0	Stream breaks

Table 2 - Quality of experience acceptability thresholds

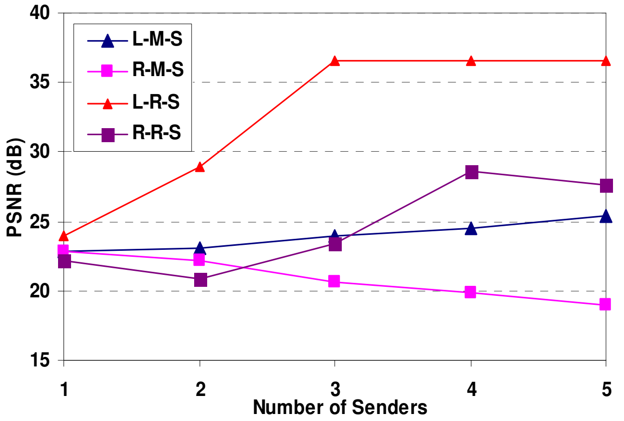


Figure 14 - PSNR

On the other hand, it is clear that the multi-source streaming is starting by an acceptable PSNR, at around 23 dB; but it keeps increasing with the increasing of dividing the stream into flows and transmitting it from more sources. This still reflects the value of dividing the video into flows and transmitting it over multiple sources and from different clusters

VI. DISCUSSION

In this paper, a localized redundant-source is proposed to offer a redundancy of the same content over the network with high quality and low end-to-end delay. This proposal has been tested and run under different scenarios and network conditions. In order to quantify its robustness, it has first been run under a point-to-point connection where there is only one sender and one receiver. Then, multi-connections were introduced where the receiver can connect from 2 and up to 5 senders (peers) simultaneously. Redundant streams are used in combination with locality-awareness to assess our initial hypothesis.

Varieties of connections have been run in two extreme congestion levels, lightly and heavily congestion, respectively. Different effective parameters on the network have been measured to show and validate how robust and practical is the proposed localized scheme with the offered redundancy by the chosen sending peers.

However, in order to adjudicate on the goodness of the localized-redundant-stream, a benchmark of a popular VoD application [20] is compared against this proposal, to show how localized-redundant-streaming behaves in contrast to the randomized scheme.

A vital network parameter is packets loss; to gauge this factor, a new way of measuring packet loss has been defined mainly to quantify the performance of the proposed localized redundant-stream, as shown in equation 1. The presented results show that the localized scheme is performing better across all QoS and QoE parameters. Consequently, by looking at packets loss ratio, it is apparent from figures 4 and 5 that the localized approach is better, particularly in case of the heavily congestion network. This was mainly due to the combination of location awareness with stream redundancy.

End-to-end delay is almost consistent on both congestion levels and particularly from 2 to 4 senders, as shown in figures 7 and 8. This was taken as the minimum delay among all the received packets. On the other hand, QoE is maintained appropriately on the localized-redundant-stream. Figures 9 and 10 provide evidence of the perceived video quality to the end-consumers. However, the most divergence can be seen between the two compared schemes on figure 10, where the network is heavily congested.

Another interesting point is the finding of the required optimum number of peers that should serve a client. According to the presented results by the localized-redundant -stream, it is so clear from all the figures that 3 to 4 peers are good enough to provide high quality to the end-users within the current configuration. In contrast to that, with the randomized approach, it is difficult to give a precise indication for that, as the inter-connections among peers are unpredictable.

We also considered localized-multi-streaming dividing the stream into flows and transmitting from different sources (1S-→5S). This was also complemented by locality-awareness and computational load distribution. An additional feature in this approach is de-clustering. In order to avoid hot spots, we made sure that higher weight was given to the peer selection process to those peers belonging to different clusters.

This was compared to the former localized-redundant-streaming over the most factors affecting the QoS and QoE. First and foremost, the achieved throughput in multi-source streaming is competing with redundant-streaming (Figure 11). Throughput is increasing since bandwidth is higher when there are multiple sources and each peer sends only part of the stream, as in L-R-S.

Looking now at packet loss, it was obvious that L-R-S is better than L-M-S. However, multi-source streaming is achieving an acceptable level of packet loss ratio starting from 13% and then decreasing till 8%, which is good according to the QoE acceptability threshold of table 2. On the other hand, from the network load perspective, multi-source streaming, with this percentage of packet loss ratio, may be preferable over redundant-streaming due to the traffic overheads incurred by redundant-streaming.

These two scenarios were also examined over the packets end-to-end delay. It was noticed that L-M-S reduces the transmission latency with the increase in number of sources. Packet loss may not affect as much as end-to-end delay since the video playback deadline is more stringent in this regard.

Another benefit of L-M-S is that streaming time is decreasing with the number of sources sending the stream. This is vital in case of short connectivity or coverage, such as mobile cellular networks.

Lastly, the perceived quality PSNR by the end-user is obtained by the decoder. However, packet loss plays a crucial role in this parameter since it is proportional to the deduction of packet loss. Therefore, redundant-streaming is showing higher PSNR than multi-source streaming.

The proposed two techniques have shown noticeable improvements over ordinary P2P streaming. From the users' perspective, minimizing network traffic is not as important as achieving a smooth QoE. In this case the localized-redundant-streaming approach (L-R-S) will be the favourable choice. On the other hand, from the network operators' perspective, multi-source streaming seems to be a better choice.

VII. Concluding Remarks

A large variety of popular applications, including VoD, live TV and video conferencing, make use of P2P streaming frameworks. These have emerged from the fundamental principles of insulation and abstraction between the network and the application layers. With this regard, several studies published recently (e.g. [21]), including also some by the authors of this article (e.g. [14] [22] [23]), have identified that when the P2P overlay is designed in isolation from the underlay physical network, the P2P stream has detrimental effects on the network itself. To aim for scalability and user QoE, P2P solutions adopt redundancy, multi-stream, caching, statistical handovers and other similar techniques, which generate substantial network management and control problems to the network operator.

This problem motivates our work aimed at studying ways to maintain the QoE and scalability of the overlay, whilst reducing its detrimental effects onto the underlay. This article represents our initial attempt to pursue network-friendly P2P streaming. Our initial hypothesis that the combination of network locality, redundant-stream, and multi-streaming can lead to significant improvements is reinforced by the findings presented herein.

The difficulty in realizing this approach in practical systems is that it entails breaking the concept of network insulation from the application. In our current work we plan to further study the potential of other cross-layer optimization techniques, which opens the way towards stimulating research. Once the overlay is made aware of the underlay, or vice versa, the potential of other techniques can be unleashed. For instance, we are studying the use of machine learning for the purpose of correlating network and application conditions and building a network-friendly overlay network.

REFERENCES

[1] M. Alhaisoni, A. Liotta, M. Ghanbari, "Multiple Streaming at the Network Edge", In Proc. Of the First International Conference on Advances in P2P Systems, and published by IEEE Computer Society, October 11-16, 2009 - Sliema, Malta.

[2] Yong Liu, Yang Guo, Chao Liang, "A survey on peer-to-peer video streaming systems". The Journal of P2P Networking and Applications, Springer, Volume 1, Number 1 / March, 2008, pp. 18-28

[3] Y. Liu, X. Liu, L. Xiao, L. M. Ni, and X. Zhang, "Location-aware topology matching in P2P systems", in Proc. IEEE Infocom, pp. 2220-2230, 2004.

[4] Z. Xu, C. Tang, and Z. Zhang, "Building topology-aware overlays using globalsoft-state", in 23rd International Conference on Distributed Computing Systems, (ICDCS), RI, USA, 2003, pp. 500-508

[5] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast", Computer Communications Review, vol. 32, pp. 205-217, 2002.

[6] B. Zhao, A. Joseph, and J. Kubiatowicz, "Locality aware mechanisms for large-scale networks", in in Proc. Workshop Future Directions Distrib. Comput. (FuDiCo), Italy, 2002, pp. 80–83.

[7] Pablo RODRÍGUEZ-BOCCA (2008), "Quality-centric design of Peer-to-Peer systems for live-video broadcasting", PhD theses.

[8] T. P. Nguyen and A. Zakhor. "Distributed video streaming over Internet", In Proc. SPIE, Multimedia Computing and Networking, volume 4673, pages 186–195, December 2001.

[9] Mushtaq, M., Ahmed, T. "Adaptive Packet Video Streaming over P2P Networking Using Active measurements", In ISCC 2006, Proceeding of the 11th IEEE Symposium on Computers and Communications, pp. 423-428, IEEE Computer Society, Los Alamitos.

[10] Mohammed Hefeeda et al, "Promise: Peer-to-Peer Media Streaming Using Collectcast", ACM MM'3, Berkely, CA,November 2003, pp. 45-54

[11] Mohamed M. Hefeeda and Bharat K. Bhargava, "On-Demand Media Streaming Over the Internet", in The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems, 2003, p. 279.

[12] Kovendhan Ponnavaikko et al, "Overlay Network Management for Scheduling Tasks on the Grid", In ICDCIT, Springer, 2007, pp. 166-171.

[13] M. Alhaisoni, A. Liotta, M. Ghanbari, "An assessment of Self-Managed P2P Streaming", In Proc. Of ICAS2009 (the 6th International Conference on Autonomic and Autonomous Systems and published by IEEE Computer Society, 21-24 April 2009 Valencia, Spain.

[14] M. Alhaisoni, A. Liotta, M. Ghanbari, "Resource awareness and trade off optimization in P2P Video Streaming", accepted in International Journal of Advance Media Communication, special issue on High-Quality Multimedia Streaming in P2P Environments.

[15] Thomas L., Stefan S., and Roger W. (2006), "eQuus: A Provably and Locality-Aware Peer-to-Peer System", In Proc. Of the Sixth IEEE International Conference on Peer-to-Peer Computing , On page(s): 3-11.

[16] Shah A., Ying Q., and Gregor B. (2008), "CliqueStream: An efficient and Fault-resilient Live Streaming Network on a Clustered Peer-to-Peer Overlay", In Proc. Of the Eighth IEEE International Conference on Peer-to-Peer Computing, On page(s): 269-278.

[17] Ragusa C, Liotta A, Pavlou G. "An adaptive clustering approach for the management of dynamic systems". IEEE Journal on Selected Areas in Communications, 2005; 23(12) 2223–2235

[18] L. Golubchik, J. Lui, T. Tung, A. Chow, and W. Lee, "Multi-path continuous media streaming: What are the benefits?" *ACM Journal of Performance Evaluation*, vol. 49, no. 1-4, pp. 429–449, Sept 2002.

[19] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee, "On multiple descriptionstreaming with content delivery networks," in Proceedings of IEEEINFOCOM, vol. 3, 23-27 June 2002, pp. 1736–1745.

[20] www.Joost.com

[21] H.Kolbe, O.Kettig, E. Golic, "Monitoring the Impact of P2P Users on a Broadband Operator's Network", Proc of IM'09, NY 1-5 June 2009. IEEE Press.

[22] N.N. Qadri, A. Liotta, "Effective Video Streaming Using Mesh P2P with MDC over MANETS", Journal of Mobile Multimedia, Vol. 5 (3), Rinton Press, 2009.

[23] A. Liotta, L. Lin, "The Operator's Response to P2P service demand", IEEE Communications Magazine, special issue on the IP Multimedia Subsystem. Vol. 45 (7), pp.76-83, IEEE, July 2007.

[24] Agboma, F., Smy, M. and Liotta, A. (2008) "QoE analysis of a peer to-peer television system". In Proceedings of IADISInt. Conf. on Telecommunications, Networks and Systems, pp.365-382.

[25] X. Jiang, Y. Dong, B. Bhargava, "GNUSTREAM: a P2P media streaming prototype", In Proceedings of ICME'03 2 (2003) 325–328.

[26] Y. Guo, K. Suh, J. Kurose, D. Towsley, "Peer-to-peer on demand streaming service and its performance evaluation", In Proceedings of ICME'03 2 (2003) 649–652.

[27] T. Nguyen and A. Zakhor, "Distributed video streaming over internet," in *Proc. Multimedia Computing and Networking*, San Jose, CA, Jan. 2002, SPIE, vol. 4673, pp. 186–195.

Increasing Energy Efficiency in Mobile Peer Networks by Exploiting Traffic Sampling Techniques

Julian K. Buhagiar
Dept of Communications and Computer Engineering
University of Malta
Msida, MSD 2080, MALTA
julian.buhagiar@ieee.org

Carl J. Debono
Dept of Communications and Computer Engineering
University of Malta
Msida, MSD 2080, MALTA
c.debono@ieee.org

***Abstract*—Wireless infrastructures have seen a drastic increase in energy requirements as technology has shifted towards higher allocated frequencies in an attempt to provide more bandwidth to accommodate more users and provide the resources for more demanding applications and services, such as video streaming and multimedia applications. Driven by the increase in demand from an always increasing subscriber base, large cities are also forcing wireless service operators to install more base stations and access points to sustain an adequate quality of service. Mobile peer networking offers a possible solution for the later with the added benefit of providing power-efficient communication, since transmission over short distances demands less transmission power using appropriate peer connectivity algorithms [1]. Thus, substantial energy can be saved, given that the power amplifier that lies in the transmitter circuitry is the most power hungry device of each mobile node. A novel algorithm based on peer nodal hierarchies, traffic mapping, and neural networks is proposed. This algorithm invokes the use of a traffic sampling matrix to optimise delivery and routing of peer node information, allowing for more efficient power distribution. Additional optimisation is provided through a state-switching algorithm that exploits the traffic sampling algorithm to switch to a more energy-efficient state algorithm when residual neighbouring resources are available. Results show that this technique presents a remarkable power efficiency improvement over standard peer-to-peer networks.**

Keywords-energy saving, mobile peer networking, power consumption, state-switching, traffic sampling

I. INTRODUCTION

The ever increasing cost of energy is pushing network operators and designers to develop solutions for more power-efficient mobile networks. With all the different wireless technologies deployed today, mobile peer-to-peer (MP2P) networks offer a possible solution to curb energy consumption by reducing the transmission distance between nodes and hence the power requirements. MP2P networks can also provide efficient distribution and delivery of high-bandwidth services such as media downloading, making them a more attractive solution. Current mobile peer network implementations are based on well established fixed-network topologies, implying that existing peer protocols are overlaid on the existing mobile networks, thus inheriting the unique challenges present in the mobile environment, such as bandwidth asymmetry and node availability [2].

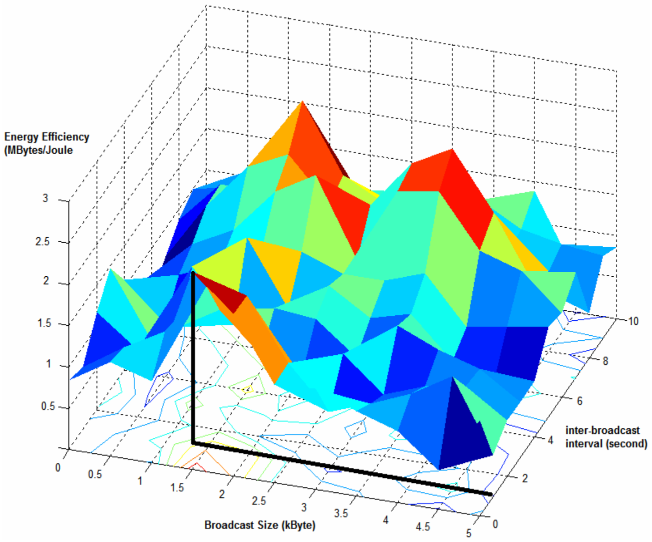


Figure 1. Energy Efficiency (in MB/Joule) for a MP2P system in node discovery mode. Assuming a mobile peer density λ of 6.25x10^{-4}m^{2}, for a mobile inter-broadcast interval of 1s, optimal energy efficiency is attained with a broadcast size of 1.35kB.

Inefficient use of this overlay network causes significant mobile power dissipation as the mobile peer clients enter/exit the network sporadically, and asynchronously connect with the network seeds. Optimal MP2P energy efficiency can only be derived through iterative calculations for each peer node. These involve parameter calculations such as broadcast size and inter-broadcast intervals, resulting in energy efficiency curves such as the one shown in Figure 1 [3]. Such algorithms are themselves computationally intensive, leading to a trade-off between energy efficiency and computational speed [4]. Moreover, the optimal solution changes with time as the wireless channel characteristics change and the nodes move in the network. Therefore, a solution that adapts itself with time is essential.

A promising solution is to apply enhanced traffic sampling algorithms that optimise the delivery and routing of the peer client information, thereby contributing to more efficient power dissipation. This paper presents a novel approach that exploits traffic sampling to optimise the energy profile in MP2P networks. Successful implementation of this technology can be deployed in mobile peer networking scenarios where efficient peer node resource information routing is required, such as TCP congestion avoidance [2].

The proposed peer network solution also provides rapid information delivery based on node routing and traffic history techniques. A neural network-based algorithm is used to optimise the node routing within the peer network from traffic reports generated by the mobile clients within the network. The algorithm sets the routing paths based on the traffic requests and the busy hour traffic allocations. History is used as a starting point for the optimisation process since traffic profiles tend to be statistically similar across weeks for similar traffic payloads. Through this information, the algorithm guarantees the most efficient resource allocation on the mobile radio interface, thereby yielding an optimal power profile.

Additional energy optimisation is provided through efficient use of the observed traffic reports to influence the state-switching of each peer node. By periodic measurement of the neighbouring nodes, the band state of each mobile node can be changed to a more efficient state, depending on the amount of residual resources the node has.

The rest of the paper is organised as follows: Section II gives an overview of peer networking. Section III describes the algorithm developed followed by testing and results in Section IV. Finally, Section V provides some comments and conclusions.

II. Mobile p2p Energy Consumption

A. *Mobile P2P Environment Challenges*

Mobile peer networking is different from the traditional wired peer-to-peer networks in the following ways:

i) Availability – the mobile user moves within the network and can therefore be out of range, switched off, or under a profile which discourages peer networking (example: roaming, non-flat rate data tariff).

ii) Connectivity – the mobile user's ping times can vary significantly throughout the networking session as the system switches to a slower packet access network to maintain the service or the current location area is an inefficiently subnet, resulting in long hop times.

iii) Topology – the peer nodes are physically mobile, and sometimes can be highly mobile [6]. The traffic density can vary significantly and consequently the underlying communication network is subject to frequent topology changes and disconnections. P2P network approaches that require pre-defined data access structures such as search routing tables [5] and spanning trees [6] are impractical in the wireless environment.

iv) Throughput – The communication throughput between two peers is constrained by the wireless bandwidth available, the channel contention, and the limited connection time. Therefore, selective communication is necessary such that the most important data is communicated.

v) Cooperation – Like many other P2P systems or mobile ad-hoc networks, the ultimate success of mobile P2P networks relies heavily on the cooperation among users. In infrastructure P2P systems, incentive is provided for peers to participate as suppliers of data, compute cycles, knowledge/expertise, and other resources. In mobile ad-hoc networks, incentive is provided for mobile hosts to participate as intermediaries/routers. In mobile P2P networks, the incentive has to be provided for participation as both suppliers and intermediaries (namely brokers) [7].

B. *Peer Node Reporting*

In modern peer networks, such as BitTorrent [8] there are two main methods for node reporting to/from peer nodes, namely report pulling and report pushing. Report pulling implies that when a mobile peer makes an explicit request, the peer network is flooded with queries and the specific report is retrieved from the mobile peers that contain the specific report data for that particular node. Report pulling is widely used in resource discovery, route discovery in mobile ad hoc networks, and file discovery by query flooding in wired P2P networks like Gnutella [9]. Flooding in a wireless network is in fact relatively efficient compared to wired networks because of the advantages derived using wireless multicast [10].

The alternative approach for peer reporting is through pushing. In this method, reports are flooded, and consumed by peers whose query is answered by received reports. To date mechanisms exist to broadcast information in:

(i) the entire peer network

(ii) in a specific geographic area (geocast)

(iii) to any one specific mobile node (unicast/mobile ad-hoc routing)

(iv) any one arbitrary node (anycast).

Additionally, report pushing methods can be further divided into stateful methods and stateless methods. Most stateful methods are topology-based, i.e. they impose a structure of links in the network, and maintain states by data dissemination.

One specific group of stateful methods contains cluster- or hierarchy-based methods [11], in which moving peers are grouped into some clusters or hierarchies and the cluster heads are randomly selected. Reports are disseminated through the network in a cluster or hierarchy manner, which means that reports are first disseminated to every cluster head and each cluster head then broadcasts the reports to the member peers in its group.

Although cluster- or hierarchy-based methods can minimise the energy dissipation in moving peers, these methods will fail or dissipate more energy in highly mobile environments as they have to maintain a hierarchy structure and frequently need to reselect the cluster heads. Each time cluster heads are elected, energy is lost in the overhead needed in passing necessary information and in the computation of the clustering algorithm.

Another stateful algorithm utilises Location-based methods. In this method, each moving peer knows the location of itself and its neighbours through some localisation technique, such as GPS or Atomic Multilateration [12].

The simplest location-based data dissemination is Greedy Forwarding, in which each moving peer transmits a report to a neighbour that is closer to the destination than itself.

However, Greedy Forwarding can fail in some cases, such as when a report is stuck in local minima, which means that the report stays in a mobile peer whose neighbours are all further from the destination. Therefore, some recovery strategies are required, such as GPSR (Greedy Perimeter Stateless Routing [13]). Other location-based methods, such as GAF (Geographic Adaptive Fidelity [14]) and GEAR (Geographical and Energy Aware Routing [15]), take advantage of knowledge about both location and energy to disseminate information and resources more efficiently.

In stateless methods, the most basic and simplest solution is through flooding techniques, such as [16]. In flooding-based methods, mobile peers simply propagate received reports to all neighbouring mobile peers until the destination or a maximum number of hops is reached. Each report is propagated as soon as is received. Flooding-based methods have many advantages, such as no state maintenance, no route discovery, and easy deployment. However, they inherently cannot overcome several problems, such as implosion, overlap, and resource blindness. Therefore, other stateless methods are proposed, such as gossiping-based methods and negotiation-based methods.

Gossiping-based methods, such as [4], improve flooding-based methods by transmitting received reports to a randomly selected neighbour or to the neighbours that are interested in the particular content. The advantages of these methods include reducing the implosion and lowering the system overhead. However, the cost of determining the particular interests of each moving peer can be huge and transmitting reports to a randomly selected neighbour can still cause the implosion problem and waste peers' memory, bandwidth and energy. Furthermore, dissemination, and thus performance, is reduced compared to pure flooding.

Negotiation-based methods solve the implosion and overlap problem by transmitting first the id's of reports; the reports themselves are transmitted only when requested [17]. Thus, some extra data transmission is involved, which costs more memory, bandwidth, and energy. In addition, in negotiation-based methods, moving peers have to generate meta-data or a signature for every report so that negotiation can be carried out, which will increase the system overhead and decrease the efficiency.

C. *Energy Consumption Model*

Before participating in peer node reporting, each MP2P client specifies the energy constraint using the algorithm: "*from time(t) until time(H) the MP2P system is allowed to use fraction F of the remaining energy*". The allocated energy covers all the energy consumed by report dissemination, including the energy used for transmission, receiving, listening, and computation. If F is the energy allocation fraction, and Ω Joules of energy is left in the node, this constraint is translated into the following specification: "*The algorithm may use no more than Ω·F Joules until time H*".

The pair ($\Omega{\cdot}F$, H) is thus the energy budget. Therefore, the lifetime demand of each individual device is accommodated [27].

Now we introduce the energy consumption model. Let the size of a message be M bytes excluding the MAC header. According to [26], the energy consumed for transmitting a message can be described using the linear equation:

$$E_n = f * M + g \qquad (1)$$

Intuitively, there is a fixed component associated with the network interface state changes and channel acquisition overhead, and an incremental component which is the size of the message. Experimental results confirm the accuracy of the linear model and are used to determine values for the linear coefficients g and f. For 802.11 broadcast systems, g=266×10^{-6} Joule, and f=5.27×10^{-6} Joule/byte ([27]).

D. *Optimal Transmission Size*

Consider a broadcast of M bytes by a mobile device x. If another neighbour of destination node y transmits during some time slot of a broadcast, then a collision will occur, and the whole broadcast is considered corrupt (i.e. unsuccessfully received) at y. If N is the number of neighbours that successfully receive the message from node x, the throughput of the broadcast sent by x, denoted T_h, is defined as:

$$T_h = M * N \qquad (2)$$

Intuitively, the throughput is the total amount of data successfully received by the neighbours of x. If E_n is the energy consumed at the network interface of x for sending the broadcast message, the energy efficiency of the broadcast by x, denoted P_{eff}, is defined as:

$$P_{eff} = \frac{T_h}{E_n} \qquad (3)$$

In other words, the energy efficiency is the throughput produced by each unit of transmission energy consumed at node x. To compute the value of T_h, consider a mobile device x broadcasting a message at an arbitrary time slot. According to [26], the expected value of T_h can be approximated by:

$$E(Th) \approx 2\pi\lambda M \int_0^r \delta(1-p')^{\lambda r^2(2q(\frac{\delta}{2r})+(\pi-2q(\frac{d}{2r})).(2T+1))-1} d\delta \qquad (4)$$

where:

λ is the number of devices per unit in area
r is the transmission range of each device in (m)
b is the data transmission speed in (bits per second)
p' is the probability that a device starts a broadcast
τ is the length of the MAC timeslot in (s)
h is the size of the MAC header in bytes
c is the time since completion of last broadcast (s)

and:

$$T = \tau(M + h).8 / b$$

$$q(a) = \arccos(a) - a\sqrt{1 - a^2}$$

Equation (4) takes into account the effect of hidden terminals as well as direct collisions. By the definition of energy efficiency, the expected value of the energy efficiency is:

$$E(P_{eff}) \approx 2\pi\lambda M \frac{\int_0^r \delta(1-p')^{\lambda r^2 (2q(\frac{\delta}{2r})+(\pi-2q(\frac{d}{2r})).(2T+1))-1} d\delta}{f(M+g)} \quad (5)$$

From equation (5), if τ, p', λ, h, b, r, f and g are fixed, then the energy efficiency P_{eff} as a function of the broadcast size M is a bell shaped curve. Thus, there is a value of M that maximises the energy efficiency, i.e. achieves the best trade off between the channel utilisation and broadcast reliability.

For the rest of this subsection we show that indeed, except for M, all the parameters of equation (5) can be determined by the mobile device. The parameters τ, h, r, b depend on the network, and are fixed for a given communication network technology. For example, h is equal to 47 in the wireless access protocol 802.11b [2]. f and g depend on the network interface hardware and can be calibrated as demonstrated by [25]. The density λ can be estimated based on the average number of neighbours over time (recall that each mobile device knows its neighbours via the neighbour discovery protocol), given the transmission range.

The probability p' is determined as follows. Let c be the number of seconds since the completion of the last broadcast of x until the time when the current broadcast size is to be determined. If, on average, a mobile device starts a broadcast of data every c seconds, then its probability of starting a broadcast in each medium access time slot is τ / c. Therefore, the broadcast probability p' is substituted in equation (5) by τ / c. For instance, if c = 5 seconds and τ = 20µs, then

$$p' = (20\times10^{-6})/5 = 4\times10^{-6}$$

Using $p' = \tau / c$, a formula is derived for the throughput in which the only unknown parameter is M. In some scenarios the actual p' may be lower than τ / c. This occurs because there is a delay from the time when the broadcast is triggered until the channel is sensed free and when the broadcast is actually started. In other words, the actual broadcast period may be larger than c. However, since the difference between p and p' is small, this delay is expected to be small as well and therefore can be neglected. Thus, the optimal value for M can be found, i.e. the value of M for which P_{eff} is maximised. This value is denoted by $M_{optimal}$, and can be outlined in Figure 1.

E. Proposed Solution

One method of determining P_{eff} of a broadcast is by measuring the energy efficiency, which can be defined as:

$$E_n = \frac{\sum(D_n)}{T} \quad (6)$$

where D is the amount of mobile peer data that is correctly received and T is the unit of peer transmission energy. The data throughput received by a mobile peer node n can be expressed as a product of the neighbours that successfully receive the broadcast (N), and the size of the broadcast b:

$$T_h = Nb \quad (7)$$

Consequently the power efficiency can be expressed as shown in equation (3). Alternatively P_{eff} can be expressed as the trade-off between the inter-broadcast interval t_b and the broadcast size b [3], as can be seen in Figure 1.

The main challenge to maximise P_{eff} is to determine which nodes are more connectable than others, and the relative connection times associated with each. This requires the construction of a reliability algorithm which builds up a table of the client nodes within the peer network with the following criteria:

(i) *availability a* in % over time window t_1,
(ii) *connectivity c* in ms over time window t_2, and
(iii) *transfer rate r* in kbps over time window t_3.

As this table is updated periodically, a list of the nodes whose performance indicators are most favourable over time t_1, t_2, and t_3 will gradually be upgraded to super-nodes. The challenge is to provide an efficient scalable traffic analysis algorithm that is susceptible to rapid variations in the mobile environment.

III. Traffic Sampling

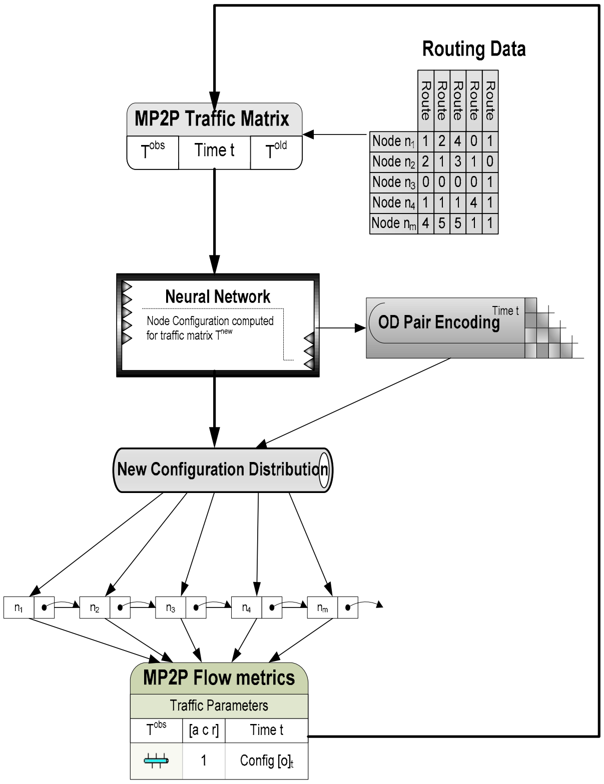
A. Traffic Analysis

Determining the metrics a, c and r over time t gives a reliability indication which helps build a traffic profile for each node. The observed traffic for each node i can be defined as:

$$T_i^{obs} = [a_i \quad c_i \quad r_i] \quad (8)$$

Most wireless network operators specify their network-wide goals in terms of Origin-Destination (OD) pairs [9]. To achieve flow monitoring goals which are specified in terms of OD-pairs, the optimisation engine needs the traffic matrix and the routing information, both of which are readily available in the network [18]. The required matrix is obtained through the traffic sampling architecture outlined in Figure 2.

To handle the traffic dynamics the following heuristic approach is used in this work. Assume that the sampling manifests for every 5-minute interval for the Fri. 9am-10am period of the current week is required. To avoid over fitting, the hourly traffic matrix for the previous week's Fri. 9am-10am period is employed after being divided by 12 (to obtain the required 5-minute interval). The resulting traffic matrix T^{old} is used as input data to compute the manifests for the first 5-minute period. At the end of this period, the flow of data from each node is collected, and the traffic matrix T^{obs} from the collected flow reports is obtained. If the fractional coverage for OD_i with the current sampling strategy is C_i and x_i sampled flows are reported, then $T_i^{obs} = x_i/C_i$. That is, the number of sampled flows are normalized by the total flow sampling rate.



	Route	Route	Route	Route	Route
Node n_1	1	2	4	0	1
Node n_2	2	1	3	1	0
Node n_3	0	0	0	0	1
Node n_4	1	1	1	4	1
Node n_m	4	5	5	1	1

Figure 2. Schematic layout of the traffic sampling architecture. The traffic matrix compiled from the flow reports acts as an input vector into the neural network architecture, which computes the optimal configuration for each node, distributed to each respective node. New traffic measurements are then generated, collected and re-parsed into the traffic matrix for re-analysis.

Given the observed traffic matrix for the current measurement period T^{obs} and the old traffic matrix obtained from the week before T^{old}, a new traffic matrix can be computed using a conservative update policy. The resulting traffic matrix T^{new} is then used as the input to the algorithm that obtains the manifests for the next 5-minute period. Thus, the following conservative update algorithm was designed, as outlined in Algorithm 1. This algorithm determines the differences between the new and the existing traffic reports, and computes a new traffic matrix entry if the difference between the values exceeds a specific threshold. The residual resources are measured and allocated to the respective nodes according to the resource utilisation of the peer nodes reporting the traffic matrix.

Algorithm 1: P2P Traffic Sampling Algorithm

```
FUNCTION {P2P_Traffic_Matrix_Update ($T_i^{obs}$, $T_i^{old}$, $\Delta$)}
//check for significant differences between $T^{obs}$ and $T^{old}$
$\delta_i = |T_i^{obs} - T_i^{old}|/T_i^{old}$ // $\delta_i$ = estimation error for ODi
if $\delta_i > \Delta$ then
// compute new traffic matrix entry
    $T_i^{new} = [0]$
else
    $T_i^{new} = T_i^{old}$
end if
if $T_i^{obs} \geq T_i^{old}$ then
    $T_i^{new} = T_i^{obs}$
end if
if $T_i^{obs} < T_i^{old}$ then
// check resource utilisation for all routers monitoring $OD_i$
    for all $x_i \in x$ do
        $C_i$ = resource utilisation of $x_i$
            // result: 1=free, 0=busy
    end for
    if $\Sigma C = i$ then
    // all nodes have residual resources available
            $T_i^{new} = T_i^{obs}$
    else
    // not enough residual resources available - revert to previous matrix
            $T_i^{new} = T_i^{old}$
    end if
end if
return($T_i^{new}$)
```

The challenge in deploying this solution is limited by the ability to process this information effectively for each node, and to detect when the information is new or no longer valid [19]. A neural network algorithm can provide the necessary framework to process and determine the optimal nodal information. A neural network which uses a back-propagation architecture promises to be effective in this application since training is done during each iteration by the traffic data that is emanating from each node in the network.

B. Back-propagation Neural Networks

Back-propagation neural networks are created by the generalisation of the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions [17]. A set of training vectors and their corresponding target vectors are used to train the neural network until it is capable of approximating a function, associate input vectors with specific output vectors, or classify each input vector in an appropriate way as defined by the user. Any function with a finite number of discontinuities can be approximated by networks having biases, a sigmoid layer, and a linear output layer.

It has been shown that back-propagation networks tend to give reasonably good classification results when they are presented with input vectors that they have never seen [20]. This generalisation property makes it possible to train the network on just a representative set of the input/target pairs and get reasonably good results without training the network on all possible input/output pairs.

The simplest implementation of the back-propagation learning algorithm updates the weights of the network and its biases in the direction in which the performance function decreases most rapidly. This is represented by the negative of the gradient of the function. A single iteration of the algorithm is given by:

$$x_{k+1} = x_k + \alpha_k g_k \tag{9}$$

where x_k is a vector of current weights and biases, g_k is the current gradient, and α_k is the learning rate.

In most neural network training algorithms the learning rate α_k is used to determine the length of the weight update, and is known as the step-size [23]. There are two different ways in which this gradient descent algorithm can be implemented: incremental mode and batch mode. In incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. On the other hand, in batch mode all the input vectors are applied to the network before the weights are updated. Both solutions eventually converge and can be adopted.

However, most gradient descent algorithms are often too slow for practical real-time problems [24]. The basic algorithm adjusts the weights in the steepest descent direction. It turns out that, although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. In the conjugate gradient algorithms a search is performed along conjugate directions, which generally produces faster convergence than steepest descent directions [25].

Most of the conjugate gradient algorithms have the step-size adjusted after each iteration. A search is made along the conjugate gradient direction to determine the best step-size that minimises the performance function along that line. The conjugate gradient algorithm used starts by searching in the steepest descent direction (g) on the first iteration of the observed traffic input vector T^{obs}, thus:

$$T^{obs}{}_0 = - g_0 \tag{10}$$

A line search is then performed to determine the optimal distance to move along the current search direction. Therefore, combining (9) and (10) yields:

$$x_{k+1} = x_k + \alpha_\mathrm{k} T^{obs}{}_k \tag{11}$$

The next search direction is determined such that it is conjugate to the previous search directions. The general procedure for determining the new search direction is to combine the new steepest descent direction with the previous search direction as:

$$T^{obs}{}_k = -g_k + \beta_k T^{obs}{}_{k\text{-}1} \tag{12}$$

where β_k is a constant.

The various versions of the conjugate gradient algorithm are distinguished by the way in which β_k is computed. One solution is to use the Fletcher-Reeves algorithm which updates this constant using:

$$\beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \tag{13}$$

This is the ratio of the norm squared of the current gradient to the norm squared of the previous gradient [20]. The conjugate gradient algorithms require only a little more storage than the simpler gradient decent ones. Thus, these algorithms work well in neural networks that utilise a large number of weights.

The conjugate gradient algorithms are usually much faster than variable learning rate back-propagation, although the results vary from one problem to another [24]. Furthermore, the computational intensity required is comparable to the standard algorithms, making them ideal for implementation in power-efficient environments such as MP2P clients [1].

A tuned neural network algorithm is essential in determining the optimal peer nodal information to process node routing and thus obtain accurate traffic reports in T^{obs}. Consequently, from equation (13), any delays in reporting and processing by the back-propagation algorithm may result in estimation errors in T^{obs} and consequently δ_i. As the back-propagation network may incur significant delays in training, when subject to a large input dataset [20], additional ongoing efforts are underway to investigate optimisation methods, such as the use of Self-Organising Maps (SOMs) neural architecture in training.

C. State-switching optimisation

Additional optimisation to the traffic sampling algorithm is obtained through the efficient use of the observed traffic matrix T^{obs} to influence the state-switching of each peer node [19]. By periodic measurement of the neighbouring nodes, the band state of each mobile node can be changed for $node_i$ to a more efficient state depending on the amount of residual resources.

Algorithm 2: State Switching Algorithm

```
FUNCTION {Check_load(node_i, $t_i^{current}$, $t_i^{old}$)}
//check time window hysteresis expiry $t^{exp}$
$\delta_i = |t_i^{current} - t_i^{old}|$ // $\delta_i$ = time elapsed from last measurement for node_i
if $\delta_i > t^{exp}$ then
    //check resource utilisation for all neighbouring nodes x monitoring node_i
    for all $x_i \in x$ do
        $C_i$ = resource utilisation of $x_i$
        // result: 1=free, 0=busy
    end for
    if ($\Sigma C = i$) then
    //all neighbouring nodes have residual resources, reduce band
        Reduce_Band(node_i)
    else
    if ($\Sigma C < i$ and $\Sigma C > y$) then
    //at least y neighbouring nodes have residual resources, keep existing band
        $F_i^{new} = F_i^{old}$
    else
    // no residual resources available, increase band
        Increase_Band(node_i)
    end if
end if
return($F_i^{new}$)

FUNCTION {Reduce_Band(node_i)}
    //Query node band state energy table
    $F_i^{old}$=lookup(node_i, phone_energy_table)
    //Subtract one band energy state
    $F_i^{new} = F_i^{old} - 1$
return($F_i^{new}$)

FUNCTION {Increase_Band(node_i)}
    //Query node band state energy table
    $F_i^{old}$=lookup(node_i, phone_energy_table)
    //Add one band energy state
    $F_i^{new} = F_i^{old} + 1$
return($F_i^{new}$)
```

Figure 3 outlines the allowed state transitions while Algorithm 2 provides a high-level description of the algorithm. It is important to note that state-switching algorithms can only be supported where the mobile client operating system (OS) supports active band selection through the use of APIs, such as provided by Symbian [21] and Android [22]. Otherwise, the client MP2P software may resort to the use of the traffic sampling algorithm only [28].

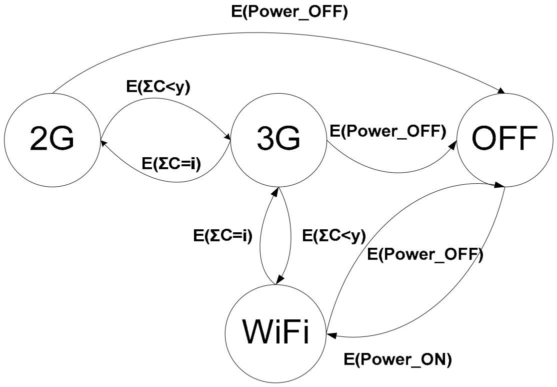


Figure 3. Allowed state transitions by the band switching algorithm, from [OFF] to [WiFi] and intermediate bands. E() represents an event and indicates the cause of the transition. Wherever $y < \Sigma C < i$, the state is transition-less.

IV. TESTING

In order to determine the energy dissipation across the different algorithms a mobile peer network was first designed and simulated using MATLAB®. The mobile peer network was designed using algorithms derived from the BitTorrent technology [8].

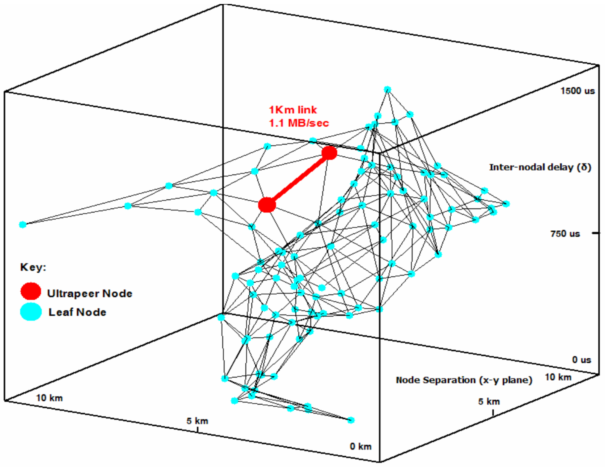


Figure 4: Mobile peer network simulation structure, showing node separation in the XY plane, and uTP inter-nodal delay (δ) in the Z plane. The two ultrapeer nodes are connected by a UMTS Release 5 wireless link with HSPA, operating at 1.4MBps (nominal rate 1.1MB/s).

The traffic sampling algorithm, utilising the back-propagation neural architecture consisting of a two-layer network, having 11 neurons in the hidden layer and 2 neurons in the top layer, and utilising a tan-sigmoid and a linear transfer function for the hidden and top layer respectively, was implemented.

Stochastic models of traffic sampling periods correlated with peer node power consumption were analysed to determine the optimum observation model for traffic sampling. High-frequency windows (1-4 min/sample) produce more frequently updated peer nodes, however this comes at a significant detriment to power consumption due to frequent polling. Conversely more conservative windows (6-15 min/sample) are more energy efficient, but they are impractical to intercept mobile peer clients with high mobility and/or cell state transitions. A 5-minute window was selected as the best trade-off between power-consumption and node data validity. Thus, the traffic sampling algorithm was built using the 5-minute observation model, and the resulting traffic matrix was used as input to the neural network.

The testing process consisted of a series of UDP connections, and high-level file transfers between two peer mobile nodes, with a 1km separation running on an emulation software [23]. These were connected across uTP, or the UDP-based implementation of the BitTorrent protocol [8], through the UMTS RNC as shown in Figure 4. The access network topologies were simulated using two-tiered wireless propagation algorithms [2]. Background packet traffic from the underlying UMTS network was incorporated to increase the network loading and investigate the proposed algorithm's energy profile under various traffic scenarios.

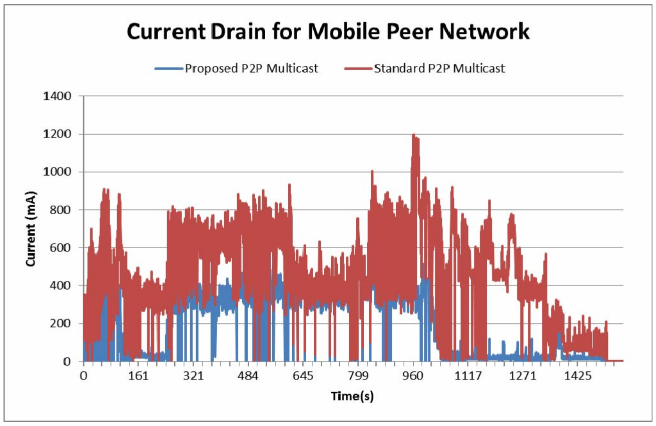


Figure 5(a). Current profile for mobile peer connectivity in mA, showing the difference in current drain between the standard mobile BitTorrent algorithm and the proposed traffic-sampled Mobile BitTorrent method.

The two peer nodes were configured to transmit (and receive respectively) a file having a size of 1.024 GB across the peer network over the uTP layer, whilst all other leaf nodes were configured as source-sink pairs to simulate the aggregate traffic on the network, generated according to a Poisson distribution. All queuing effects were simulated on the ultra-peer link using the FIFO queuing algorithm [25]. The network link capacity was set to 1.4Mbps and the round-trip propagation delay for the two ultra peer nodes (δ) was set to 1000μs^{-1}. The simulation window was 1500 seconds with a sampling time of 10 samples/second. The entire tests were affected over a period of 7 days with hourly measurements during the UMTS network busy hour. After the observation period expired, the results were collated with a confidence interval of at least 95% across all tests.

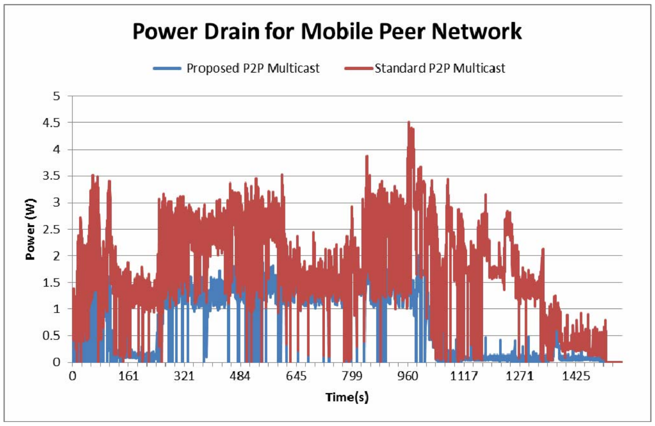


Figure 5(b). Power profile for mobile peer connectivity in W, showing the difference in power dissipation between the standard mobile BitTorrent algorithm and the proposed traffic-sampled Mobile BitTorrent algorithm.

In order to measure the effective mobile CPU load and resulting current and voltage dissipation, the mobile components were emulated with resource load monitoring capabilities to gauge the effective portion of processing cycles devoted to peer traffic routing. The resulting profile was fed into the mobile emulation software to determine the resulting drain on current and voltage respectively [22-23].

Figures 5 (a) and (b) show the difference in current drain between the standard and traffic sampled mobile peer client algorithms. The resulting drain is approximately 50% of the nominal current used by the standard algorithm. The optimal traffic sampling also resulted in the file transfer completing in 960 seconds, well ahead of the 1505 seconds taken by the standard algorithm.

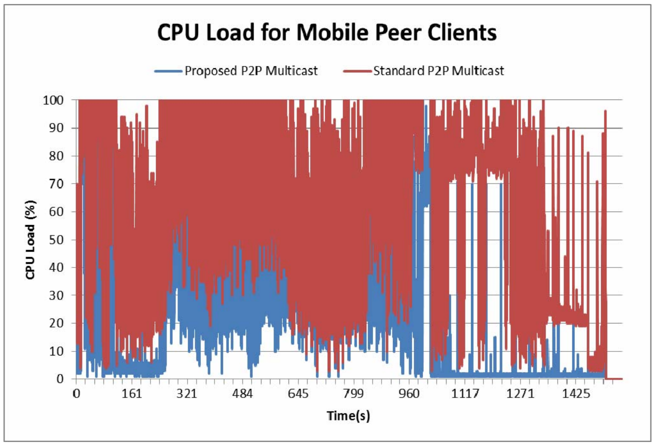


Figure 5(c). CPU load for the mobile peers, showing the difference in load between the standard mobile BitTorrent algorithm and the proposed traffic-sampled Mobile BitTorrent algorithm.

Figure 5(c) illustrates the discrepancy in CPU load for the proposed sampled BitTorrent client compared to the standard algorithms. The additional CPU cycles using the standard algorithm are the result of iterative routing of peer traffic through congested peer nodes, and/or redundant peer information, leading to a higher net CPU load on the mobile client and contributing to redundant routing information in each peer client that is discarded due to invalid or untimely data. Such situations occur when the routing information changes during data transmission between peer nodes, and is symptomatic of high latency or poor peer client localisation [2]. In this case, there is significant processing load efficiency using the traffic-sampled peer protocol, which is due to the rapid information delivered because of the lower network delay.

To further validate the performance improvements in a typical service deployment when compared to similar algorithms, a series of file transfer tests was performed across the network topologies. Table 1 illustrates the performance of the mobile simulator [22-23] for a number of 1GB file transfers across the peer networks using the Greedy Perimeter Stateless Routing (GPSR), Geographic Adaptive Fidelity (GAF) and Geographic and Energy Aware (GEAR) algorithms, when compared with the proposed Traffic Sampling (TS) technique. The performance exhibited by the TS model compares favourably, particularly in the file transfer times and processor loads, compared with the alternative methods presented in [12-15].

Although GPSR has a smaller memory footprint, the average RTT is still longer than the TS method because the rest of the node paths are unknown and have to be discovered. The TS algorithm deduces the link of each next-step based on the last measured performance of each node, contributing to a shorter RTT, which compares more favourably to other research efforts reported in [13] and [14], as can be seen from the results given in Table 1. This effectively reduces the CPU loading, thereby leading to a lower current and power drain on the power supply of the mobile peer client.

TABLE I. MOBILE PEER NETWORK PERFORMANCE

	P2P Key Performance Indicators (KPIs)				
	Test	***GPSR***	***GAF***	***GEAR***	***TS***
1	Memory Footprint (kB)	618	1125	1982	1238
2	File Transfer Time (s)	1142	1223	1890	960
3	Round-trip-time (RTT)	7s	9s	6s	2s
4	CPU Load (%)	32%	58%	89%	42%

A second series of tests was performed to determine the relative performance of the traffic sampling mechanism as a distribution of the physical peer location. As the peer network is mobile there is a significant variance with fixed peer networks, resulting in issues outlined in Section II. Consequentially a recursive test was performed across the 10km square test network with each peer node effecting a UDP burst pattern in alternate send/receive mode.

Fig. 6 shows the error *z(k)* as a function of location across the mobile peer network. Whilst the stability of the traffic sampling controller is consistent throughout the network, slightly higher errors are exhibited in areas where there are fewer peer nodes. This is due to the inverse relationship of the controller error with the background traffic and is similar to the performance shown in Fig. 6.

The Traffic Sampling mechanism displays significantly higher performance under loaded traffic conditions, and is thus suited for mobile peer networks, as these tend to exhibit higher instances of local congestion due to insufficient resources [2]. Judicious use of this algorithm enables the deployment of peer technologies in mobile networks without causing a packet traffic overload to the network, and therefore contributes to more efficient resource allocation in mobile peer networks.

An observational note is that the CPU load on the mobile client, and consequently the power drain, is highly correlated to the size of the peer algorithm memory footprint. One factor is due to the additional processor run-time loads involved in general memory management (pointer allocation, matrix operations, etc.) of the mobile operating system [21, 22].

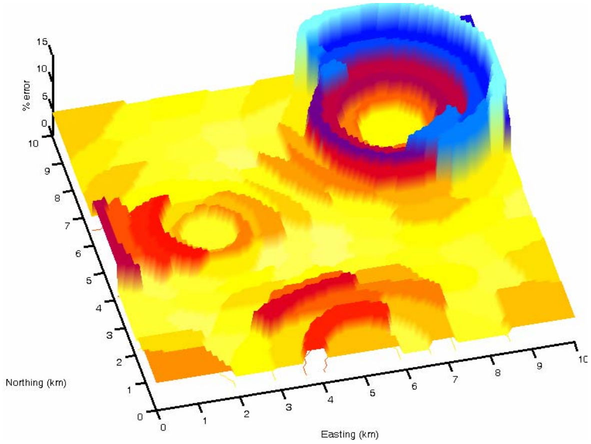


Figure 6: Traffic Sampling controller distribution error z(k) throughout the mobile peer network. Whilst the stability of the traffic sampling controller is consistent throughout the network, slightly higher errors are exhibited in areas where there are fewer peer nodes. This is due to the inverse relationship of the controller error with the background traffic.

V. CONCLUSIONS

This paper has presented a solution that exploits traffic data and neural networks in wireless peer networks to optimise the power consumption profile in mobile clients. Results have shown that the solution also provides better packet allocation efficiency on the radio interface compared to other algorithms [12-15], for the same number of peer clients. The significant improvement has been obtained by consuming fewer mobile CPU cycles by providing and allocating resources while taking into account the conditions of the radio environment, together with the availability of traffic channels at any specific time or area within the wireless network.

The net result from the effective use of this method is to deploy the algorithm across multi-technology topologies in a step towards MP2P convergence. The simulation results have shown that the proposed method promises high power efficiency across network switching topologies as different technologies are exploited to provide various mobile services using these overlay networks.

Research is currently ongoing to investigate peer node transmission across smaller inter-nodal differences to reduce power consumption, using an algorithm based on a derivative of the state-switching method used in this work. This will allow for more efficient use of the radio spectrum as well as the peer node burst rates with peer nodes having relatively close proximity.

Additional research is also envisaged in investigating the performance of the neural network algorithm in high mobility scenarios, that is where peer users are moving at speeds in excess of 60ms^{-1}, and to determine an optimal feedback mechanism to respond efficiently across multiple radio access network topologies. These methods will allow for the mobile power dissipation to be more efficient in a technology- and mobility-agnostic environment.

REFERENCES

[1] J.K. Buhagiar, C.J. Debono “Exploiting Traffic Sampling Techniques to Optimize Energy Efficiency in Mobile Peer Networks,” in Proc. of the 2009 First International Conference on Advances in P2P Systems (AP2PS 2009), pp. 72-77, October 2009.

[2] J.K. Buhagiar, C.J. Debono “Exploiting Adaptive Window Techniques to Reduce TCP Congestion in Mobile Peer Networks,” in Proc. of the 2009 IEEE Wireless Communications and Networking Conference (WCNC 2009), April 2009.

[3] Y. Luo, O. Wolfson, “Mobile P2P Databases”, in NSF Workshop on Data Management for Mobile Sensor Networks (MobiSensors), Pittsburgh, PA, January 2007.

[4] A. Datta, S. Quarteroni, and K. Aberer, “Autonomous Gossiping: A Self-Organizing Epidemic Algorithm For Selective Information Dissemination in Wireless Mobile Ad-Hoc Networks,” in The International Conference on Semantics of a Networked World, 2004.

[5] A. Helmy, Efficient Resource Discovery in Wireless AdHoc Networks: Contacts Do Help. Book Chapter in Resource Management in Wireless Networking by Kluwer Academic Publishers, May 2004.

[6] R. Krishnan, M. Smith, R. Telang, The economics of peer-to-peer networks, Carnegie Mellon University, 2002.

[7] J. Hellerstein, W. Hong, S. Madden, K. Stanek “Beyond Average: Toward Sophisticated Sensing with Queries,” in Proc. of the 2nd Int. Workshop on Information Processing in Sensor Networks, 2003.

[8] BitTorrent [Online]. http://en.wikipedia.org/wiki/BitTorrent © 2009

[9] Gnutella [Online]. http://en.wikipedia.org/wiki/Gnutella © 2009

[10] V. Sekar, M. Reiter, W. Willinger, H. Zhang, “Coordinated Sampling: An Efficient Network-wide Approach for Flow Monitoring,” Technical Report, CMU-CS-07-139, Computer Science Dept., Carnegie Mellon University, 2007.

[11] D. Kim, M. Lee, L. Han, H. In “Efficient Data dissemination in Mobile P2P ad-hoc networks for ubiquitous computing”, in Proc. of Int. Conf. on Multimedia and Ubiquitous Engineering, pp 384-389, 2008.

[12] A. Visvanathan, J. H. Youn, and J. Deogun, “Hierarchical Data Dissemination Scheme for Large Scale Sensor Networks,” in IEEE International Conference on Communications (ICC'05), pp. 3030-3036, May 2005.

[13] B. Karp and H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Sensor Networks,” in The 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), pp. 243-254, Aug 2000.

[14] Y. Xu, J. Heidemann, and D. Estrin, “Geography-informed Energy Conservation for Ad hoc Routing,” in The ACM International Conference on Mobile Computing and Networking, pp. 70-84, Rome, Italy, July 2001.

[15] Y. Yu, R. Govindan, and D. Estrin., “Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks,” Technical Report UCLA/CSD-TR-01-0023,UCLA, May 2001.

[16] R. Oliveira, L. Bernardo, and P. Pinto, “Flooding Techniques for Resource Discovery on High Mobility MANETs,” Workshop on Wireless Ad-hoc Networks, 2005.

[17] J. Kulik, W. Heinzelman, and H. Balakrishnan, “Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks,” Wireless Networks, vol. 8, pp. 169-185, 2002.

[18] H. Ballani, P. Francis, “CONMan: A Step Towards Network Manageability”, in Proc. of ACM SIGCOMM, 2007.

[19] V. Sekar, M. Reiter, W. Willinger, H. Zhang, R. Kompella, D. Andersen, “cSAMP: A System for Network-Wide Flow Monitoring,” in Proc. of the 5th USENIX Symposium on Networked Systems Design and Integration, 2008.

[20] S. Lawrence, C. L. Giles, A. C. Tsoi, A. D. Back, “Face Recognition: A Convolutional Neural Network Approach,” IEEE Transactions on Neural Networks - Special Issue on Neural Networks and Pattern Recognition, Vol. 8, No. 1, pp 98-113, 1997.

[21] Symbian OS Emulator 5th Edition, Symbian C++ SDK © 2009 Symbian Foundation.

[22] Google Android OS Emulator 1.5r2, 2.0 © 2008-9 Google, http:// code.google.com/android

[23] Control Systems Toolbox MATLAB © Mathworks 2009.

[24] Neural Networks Toolbox MATLAB © Mathworks 2009.

[25] P2P Multicast Library (PML), © Sourceforge, 2009, http://pml.sourceforge.net

[26] L. Feeney, M. Nilson, “Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment,” INFOCOM, 2001.

[27] O. Wolfson, B. Xu, R. Michael Tanner, “Mobile Peer-to-peer Data Dissemination with Resource Constraints,” in Proc of the International Conference on Mobile Data Management, pp 16-23, 2007.

[28] J.K. Buhagiar, C.J. Debono “Optimizing Multicast Protocols to Reduce Energy Dissipation in Mobile Peer Networks,” in Proc. of the 2010 IEEE Wireless Communications and Networking Conference (WCNC 2010), April 2010.

Replica Placement Algorithm based on Peer Availability for P2P Storage Systems

Gyuwon Song*‡, Suhyun Kim‡, Daeil Seo*‡ and Sunghwan Jang*‡
Human Computer Interaction and Robotics Department, Imaging Media Research Center‡*
University of Science and Technology, Korea Institute of Science and Technology‡*
Seoul, Korea
{*sharp81, suhyun.kim, xdesktop, jangc*}*@imrc.kist.re.kr*

***Abstract*—Peer-to-peer (P2P) technology is an emerging approach to overcoming the limitations of the traditional client server architecture. However, building a highly available P2P system is quite challenging, in particular a P2P storage system. The reason is due to the fundamental nature of P2P systems: peers can join and leave at any time without any notice. Replication is one of the strategies in overcoming the unpredictable behavior of peers. A good replication algorithm should use the minimum number of replicas to provide the desired availability of data. The popular approach in the previous studies is a random placement of replicas, but it ignores the wide difference in the availability of each peer. In this paper, we propose PAT (Peer Availability Table) in order to analyze and predict the state of nodes and develop a replica placement algorithm, which exploits the availability pattern of each individual peer. By comparing our algorithm with a random placement scheme, we show that our algorithm dramatically improves the data availability with moderate overhead in terms of memory consumption and processing time in both ideal and practical conditions. Additionally, we demonstrate the application of PAT as an analysis tool for various P2P systems.**

***Keywords*-Peer-to-Peer storage system, replica placement, peer model, availability, BitTorrent**

I. Introduction

Peer-to-peer (P2P) technology is an emerging approach to overcoming the limitations of the traditional client-server architecture. P2P systems can provide high scalability and reliability by using peers' donated resources, including computing power, network bandwidth, and disk space. The P2P storage system focuses on storage service among the P2P systems. A previous version of this paper is published in [1], in which a replication scheme to build a highly available P2P storage systems. P2P storage systems such as OceanStore [2], Farsite [3], Freenet [4], and PAST [5] take advantage of the rapid growth of network bandwidth and disk size to provide persistent storage without central servers. Unlike P2P file sharing systems, P2P storage systems provide the functions that not only read data but also write, as does a traditional storage system. In this system, all data should be stored at other nodes redundantly because data could be corrupted. Also, data should be encrypted to guard the user' s privacy because we cannot trust every peer. The users in the P2P storage system can access their data anytime, anywhere even though the computer that has worked with that data recently may be offline. Another advantage of the P2P storage system is that we have seemingly limitless storage space by using other nodes' storage. Namely, we can use the P2P storage system for storing data at other nodes' disks and then retrieve that data on demand. Furthermore, once data is written to the P2P storage system, that data is replicated and distributed to other nodes automatically. Thus, we can use the P2P storage system as an automatic backup solution.

However, building a highly available P2P system is quite challenging, in particular a P2P storage system. The reason is due to the fundamental nature of the P2P systems: peers can join and leave at any time without any notice. In other words, peers are not always available. Moreover, each and all of the availabilities are diverse. To make it clear, we set the definition of the $availability$ to be defined as "The degree to which a system, subsystem, or equipment is operable and in a committable state at the start of a mission, when the mission is called for at an unknown, i.e., a random, time" [6]. The various availabilities of peers are important keys to improving the data availability in the P2P storage system. Assume that several peers are not available at the moment. Then, generally the data availability would be decreased. At this point, however, the data availability could improve if the remaining available peers take the responsibilities for storing and providing the required data as a substitute for unavailable peers. In order to this, the popular approach in recent studies is the random placement of the replicas, but that ignores the important properties of peers in the P2P storage system. The motivation of our research, millions of nodes have the potential to improve availability, is a highly available P2P storage system.

This paper argues a replica placement algorithm to enhance the data availability of the P2P storage system. The main idea is that select nodes, which have the most different availability but reasonable one among peers for replica storing. To evaluate the difference between peers' availability, we make PAT (Peer Availability Table) to represent a peer's availability. PAT is automatically managed by using a DHT-based P2P system. This novel algorithm can maximize the data availability efficiently with minor overhead.

The rest of this paper is organized as follows: in section II we describe a survey of related work; after the description of the measurements and analysis of host availability results in section III, we propose our algorithm in section IV and its simulation evaluation in section V. In section VI, applications of our peer model are presented; and finally, we discuss the limitation of our work, and conclude this paper in section VIII.

II. Related work

There are many kinds of P2P systems such as file sharing, computing resource sharing, VoIP, and so on. Among them Farsite[3], Freenet [4], OceanStore [2], FreeHaven [7], Eternity [8], and PAST [5] are global storage systems intended for providing the scalability and self-organization of systems with persistence and reliability. [4] [7] [8] are more focused on the anonymity of users and anti-censorship for contents sharing. [3] is a server-less distributed storage system that has traditional storage system semantics. A directory group is used to ensure that the files are never lost. It showed that data is never lost as the maximum size of the clients is at the order of 10^5. Following this, we chose the size of the peer availability measurement. [2] provides a global persistent storage service that supports updates on replicated data. It uses erasure coding to provide redundancy without the overhead of strict replication and is designed to have a very long Mean-Time-To-Data-Loss (MTTDL). Also, [5] is a simple storage service for persistent, immutable files. It uses randomization to ensure diversity in the set of nodes that store a file's replicas and to provide load balancing. [9] [10] [11] described the replication method based on location, data consistency.

In order to improve the availability, a measurement of peer availability has to be done first. Since BitTorrent [12] has become the most popular among P2P file sharing systems, currently, there are many studies of measurements and analysis on BitTorrent systems. Izal et al. [13] analyzed a five-month workload of a single BitTorrent system for software distribution with thousands of peers, and evaluated the performance of BitTorrent at the flash crowd period. Bellissimo et al. [14] analyzed the BitTorrent traffic of thousands of torrents over a two-month period regarding the shared file characteristics and client access characteristics. Guo et al. [15] provided an understanding of torrent evolution in the BitTorrent systems and the relation among multiple torrents over the Internet. Measurements [16] [17] [18] characterize the P2P file sharing system's traffic over the Internet, including Napster, Gnutella, and KaZaa systems. But, they mainly focused on the performance of those systems. To easily measure a P2P network, Global Internet Measurement System (GIMS) was proposed in [19].

Some redundancy schemes are proposed to improve the data availability. In [3], their simulation showed that the random replica placement is better than the replacements that consider availability, due to the fairness of nodes' load. However, it was designed to support typical desktop workloads in academic and corporate environments. Because of this, nodes are less dynamic than in real P2P environments. TotalRecall [20] also uses a random placement scheme and proposed a static model for estimating data availability. They evaluated the data availability by the mean availability of peers as a parameter. Bhagwan et al. [21] and Blake et al. [22] used this scheme in the same manner. It is not a practical approach because they use a static model to evaluate host availability. Tian et al. [23] [24] studied the dynamic pattern of the Maze system and proposed a similar-MTTF-MTTR data placement scheme under the time-related model of data availability. They suggested the data availability should be considered as not a constant probability model, but a time-related probability model. [25] also pointed out that which uses the information of the session time to prevent the burst failures.

[3] [26] use the Hill-Climbing algorithm, which uses peers with high availability to replace peers with low availability. They are based on the host availability measurement to improve the data availability. Our research started from this concept. However, it is hard to analyze the host availability under this scheme since there is no availability model to measure it.

III. Measurement and Analysis of Availability

In this section, we describe a measurement and its analysis of the peer availability. We indentify the peer availability with the host availability and mainly use the former in this paper. The choice of the P2P system to measure the peer availability is quite an important decision because the results of measurement and analysis could be easily biased if we chose a P2P system which is popular in a specific country or is used for academic or research infra such as [27] [28]. So, we chose BitTorrent [12] as a representative P2P system. BitTorrent is a P2P file sharing protocol and has recently been showing strong growth. Usage of the protocol accounts for significant Internet traffic, though the precise amount has proven difficult to measure. According to a report [29], BitTorrent traffic occupied 53% of all P2P traffic on the Internet in June 2004. There are millions of simultaneous users and various client applications, as well. BitTorrent-like systems work in the following manner. The content provider creates a *.torrent* (meta-data) file for content sharing, and publishes that file on a public web site or tracker web site. For each torrent file, there is a tracker site, whose URL is encoded in the torrent file, to find peers with whom to exchange the file chunks. The content provider then runs a BitTorrent client with a complete file to share as a seed. Another user who wants to download the file starts a BitTorrent client with the *.torrent* file as a leech. After a handshake process, data transferring begins. Since any node can join or leave at anytime, data availability is highly reliant on the arrival and departure of peers in particular seed nodes.

A. Methodology

Our measurements have some assumptions: 1) BitTorrent users represent the other users in most P2P systems very well. 2) BitTorrent users keep running a BitTorrent client when they are using their computers. 3) There is no hacked client that is designed to not respond to a BitTorrent handshake. 4) Network addresses of the users in BitTorrent are not changed once assigned, so it can be used as an identifier. Based on this, we intensively measured the availability with the numerous peers in the BitTorrent system. We did not consider the tracker sites or system performances to up/download files because we were mainly focusing on the peer availability.

Protocol	BitTorrent v1.0
Date of measurement period	October, 2008
Duration of measurement	Oct.6 Oct.20 (2 weeks)
Measurement interval	Every 5 minutes
Measurement methodology	BitTorrent Handshake
Number of torrent files	100
Number of peers	96,749
Result record sets	492,829,138

Table I
SUMMARY OF THE MEASUREMENTS

Table I outlines our measurements. To analyze the peer availability of BitTorrent users, we gathered a large size of BitTorrent clients' log files and then extracted a list of the IP addresses and port numbers. Popular tracker sites such as Suprnova.org, Piratebay.org, and Movierg.com were used to obtain one hundred torrent files. Those torrent files had held a top 10 rank just before our measurement in Audio, Video, Application, Game, and Other categories. The number that we collected of distinct peers added up to 96,749.

We devised a multi-thread crawler to evaluate the peer availability concurrently. Since the crawler does not participate in swarm for transferring contents, no file chunk was transferred. The crawler tries to establish a connection (BitTorrent handshake) with the listed peers every 5 minutes and then inserts the results into the database. Inserted records total more than 490 million and all peers have exactly 4,032 records after filtering out. It means no errors occurred during our measurement.

B. Result Analysis

Geographical distribution Table II shows the geographical distribution of peers that are piled in our measurement. We extracted the geographical distribution from the users' IP addresses using MaxMind GeoIP [30].

North america holds the foremost position among other countries and Europe is the second. The interesting thing is that peers in BitTorrent are distributed world-wide in 191 countries, however distribution is not even but dominated by just three countries(US, UK, and Canada).

Rank	Country	# of peers	Percent
1	United States	22,532	23.29%
2	United Kingdom	9,043	9.35%
3	Canada	7,534	7.79%
4	Australia	4,168	4.31%
5	Sweden	3,091	3.19%
6	Poland	2,984	3.08%
7	Brazil	2,681	2.77%
8	France	2,669	2.76%
9	Netherlands	2,075	2.14%
10	India	2,035	2.10%
11	Norway	1,946	2.01%
12	Spain	1,767	1.83%
13	Portugal	1,587	1.64%
14	Philippines	1,581	1.63%
15	Germany	1,538	1.59%
16	Italy	1,516	1.57%
17	Greece	1,509	1.56%
18	Malaysia	1,457	1.51%
19	Korea	1,300	1.34%
20	Finland	1,253	1.30%
Others	< 1%	22,483	23.24%
Total	191 countries	96,749	100%

Table II
GEOGRAPHICAL DISTRIBUTION OF PEERS

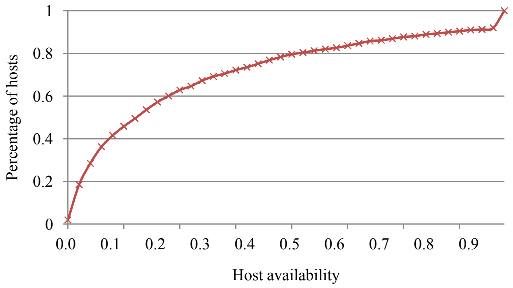


Figure 1. Cumulative distribution of the host availability

Peer availability We plot the cumulative distribution of the peer availability in Figure 1. The average of the availability is 28.39% and the median is 15.67%. It indicates that about 90% of the peers were not available which means only less than 10,000 peers frequently joined to BitTorrent during the two weeks of our measurement. Since neither errors with our crawler system nor network occurred while probing, probably BitTorrent users have a temporary usage pattern and serious lower availability. We deduce possible causes from this result. The first is a network related problem such as DHCP, firewall or NAT. Firewall and NAT do not allow the crawler to reach the peers which are behind them. Moreover, a peer's network address may be changed when it is newly leased from the DHCP server. [28] also pointed out limitations of which all these approaches rely on IP addresses. Probing by IP address does not accurately capture the available characteristics of the hosts. IP address probing would consider each IP address a new host, thus greatly overestimating the number of hosts in the system and underestimating their availability. Using a unique identifier for a BitTorrent user could solve that, but most BitTorrent systems, including trackers and clients, do not support that in order to avoid an invasion of privacy or legal issues. The second possibility is derived from a specific usage pattern of BitTorrent users. Once a downloading process is completed, users do not run a BitTorrent client until they need to download another file. Basically, users in the P2P system tend toward selfishness; they want to use other resources freely as a free-rider and to avoid uploading what they have already downloaded from others. To allure users to share their surplus resources, a compensation mechanism is indispensably required.

On the other hand, about 0.9% of peers keep a ready-available state such as a server. Those peers appear at the right side of the curve in Figure 1. In all probability they use an exclusive BitTorrent machine all day to share contents. To improve the data availability, P2P storage systems should utilize these altruistic peers. Overall, the result of our measurement is similar to [23] [28] with respect to the host availability. And, some peers have diurnal online patterns but high available peers do not [23]. We have been maintaining the host availability measurement for long periods of analysis.

Time-related variation of peer availability Figure 2 presents the variations of three peers' availability as time passes. Three peers were sampled among those who had 28% availability which is the mean availability of the whole of the peers and those are distinguished by red, green, and blue. The circle in Figure 2 represents a wall clock (for 24 h) and a length of the radius refers to the availability. Namely, the red peer is highly available from 03:00 - 15:00, the green peer from 18:00 - 05:00, and the blue peer from 14:00 - 21:00. Even though they have the same average availability, their time-related variation of peer availability is conspicuously changed as time passes during the day. Here is a significant chance to improve the data availability. Previous works have a simple model to represent the peer availability, such as a static model. It assumed all peers have a static availability in the long term but it will change in the real P2P environment, as shown in the graph. It implies that we can maximize the data availability by selecting well-timed peers for failure tolerance. If we chose those three peers for replica storing, then we can access the data at anytime with high probability. However, assuming that we randomly, by the previous works' policy, choose some peers who have a static mean availability, $p = 0.3$. Then the availability is 90% when 12:00 15:00 and the rest of time's availability is nearly 0%. Possibly it is hard to access to the data except when 12:00 - 15:00. This point is the motivation of our research.

Other results If we extend the online pattern window from a day to a week, we can find the diverse distribution of availability along with, not only time of day, but also day of the week. For example, a five-day a week worker's home PC would not be used during the weekdays but be used mainly on the weekends. Also, the cultural special days which lead people to go out, such as Thursdays for a shopping day in Australia, make the weekly pattern more distinct.

IV. Replica Placement algorithm

After considering the factors in the previous section, we reached these preliminary conclusions. Many peers in the P2P system have low availability. Their availabilities are dynamically changed as time passes, not only the time of a day but the day of the week. In this environment, i.e., peers can join and leave the system at any time and node failure is no longer an exceptional event, but is common. Even though the P2P systems employ some schemes of data redundancy to recover unavailable data, it remains unclear what availability guarantees can be made using existing systems, or conversely how to best achieve a desired level of availability using the mechanisms available [21]. In this section, we describe the details of our system for replica placement. In particular, we propose a probabilistic model to represent the peer availability and a novel algorithm to improve the data availability based on our peer model for a highly available P2P storage system. Since we assume our system is built on a DHT system as [2] [5], it follows the DHT system's protocol.

A. Replication schemes

The goal of the replication is to use the minimum number of replicas to provide a high availability of data. To do this, both file replication and erasure coding schemes are generally used in a P2P system. File replication is a simple strategy that makes n copies of the file and puts them on different hosts. However, reproducing an entire file that is not accessible can be a burden in both storage space and time. Block-level replication makes it somewhat better, but if any single block cannot be found then the entire file object is useless. Erasure coding [31] provides the property that a set of b original blocks can be reconstructed from any m coded blocks taken from a set of cb (where m is typically close to b, and c is typically a small constant). Then, m coded blocks are stored at different hosts. As mentioned above, the key idea of the replication is that it makes redundancies of the original file and distributes them over other hosts for failure tolerance. In these procedures, we are mainly focusing on the 'other hosts', i.e., the best place to store redundancies of the file in order to maximize the data availability.

B. Peer Availability Table

Peer Availability Table (PAT) is a probabilistic model which represents peer availability. We designed a peer availability table and its manipulation methods. PAT indicates a

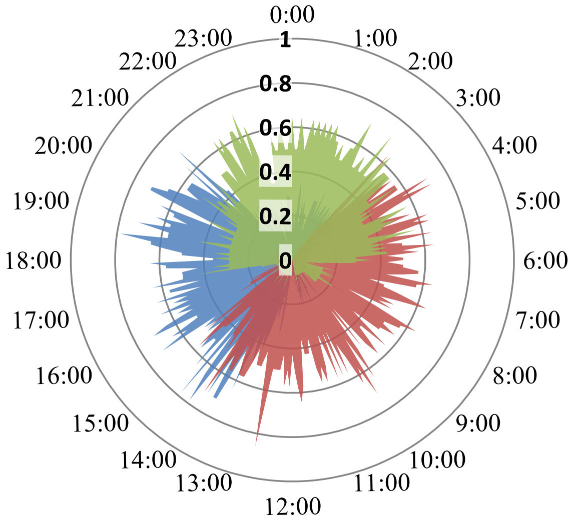


Figure 2. Host availability distribution with time of day for three sampled peers, which have the mean availability

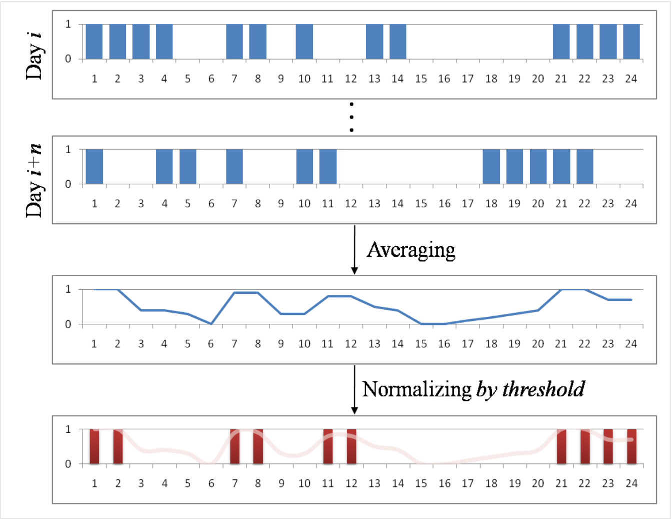


Figure 3. Process of building a Peer Availability Table

peer's availability at every five minute interval during a one-week period. A time slot is a basic unit to discretely divide the stream of time. The length of a time slot is five minutes, the same as our measurement interval. So an hour has 12 time slots, a day has 288, and a week has 2,016. We define the time slot 0 as Monday 0:00 and the last one as Sunday 23:55. The availability with time slot i is calculated by

$$AV_i = \frac{OnlineCounts}{VerificationCounts}$$

PAT can reflect the diverse aspects of characteristics of a peer. Assume that user A uses his office PC at the work place during the daytime on weekdays but uses his home PC at home all day long on the weekends (i.e., User A rarely uses his home PC on weekdays and his office PC on weekends). If PAT includes just daily availability, the availabilities of these two PCs would be underestimated by a mean value even if some days' availabilities are relatively high which it cannot neglecte to utilize. Besides, the cultural special days, which incur less use for a P2P system than other days, such as Thursdays for a shopping day in Australia, are well reflected. More importantly is that since PAT has the information of the weekly availability, the P2P storage system can acknowledge the long-term availability of a peer that uses it for detecting a permanent leave of the system. The threshold time to decide a peer's permanent leave can be assigned from several hours to seven days. For example, suppose a peer's PAT is as follows:

Time Solot	0	1	2	...	2015
Verification count	28	20	17	...	25
Online count	21	17	17	...	20

Table III
AN EXAMPLE OF PAT

Table III refers that the peer who has 0.75 availability on Monday 0:00 - 0:04 and 0.8 availability on Sunday 23:55 - 23:59. After some weeks, if all availability values converge to zero for every time slot, the system can then perceive that this peer leaves the system permanently and starts a recovery process for inaccessible data. The $VerificationCounts$ can be different from others. Because there is no central server to evaluate all peers' availability in the P2P system, all peers have to evaluate partially along with neighbor peers and by themselves. Each peer manages a personal PAT and neighbors' PATs only when they are online. Since peers independently join and leave the system and only verify when they are online, the $VerificationCounts$ can be different with each other's neighbors. The process of PAT management is as follows. A peer's PAT is initialized with all values at zero for the first time bootstrapping to the P2P storage system. During the bootstrapping step, the peer gets a routing table that includes neighbor nodes lists. According to the list, the peer creates PATs for neighbor nodes. The peer's PAT is actively updated every 5 minutes by itself, while the neighbor nodes' PATs are updated passively. The passive method is based on a heartbeat message. All nodes in DHT-based systems send a heartbeat message (or keep-alive message) to their neighbor nodes to inform them that a node is available periodically. Peers who receive a heartbeat message from another node increase the $VerificationCounts$ and the $Onlinecounts$ of the sender's PAT at the correspondence time slot to the received time. Following these processes, all nodes' PATs are maintained up-to-date. Since a user's usage pattern may be changed over time, clearly, the PAT should be renewable in a few weeks or months. This is a topic for future consideration.

C. The Similarity of PAT

According to section III, peers are very dynamic (i.e., each peer has a diverse PAT), but we can classify peers by the degree of similarity of PATs. Because there are a large number of peers in the P2P system, it is probable that there exist some peers who have a similar PAT. The degree of similarity between peer A and peer B is calculated by

$Sim_{A,B} = \Sigma P_{A_i} * P_{B_i}$, where P_{A_i} is available at time slot i on peer A's PAT, and P_{B_i} is respectively. A high degree of similarity between peer A and peer B means that their usage patterns are very much alike, while a very low degree of similarity means they use the system oppositely from each other. Our novel algorithm for replica placement begins at this point. Choosing peers who have a high degree of similarity ensures a highly available P2P storage system when trying to access data at a usual usage time for the owner. On the other hand, choosing peers who have a low degree of similarity also ensures even when trying to access data at quite a different time from his normal usage pattern. Note that when selecting nodes, these nodes' average availabilities must be greater than the threshold to guarantee minimum availability since the similarity will be zero if calculated with a peer whose availability converges to zero. Therefore, if the degrees of similarity were less than a threshold, that combination would be discarded.

D. Replica Placement Algorithm

In this section, we describe how our algorithm works for replica placement based on PAT. To build a highly available P2P storage system, all data must be stored redundantly. The core algorithm, to be brief, is that it finds the best combinations among peers who have various similarities of PAT to maximize the data availability. We refer to the chosen peers who maximize the data availability as a max_list. The size of a max_list is decided by a given parameter as target availability, so it would minimize overhead that is related to network bandwidth and storage space. The following pseudo-code is our algorithm to construct a max_list. The parameters of this method are a $file\_id$ that wants to store and target availability that represents how important it is. In P2P storage systems such as [5], mostly, each node is assigned a 128-bit node_id, derived from a cryptographic hash of the node's public key. Each file is assigned a larger bits $file\_id$ rather than a $node\_id$. When a file is inserted into the system, some nodes are selected whose $node\_id$ are numerically closest to the 128 most significant bits of the $file\_id$. The $candidates\_list$ is compiled of those nodes and is relatively long. To filter out useless nodes in the $candidates\_list$, similarities are calculated with the host user's PAT and organized into the $similar\_list$. This procedure would reduce the processing cost of node filtering. Composing a max_list is as follows. A peer who has the highest similarity value is added in the max_list. Then, generates all combinations of nodes, i.e., ${}_nC_r$, where n is the size of the $similar\_list$ and r is $n-1$, and calculates the availability with their PATs. The data availability is evaluated by integrating the probability density function of the peer's PAT. The loop will be terminated if evaluated availability is greater then the $target\_availability$. Finally, the combination nodes that have the highest data availability construct the max_list. Replicas are placed to nodes according to the

max_list.

Algorithm 1 MaxList($fileId, target\_availability$)

```
$candidates\_list \Leftarrow$ nodes for given $file\_id$
**for all** each node $n$ in $candidates\_list$ **do**
  $similarity \Leftarrow$ calc similarity $(n, me)$
  **if** $similarity > threshold$ **then**
    $similar\_list \Leftarrow$ add node $n$
  **end if**
**end for**

**for** $i = 0$ to $similar\_list.size$ **do**
  $data\_availability\_list \Leftarrow$ calc $dataAvailability$ with
  all combinations $(similar\_list.size, i)$

  sort list order by $data\_availability$
  **if** $data\_availability > target\_availability$ **then**
    break
  **end if**
**end for**

$max\_list \Leftarrow$ the top entry of the $data\_availability\_list$
**return** $max\_list$
```

V. Simulation evaluation

We designed a simulation environment to evaluate our new algorithm. The simulator is based on a P2P storage system, which is built upon a DHT system. We detached a part that is node selecting from the system and simulate this part only. Even though we do not simulate a whole P2P storage system, the result of our simulation is worthy because we use real trace data that we measured which evaluates data availability in our simulation by replaying that data. We explore the improvement of data availability by our novel algorithm and compare it with the random placement algorithm. Our simulation is focusing on how to achieve the best data availability using replication schemes with low overhead. All simulations were performed 100 times with various simulation settings under an Intel Core Duo 2GHz/2GB RAM PC.

A. Assumptions

Actually, there may be hundreds of thousands of users in a P2P system. Since it is impossible to simulate the whole size of the P2P system, 1,000 peers were randomly sampled among all peers that we had measured for the peer availability. As we mentioned in section III, peers have various availabilities and its average is very low. Douceur et al. [32] reported that about 60% of the nodes in Napster and Gnutella systems had less than 70% availability, due mainly to the dynamic behavior. Moreover, our measurement of the BitTorrent system shows that about 80% of the nodes have less than 50% availability and the top 10% of the nodes have more than 90% availability. Therefore, the modeling of peers for a simulation is a critical factor of the data availability measurement since it is one of the immediate causes. We made peer models of three types: a dynamic peer who cyclically turns on and off the system; a server-like peer who is always online; and, an inactive peer who rarely uses the system, i.e., $p < 0.1$. To make a worsecase scenario, we assume no server-like peers exist in our simulation but inactive peers are considered from 0% to 75%. In our simulation, all used PAT data are real measured data during a two-week period and we evaluated the data availability by replaying its results. The evaluation ran a hundred times during a one-week period for each data. To simplify our simulation has some limitations; the network environment is error free and all data transmission is completed at once. The size of the $candidates\_list$ and $target\_availability$ are given as a parameter. In future work, we consider the whole simulation of a P2P storage system in a real network based on long-term measurement analysis.

B. Simulation Results

Our simulation results are divided into three topics: the improvement of data availability, the overhead of processing cost and memory usage, and the differences with respect to the change of configuration settings. We measured the data availability improvements and the processing cost on the ideal condition in which there are no inactive peers. In this condition we only changed the size of the $candidates\_list(n)$ and the $max\_list(k)$, which is referred as ${}_nC_k$. Note that the actual size of the max_list is fixed to fulfill the $target\_availability$. However, to compare more clearly, we input that as a parameter. We also observed the overhead of our algorithm with respect to memory usage. Lastly, we evaluated the effects of the ratio of peer models between a dynamic peer and inactive peer. Studies show that most peers have very low availability in our measurements as well as prior works. So, the configuration schemes of the inactive peer ratio were set as 0% (ideal), 10%, 30%, and 50%. In this simulation, we treat a peer whose availability is less than 10% for two weeks as the inactive peer.

We plot the main results of our simulation in Figure 4. Using our new algorithm can reduce the number of replicas of a file by half rather than a random placement in order to provide the data availability of over the 99.99%. Since the size of multimedia contents has been increasing recently, the resource usage is seriously important in both network bandwidth and disk space. Therefore, the result means that our algorithm can greatly improve efficiency of the resource usage rather than a random placement scheme. The size of the $candidates\_list$ does not impact on the data availability. However, the processing cost rapidly increases along with its size, as shown in Figure 5. The processing time is just 1 (msec) to select nodes by random but increased from 90 to

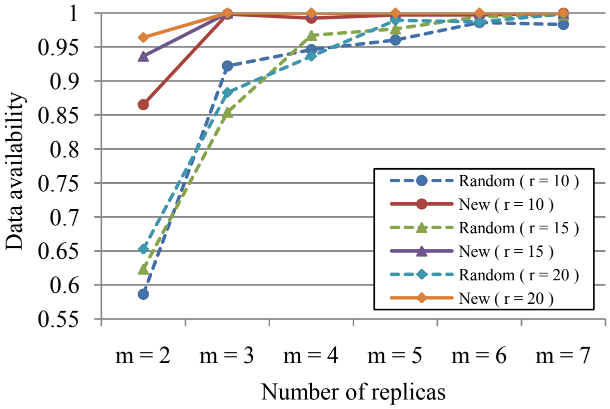


Figure 4. The data availability and required number of replicas

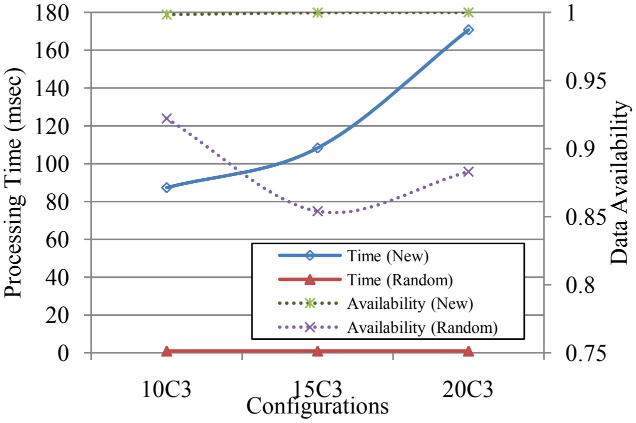


Figure 5. The processing time overhead with data availability

170 (msec) according to following configurations of ${}_{10}C_3$ to ${}_{20}C_3$. Considering the network latency time in real networks, a faster setting is much better but we think it is an affordable cost to get the high availability. We conclude that the size of the *candidates_list* is 15 which is sufficient to provide very high data availability when using our new algorithm. Figure 6 shows the memory usage for each algorithm and it includes all users of PATs that are used in our simulation and related objects to manage PATs. There is a slight difference between the two algorithms, about 1.5MB. In order to use our algorithm, the host PC uses more memory but most PCs are good enough to ignore that. Considering all these results, our replica placement algorithm can greatly improve the data availability with affordable overheads rather than random placement algorithm.

Finally, we evaluated the effects of the ratio of peer models between a dynamic peer and inactive peer. We set the ratio of inactive peer as 10%, 30%, and 50% and compared it with the ideal condition (0%). At this point, we use a ${}_{10}C_3$ scheme for our algorithm. Our simulation was designed to perform under more practical situations. Figure 7 shows the results. Though the ratio of the inactive peer increased from 10% to 50%, our algorithm provided over 90% data availability. On the other hand, the random placement scheme's data availability had fallen to under 65%.

In conclusion, we show that our algorithm can greatly improve the data availability while minimizing the waste of resources rather than random placement in ideal and practical conditions, as well.

VI. Application of Peer Availability Table

Peer Availability Table (PAT) is the core system of our new replica placement algorithm. Namely, the system, using PAT, logs peers' availability and give a clue to predict their usage patterns based on it. As we mentioned above section, PAT system is well suited to BitTorrent system. In this section, moreover, we show a feasibility study on PAT

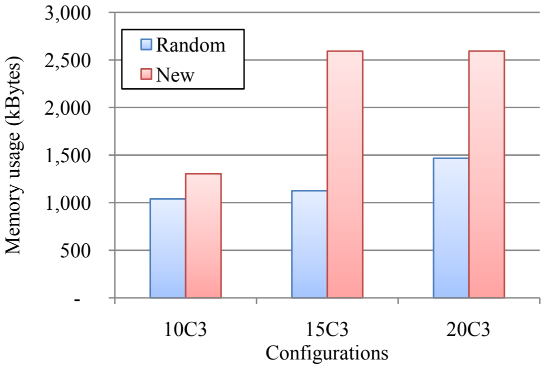


Figure 6. The comparison of the memory usage (Kbyte)

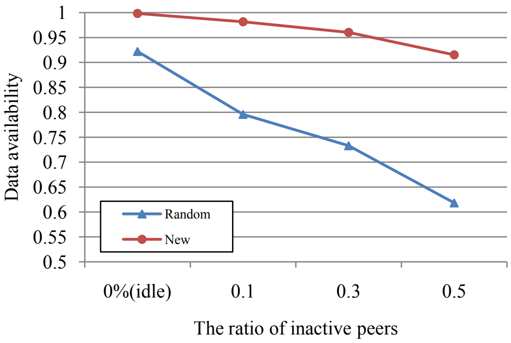


Figure 7. A change of the data availability with the ratio of inactive peers

system for other P2P systems. To explain it, we chose KAD trace [33] and Skpe trace [34].

The reason why we choose is that KAD trace has a huge log results in terms of number of peers and experiment period and has the measurement resolution same as ours. By the way, KAD is a Kademlia-based routing protocol [35] implemented by several peer-to-peer applications such as eMule and aMule which have a lot of simultaneous connected users. Skype, and then, which is P2P based VoIP system, has became a killer application of P2P systems recently. By the analysis of these trace data using PAT, we present an evidence that the feasibility of PAT as a tool to analyze of usage pattern for not only BitTorrent system but also any other P2P systems.

A. KAD

We sampled 12,241 peers which had checked as online for over 30 days among 400,375 whole peers. Though the original KAD trace has 5 minutes resolution of measurement, we adjust that to 1 hour since Skype trace has 1 hour resolution. Therefore the length of PAT was set 24 for a day and 168 for a week. Figure 8 shows the average of availability for a day and a week. The x-axis means time and the y-axis means the average of availability. The red line shows a daily pattern, the blue line shows a weekly pattern, and the green bar means the difference between two. Since a difference between a daily pattern and weekly pattern means over-/underestimation of availability, we have already pointed out that a daily/weekly pattern of peers clearly exists and a weekly pattern is more important in order to predict the usage pattern accurately. Specific days(Day1 and Day2) have a high availability rather than others(5 days), and then we can speculate that Day1 and Day2 are weekend, others are weekday.

B. Skype

Sky trace has total 4,000 nodes' result with 1 hour measurement resolution. Like as KAD, 2,081 super nodes of Skype trace are selected in this analysis. Interesting thing that the blue line decreases at Day5 and Day6 in figure 9. is the contrary result to KAD trace. Namely, two days' availabilities increase in KAD but decrease in Skype. In general, KAD is used to share file sharing but Skype is used to talk with acquaintances. Thus we can speculate that KAD is more employed at weekend for sharing files, however Skype is less employed at weekend for contacting through the Internet.

C. Usage pattern prediction

Based on PAT, we can predict nodes' states with high probability. In order to show the probability of prediction, we made a simple simulator. The simulator built PATs using the whole trace data of KAD and Skype except last one week result. And then, the simulator simply counts a difference a given PAT and a last one week log(which is excluded when building a PAT) of real trace. The result of prediction accuracy is plotted in figure 10. In KAD case, all results show a high accuracy ($> .9$) and the result which is using weekly PAT is slightly better than daily PAT. But the results of Skype are differ from KAD. The results which is using weekly PAT show $> .8$ stably, but the others which is using daily PAT go on worsening. This is another proof to illustrate that the weekly PAT is proper length.

VII. Discussion

As we described above, our novel algorithm can improve the data availability efficiently. However, there are some implementation issues in practical since it completely depends on availability logs of peers. Namely, all logs for each peer must be stored, and it is difficult to decide that who verify the status of a peer and how to share that logs. It remains an open problem to determine the best settings for real P2P environment.

VIII. Conclusion

In this paper, we studied an algorithm to improve the data availability in a P2P storage system. It is one of the most difficult topics in a peer-to-peer system. We showed that

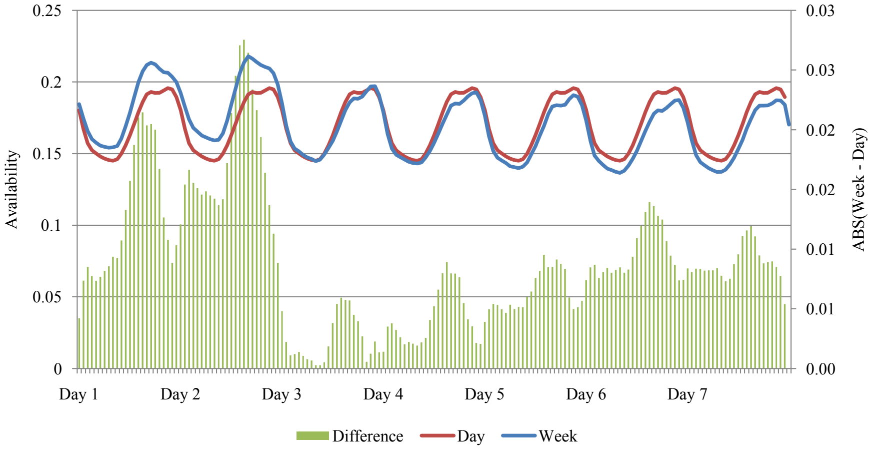


Figure 8. KAD trace

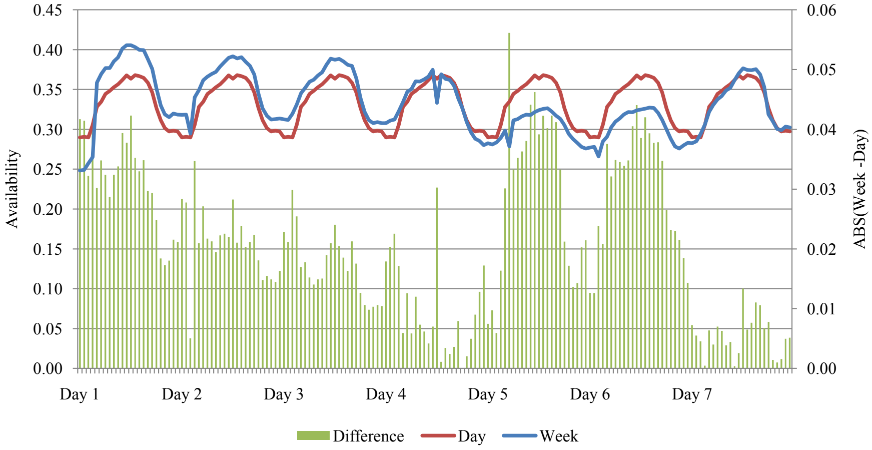


Figure 9. Skype trace

the peers' availability in a BitTorrent system as a result of our measurement. The results imply that the peer availability changed not only from time of day but also day of the week. The mean availability was relatively low due to the limitation of the IP based measurement.

Based on this result, we developed a probabilistic model, referred to as Peer Availability Table (PAT), representing a peer's weekly availability which means that it can cover the range from short-term availability to long-term availability in a simple manner. It can be used to find a peer who has similar usage patterns or detect permanent leave of a peer to failure recovery. We then propose a replica placement algorithm to maximize the data availability. To build a highly available P2P storage system, all data must be stored redundantly. The key was to make redundancies of the original files and distribute them over other hosts for failure tolerance. In these procedures, we mainly focused on the storage area of the file redundancies. Unlikely previous works that are a random placement scheme, our algorithm found the best combination of peers to provide the highest data availability among candidate peers. Because calculating data availability with all combinations is highly complex, we used a heuristic method to reduce the case of combinations by an estimation of similarity between PATs.

By comparing our algorithm with a random placement scheme, we showed that our algorithm dramatically improved the data availability with moderate overhead in terms of memory consumption and processing time in both ideal and practical conditions. Additionally, we demonstrate a feasibility of PAT as an analysis tool for P2P systems such

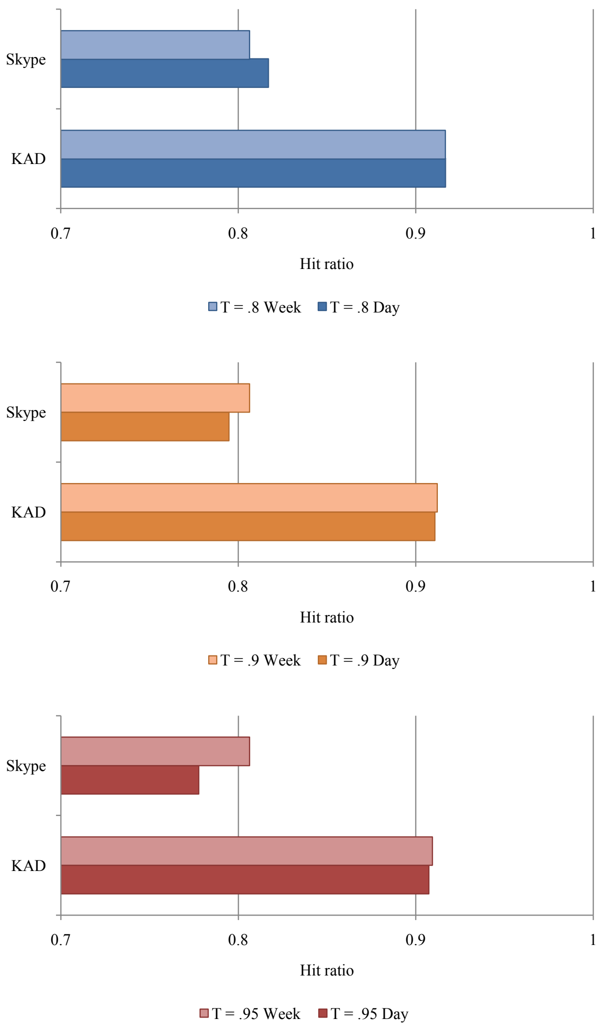


Figure 10. Prediction accuracy, T = threshold for building PAT

as KAD and Skype. It means that PAT is well suited not only BitTorrent system but also other P2P systems. Applying our new replica placement algorithm to other P2P systems and its verification in the real network are our ongoing works.

Acknowledgment

This work was supported by the IT R&D program of MKE/KEIT. [KI002119, Development of New Virtual Machine Specification and Technology]

References

[1] G. Song, S. Kim, and D. Seo, "Replica Placement Algorithm for Highly Available Peer-to-Peer Storage Systems," *Proceedings of First International Conference on Advances in P2P Systems*, pp. 160–167, 2009.

[2] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, C. Wells *et al.*, "Oceanstore: An architecture for global-scale persistent storage," *ACM SIGARCH Computer Architecture News*, vol. 28, no. 5, pp. 190–201, 2000.

[3] A. Adya, W. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. Douceur, J. Howell, J. Lorch, M. Theimer, and R. Wattenhofer, "Farsite: Federated, available, and reliable storage for an incompletely trusted environment," *OPERATING SYSTEMS REVIEW*, vol. 36, pp. 1–14, 2002.

[4] I. Clarke, O. Sandberg, B. Wiley, and T. Hong, "Freenet: A distributed anonymous information storage and retrieval system," *LECTURE NOTES IN COMPUTER SCIENCE*, pp. 46–66, 2001.

[5] A. Rowstron and P. Druschel, "Storage management and caching in past, a large-scale, persistent peer-to-peer storage utility," *ACM SIGOPS Operating Systems Review*, vol. 35, no. 5, pp. 188–201, 2001.

[6] F. Standard, "1037c: Glossary of telecommunications terms," *The Institute for Telecommunication Sciences, Oct*, 2006.

[7] R. Dingledine, M. Freedman, and D. Molnar, "The free haven project: Distributed anonymous storage service," *LECTURE NOTES IN COMPUTER SCIENCE*, pp. 67–95, 2001.

[8] R. Anderson, "The eternity service," vol. 96, 1996, proceedings of Pragocrypt.

[9] J. Gossa, J. Pierson, and L. Brunie, "FReDi: Flexible Replicas Displacer," in *Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006. International Conference on*, 2006, pp. 16–16.

[10] E. Sithole, G. Parr, S. McClean, and P. Dini, "Evaluating Global Optimisation for Data Grids using Replica Location Services," in *Networking and Services, 2006. ICNS'06. International conference on*, 2006, pp. 74–74.

[11] H. Yoshinaga, T. Tsuchiya, H. Sawano, and K. Koyanagi, "A Study on Scalable Object Replication Method for the Distributed Cooperative Storage System," in *Proceedings of the 2009 Fourth International Conference on Digital Telecommunications-Volume 00*. IEEE Computer Society, 2009, pp. 96–101.

[12] B. Cohen, "Incentives build robustness in bittorrent," vol. 6. Berkeley, CA, USA, 2003, workshop on Economics of Peer-to-Peer Systems.

[13] M. Izal, G. Urvoy-Keller, E. Biersack, P. Felber, A. Hamra, and L. Garces-Erice, "Dissecting bittorrent: Five months in a torrent's lifetime," *LECTURE NOTES IN COMPUTER SCIENCE*, pp. 1–11, 2004.

[14] A. Bellissimo, B. Levine, and P. Shenoy, "Exploring the use of BitTorrent as the basis for a large trace repository," *University of Massachusetts, Tech. Rep*, 2004.

[15] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurements, analysis, and modeling of bittorrent-like systems," 2005.

[16] S. Saroiu, K. Gummadi, R. Dunn, S. Gribble, and H. Levy, "An analysis of internet content delivery systems," *ACM SIGOPS Operating Systems Review*, vol. 36, no. si, p. 315, 2002.

[17] S. Saroiu, P. Gummadi, and S. Gribble, "A measurement study of peer-to-peer file sharing systems," vol. 2002, 2002, proceedings of Multimedia Computing and Networking.

[18] P. Gummadi, S. Saroiu, and S. Gribble, "A measurement study of napster and gnutella as examples of peer-to-peer file sharing systems," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 1, pp. 82–82, 2002.

[19] D. Xu, T. Liu, D. Qian, Z. Luan, and M. Tang, "A New P2P-like Architecture for Large Scale End to End Network Measurement," in *Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006. International Conference on*, 2006, pp. 62–62.

[20] R. Bhagwan, K. Tati, Y. Cheng, S. Savage, and G. Voelker, "Total recall: System support for automated availability management," 2004.

[21] R. Bhagwan, S. Savage, and G. Voelker, "Replication strategies for highly available peer-to-peer storage systems," *Proceedings of Fu-DiCo: Future directions in Distributed Computing, Jun*, 2002.

[22] C. Blake and R. Rodrigues, "High availability, scalable storage, dynamic peer networks: Pick two," 2003.

[23] J. Tian and Y. Dai, "Understanding the dynamic of peer-to-peer systems," 2007, proc. of the 6th Int'l Workshop on Peer-to-Peer Systems.

[24] J. Tian, Z. Yang, and Y. Dai, "A data placement scheme with time-related model for p2p storages," 2007, pp. 151–158, peer-to-Peer Computing, 2007. P2P 2007. Seventh IEEE International Conference on.

[25] K. Kim, "Time-related replication for p2p storage system," in *Proceedings of the Seventh International Conference on Networking*. IEEE Computer Society, 2008, pp. 351–356.

[26] T. Schwarz, Q. Xin, and E. Miller, "Availability in global peerto-peer storage systems," 2004.

[27] W. J. Bolosky, J. R. Douceur, D. Ely, and M. Theimer, "Feasibility of a serverless distributed file system deployed on an existing set of desktop pcs," *SIGMETRICS Perform. Eval. Rev.*, vol. 28, no. 1, pp. 34–43, 2000, 339345.

[28] R. Bhagwan, S. Savage, and G. M. Voelker, "Understanding availability," *Peer-to-Peer Systems Ii*, vol. 2735, pp. 256–267, 2003, kaashoek, F Stoica, I 2nd International Workshop on Peer-to-Peer Systems FEB 21-22, 2003 BERKELEY, CALIFORNIA.

[29] A. Parker, "The true picture of peer-to-peer filesharing," 2004.

[30] MaxMind, "Maxmind geoip country database."

[31] F. MacWilliams and N. Sloane, *The theory of error-correcting codes*. North-Holland Amsterdam, 1977.

[32] J. Douceur, "Is remote host availability governed by a universal law?" *ACM SIGMETRICS Performance Evaluation Review*, vol. 31, no. 3, pp. 25–29, 2003.

[33] M. Steiner, T. En Najjary, and E. Biersack, "Analyzing peer behavior in KAD," *Institut Eurecom*, 2007.

[34] S. Guha, N. Daswani, and R. Jain, "An experimental study of the skype peer-to-peer voip system," in *Proc. of IPTPS*, vol. 6. Citeseer, 2006.

[35] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," *Proceedings of IPTPS02, Cambridge, USA*, vol. 1, pp. 2–2, 2002.

Quality and Performance Optimization of Sensor Data Stream Processing

Anja Klein
SAP Research Center Dresden, SAP AG
Chemnitzer Str. 48
01187 Dresden, Germany
anja.klein@sap.com

Wolfgang Lehner
Database Technology Group, TU Dresden
Helmholtzstrae 10
01069 Dresden, Germany
wolfgang.lehner@tu-dresden.de

Abstract—**Intelligent sensor devices together with data stream management systems allow the automatic recording and processing of huge data volumes to guide any kind of process control or business decision. However, a crucial problem is posed by data quality deficiencies due to imprecise sensors, environmental influences, transfer failures, etc. If not handled carefully, they misguide decisions and lead to inappropriate reactions. In this paper, we present the quality-driven optimization of stream processing that improves the resulting quality of data and service. After an introduction to data quality management in data streams, we define the targeted optimization problem comprising the optimization objectives and parameters that configure the required stream processing operators. Based on the generic optimization framework, we discuss and evaluate the optimization execution in batch and continuous mode. Further, we shed light on the crucial definition of the stream partition length used for the optimization that significantly influences the optimization performance. Finally, we provide a detailed validation of the proposed optimization strategies as well as the scalability of the overall approach not only at artificial data streams, but also using the real-world example of contact lens production monitoring.**

Keywords - Data Quality; Data Stream Processing; Query Optimization; Heuristic Optimization

I. Introduction

Data stream management systems have been developed to process continuous data flows of high data rate and volume. For example, turnover values or sales volume may be streamed from distributed affiliations to the central controlling department to derive management strategies. Further, data streams are recorded in sensors networks to control manufacturing processes or maintenance activities. In this paper we illustrate the quality control in contact lens manufacturing, where lens thickness and axial difference are measured to derive a quality indicator for the production line.

In most applications data stream systems encounter restricted resources, such as limited memory capacity, data transfer capability and computational power. To meet these constraints, data stream volume has to be reduced by processing the streamed information. Data reduction always goes along with a loss of information. Data processing results such as aggregations can only be approximated, so that answers are incorrect or incomplete with respect to the true outcome. Moreover, most data stream sources suffer from limited data quality in multiple dimensions from the beginning. The correctness is decreased for example by restricted sensor precisions, by typos in text elements, or by RFID readers failing to scan an item properly. The completeness of a data stream is reduced whenever a true world event is missed due to sensor or system malfunction. Information and decisions derived from such falsified sensor data are likely to be faulty, too. Therefore, data quality problems have to be handled carefully.

Data quality information, that describe data quality deficiencies due to data sources and/or data processing, can be recorded and transfered in the data stream, e.g., by the quality propagation model (QPM) presented in [1]. It provides a comprehensive framework for data quality management in streaming environments. Only then, data quality information are provided to the user to enable the comprehensive evaluation of imprecise and/or incomplete data stream processing results. Faulty information can be detected to prevent from incorrect decisions. Furthermore, quality information may identify processed data stream results as too insecure to derive any suitable decision. In that case, the data quality has to be improved to re-enable the confident decision-making.

This paper details and extends the quality-driven optimization of sensor data stream processing to improve the resulting data quality, while complying to the given system constraints, that we first published in [2]. Based on data quality information provided in the stream, the data stream operators are configured to maximize the quality outcome to meet user-defined quality requirements. The online tuning is performed continuously in parallel to the traditional data stream processing. The optimal operator configuration is adapted to varying data stream characteristics and changing user-defined requirements on resulting data quality.

For the field of data quality improvement and optimization our contributions are as follows.

- We present the definition of streaming data quality and discuss the conflicts between the embraced data quality dimensions. Further, we identify candidates for the data quality improvement out of a comprehensive set of data stream operators. We discuss derived parameters and their impact on the data quality-driven improvement of the stream processing in Section II.

- In Section III, we propose the optimization framework that adapts the stream processing for data quality improvement. We discuss the optimization execution in batch and continuous mode that supports the determination of the applied stream partition length. Further, we present the data quality improvement task as multi-objective optimization problem and range it in the traditional operations research classification.
- We discuss the specific components of the heuristic optimization algorithm Quality-driven evolution strategy (QES) to solve the data quality optimization problem in Section IV.
- We present a comprehensive evaluation of the presented algorithm in Section V. We analyze the practicability of the proposed optimization framework, validate the influences of different optimization strategies, compare the performance of QES to further optimization heuristics and evaluate the capability at the example of the contact lens production control.

We complete this paper with a discussion of related work in the field of data stream quality, quality improvement and optimization in Section VI and concluding remarks in Section VII.

II. Problem Analysis

In this section, we first define the data quality in data streams to derive objectives for the data-quality driven optimization, which are then discussed in detail. Afterwards, the candidates for data quality improvement are described. First, the sampling rate configuration is illustrated, followed by the interpolation configuration. Then, the configuration impact of group size for aggregation, frequency analysis and filtering as well as of data quality window size is depicted.

A. Data Quality in Data Streams

A data stream comprises a continuous stream of m tuples τ, consisting of n attribute values $A_i (1 \leq i \leq n)$ and the represented time interval $[t_b, t_e]$. To allow for the efficient management of data quality in data streams, we adopt the data quality window approach introduced in [1]. DQ information is not forwarded for each single data item, but aggregated over ω_i data items independent for each stream attribute A_i. The stream is partitioned into κ_i consecutive, non-overlapping jumping data quality windows $w(k)$ $(1 \leq k \leq \kappa_i)$, each of which is identified by its starting point tw_b, its end point tw_e, the window size ω_i and the corresponding attribute A_i as illustrated in Figure 1. Beyond the data stream items $x(j)(tw_b \leq j \leq tw_e)$, the window contains $|Q|$ data quality information q_w, each obtained by averaging the tuple-wise DQ information over the window.

Furthermore, the executed data processing steps have to be tracked in the quality propagation model (QPM) to not lose data quality information. When data streams are aggregated or joined, their data quality information have to be summarized, too. Only then, the data quality path through the processing can be monitored and data quality deficiencies introduced during data stream processing are captured.

Timestamp	…	210	211	212	213	214	215	216	217	218	219	…
CenterThickness	…	0.412	0.403	0.409	0.398	0.392	0.415	0.394	0.410	0.387	0.403	…
Accuracy	…	0.004					0.0039					…
Confidence	…	0.00053587					0,00081967					…
Completeness	…	0.9					0.8					…
AmountOfData		1					1					

t_b= 210 t_e= 214 ω = 5 t_b= 215 t_e= 219 ω = 5

Fig. 1. Data stream sample

The probably best-known/most referenced, comprehensive and balanced definition of data quality was presented 1996 by Wang et al. [3]. They distinguish four data quality categories of 15 data quality dimensions incorporating data quality aspects of raw data, data sets as well as user requirements. We define data quality in data streams based on their findings on intrinsic (accuracy, believability, objectivity, reputation) and contextual data quality dimensions (value-added, relevancy, timeliness, completeness, amount of data). Due to high data stream volume and rate, a manual evaluation of data quality is not possible, so that subjective dimensions of representation and accessibility cannot be measured to describe data streams.

While the accuracy describes the systematic error of data stream values resulting from deficiencies in data sources, the believability in context of data streams can be equated with the confidence describing random errors produced by unforeseeable influences (e.g., environmental affects in sensor networks). Objectivity and reputation of automatic data stream sources can be assumed as guaranteed and do not have to be monitored. As value-added and relevancy are aspects subjective to the respective user, the contextual dimensions reduce to timeliness or up-to-dateness, completeness defining the rate of originally measured stream items compared to interpolated ones and the amount of data d of raw data items represented by an aggregation result.

Definition: *The data quality Q of a data stream D is defined by the set of five data quality dimensions: accuracy a, confidence ϵ, completeness c, amount of data d and timeliness u.*

As stated in the introduction, the data quality-driven optimization configures the data stream processing to maximize the above defined stream data quality while guaranteeing the compliance to the restricted system resources. As metric for the resource load, we use the data stream volume V. The less volume is needed, the better the constraints are met. Hence, the minimization of the data volume is admitted to the optimization goals. Finally, from the user's point of view a high data volume also has positive effects. The more details are given in the data stream, the better decisions can be derived. To measure this data granularity, we select the timeframe T represented by one data stream tuple. While raw data depict one point in time, the result of a data stream aggregation represents a larger time interval. The wider the timeframe, the lower is the granularity. To support the detailed evaluation of streaming data the granularity has to be maximized.

Operator	Parameter	a	ϵ	c	d	u	V	T
Projection	—							
Selection	—							
Join	—							
Aggregation	GroupSize l				+		-	-
Sampling	Rate r_{sa}		-	-		+	+	
Frequ. an.	GroupSize l				+		-	-
Filter	GroupSize l							
Algebra	—							
Threshold	—							
	WindowSize ω					+	-	(-)

Accuracy a Completeness c Stream volume V
Confidence ϵ Amount of data d Granularity T
Timeliness u

TABLE I
OPTIMIZATION CANDIDATES FOR QUALITY IMPROVEMENT

Table I summarizes the optimization objectives composed of data quality dimensions, data stream volume, and granularity. To identify candidates for the data quality improvement, we analyze the operator repository of the QPM consisting of traditional data stream operators like join and selection, operators of the signal analysis, which are often applied to sensor data streams, and operators of the numerical algebra like addition or division as well as the threshold control. Besides the operator configurations, we present the size of the jumping data quality windows ω as interesting parameter. Table I shows the impact of a parameter increase: the quality values are either increased (+) or decreased (-).

B. Objectives

This section defines the fitness functions for each objective of the data quality-driven optimization. As the accuracy describes defects or imprecisions of data stream sources, it cannot be improved by any operator configuration. This data quality can be removed from the list of optimization objectives.

The objectives determined above span different value domains. For example, the window completeness constitutes values in the range $0 \leq c_w \leq 1$, while the absolute statistical error in the dimension confidence is unlimited $0 \leq \epsilon_w \leq \infty$. To allow the quantitative comparison of different objectives, we normalize the objective functions to the range $[0, 1]$.

1) Confidence: The confidence illustrates the statistical error ϵ due to random environmental interferences (e.g., vibrations, shocks) defining the interval $[v-\epsilon; v+\epsilon]$ around the data stream value v containing the true value with the confidence probability p. ϵ is defined by the $(1-p/2)$-quantile α and the data varianceσ^2 of each data quality window w. For example, for $p = 99\%$ the initial confidence of a data quality window including the lens thickness measurements

$$\{0.396mm, 0.428mm, 0.412mm, 0.379mm, 0.403mm\} \quad (1)$$

is set to $\epsilon_w = \alpha \cdot \sigma = 2.58 \cdot 0,000286mm = 0,0007224mm$.

The average statistical error over all data stream attributes has to be minimized to maximize the data quality confidence. The objective function is normalized by division with the maximal statistical confidence error ϵ_{max} in the stream. The objective f_ϵ is defined as follows.

$$f_\epsilon \quad : \quad min \qquad \frac{1}{n \cdot \epsilon_{max}} \sum_{i=1}^{n} \frac{1}{\kappa_i} \sum_{k=1}^{\kappa_i} \epsilon_w(k) \qquad (2)$$

2) Completeness: The completeness addresses the problem of missing values due to stream source failures or malfunctions. Multiple estimation or interpolation strategies exist to deal with missing values in ETL processes and data cleansing [4]. We apply the linear interpolation as compromise between the quality of value estimation and computational capacity. The data quality dimension completeness c is accordingly stated as the ratio of originally measured, not interpolated values compared to the size of the analyzed data quality window.

For example, the sensor for axial difference misses the lens at timestamp $t =' 237'$. To nevertheless derive the quality indicator, the missing value is computed as follows.

$$ax('237') = \frac{1}{2} \cdot (ax('236') + ax('238')) \qquad (3)$$

To note the sensor failure, the completeness of the data quality window $['230','239']$ is set to $c_w = 0.9$.

To conform with the objective above, the objective of maximal completeness is transformed to the minimizing problem f_c, which minimizes the ratio of interpolated data items. Here, no normalization is required as the domain $[0, 1]$ is already provided by the completeness definition.

$$f_c \quad : \quad min \qquad \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\kappa_i} \sum_{k=1}^{\kappa_i} (1 - c_w(k)) \qquad (4)$$

3) Amount of Data: The amount of data determines the set of raw data x used to derive a data stream tuple $y = f(x)$. The higher the amount of data, the more reliable is the processed information. To eliminate outliers to derive statistically stable information of the production line quality, the quality indicator is averaged over a certain set of contact lenses. Thereby, the aggregation group size $l = 20$ leading to $d = 20$ produces more reliable results than $l = 2(d = 2)$.

To transform the objective of maximal amount of data to a minimization problem, we calculate the difference to the highest possible amount of data $d = m$, that comprises the complete data stream. The maximum m serves at the same time as normalization weight.

$$f_d \quad : \quad min \qquad \frac{1}{n \cdot m} \sum_{i=1}^{n} \frac{1}{\kappa_i} \sum_{k=1}^{\kappa_i} (m - d_w(k)) \qquad (5)$$

4) Timeliness: The timeliness is defined as difference between tuple timestamp $t(j)(1 \leq j \leq m)$ and current system time $clock$. For example, the timeliness of $ax('13:26:54')$ at $'13:28:10'$ is $u = 76s$. To maximize the data quality dimension timeliness, the average tuple age normalized by the maximum age u_{max} has to be minimized.

$$f_u : \quad min \qquad \frac{1}{m \cdot u_{max}} \sum_{j=1}^{m} u(j) \qquad (6)$$

$$= \qquad \frac{1}{clock - t_{min}} \cdot \left[clock - \frac{1}{m} \sum_{j=1}^{m} t(j) \right] \quad (7)$$

5) Data Stream Volume: The data stream volume V defines the number of transfered data values in n stream attributes over m data tuples. Besides, the transfered data quality information have to be incorporated. The additional volume is computed based on the number of transfered data quality dimensions Q_i per attribute A_i and the average data quality window size $\bar{\omega}_i$.

$$V \quad = \quad m \cdot (n+1) + \sum_{i=1}^{n} \frac{m}{\bar{\omega}_i} \cdot |Q_i| \qquad (8)$$

The average volume of a data stream tuple of the contact lens stream described with $|Q_i| = 4$ dimensions and an average data quality window size of $\bar{\omega}_i = 20$ results in $V = (4+1) + 1/20 \cdot 4 = 5.2$. To normalize the stream volume to the data range $[0, 1]$, we refer to the maximal stream length $m_{max} = r_{max}/r \cdot m$ determined by the current stream rate r and the maximal manageable rate r_{max}(e.g., $r_{max} = 1/ms$). The maximal data volume further depends on the maximal data quality window size $\omega = 1$, such that

$$V_{max} \quad = \quad m_{max} \cdot (n+1) + m_{max} \cdot \sum_{i=1}^{n} |Q_i|. \qquad (9)$$

To minimize the costs for data stream transfer and processing, the normalized data stream volume has to be minimized.

$$f_V \quad : \quad min \qquad \frac{V}{V_{max}} \qquad (10)$$

6) Granularity: The data stream granularity T is measured as the average timeframe $[t_e - t_b]$ of all data stream tuples. For example, the timeframe of the averaged quality indicator with $l = 20$ constitutes in $T = 20s$. For raw data items describing one point in time, the granularity equals 0, as $t_e = t_b$. To maximize the granularity, the average timeframe normalized by its maximum T_{max} has to be minimized.

$$f_T \quad : \quad min \qquad \frac{1}{m \cdot T_{max}} \sum_{j=1}^{m} t_e(j) - t_b(j) \qquad (11)$$

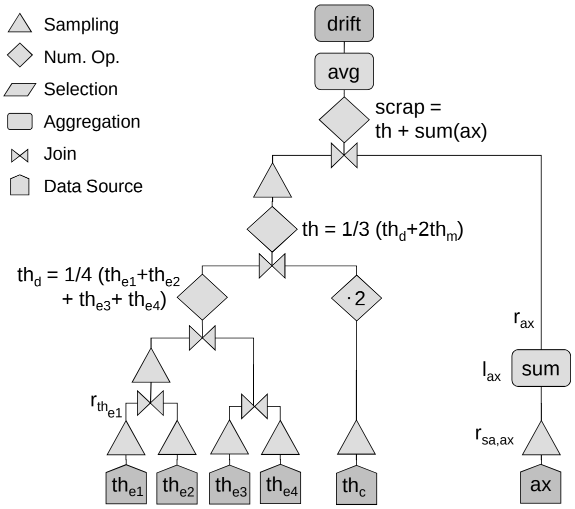


Fig. 2. Processing tree

C. Configuration Parameters

The analysis of typical data stream operators identified sampling, aggregation and frequency analysis as configuration candidates for the data quality-driven optimization. This section first defines the dimension of the optimization problem domain. Afterwards, the impact of the configuration parameters of the identified candidates are discussed.

Figure 2 shows the processing graph of the contact lens application. The goal is to monitor the overall production quality, i.e., the fraction of contact lenses that have to be removed from the production line as scrap, to predict the next maintenance date. Therefore, four thickness measurements th_e at the lens edge are summarized and added to the weighted center thickness th_c. All sensor streams are sampled up $r_{sa} > 1$ or down $r_{sa} < 1$ to reduce the overall data volume and adjust the data stream rates in case of missing data tuples. Then, the axial difference stream ax is aggregated to sum up potential errors. The measurements are combined to determine the scrap indicator $scrap$. If $scrap < 0.5$ the contact lens does not fulfill the required quality levels and has to be removed from the production. To produce stable monitoring results, individual scrap indicators are averaged in sliding windows. Finally, the drift of the scrap fraction, i.e the drift of the lens production quality, can be used to guide the maintenance planning of the production line.

To optimize the resulting data quality, all sampling rates and group sizes can be modified. For example, the contact lens scenario comprises $|sam| = 8$ sampling operators and $|agg| = 3$ aggregations. If applied in the data stream processing, also the group size of performed frequency analyses fre can be optimized. The size of the data quality windows constitutes the last optimization parameter, that can be determined independently for each data source $s \in S$. The contact lens quality requires $|S| = 6$ sensors, adding 6 parameters to the optimization problem. The overall problem optimizes $dim = |sam| + |agg| + |fre| + |S|$ parameters, e.g., $dim = 8 + 3 + 0 + 6 = 17$, defining the dimension of the problem domain.

1) Sampling Rate: The down-sampling reduces the data stream volume by randomly skipping a given set of data items defined by the sampling rate $r_{sa} < 1$. The information loss provoked by down-sampling represents a statistical error and has to be captured in the dimension confidence. Reconsider the data quality window presented above in Equation 1 with the true average $avg = 0.4mm$ sampled with $r_{sa} = 0.5$. The sample average ranges from $avg = 0.386mm$ to $avg = 0.414mm$ corresponding to a of $e^+ = 0,0143mm$.

Definition: *The statistical error due to down-sampling can be estimated based on the confidence interval defined by Haas in [5], such that*

$$\epsilon_+ = \frac{\alpha \cdot \sigma(w)}{\sqrt{\omega \cdot r_{sa}}} \cdot \sqrt{1 - r_{sa}}, \tag{12}$$

where $\sigma^2(w)$ states the variance of data stream values in the respective DQ window and α describes the confidence probability p as the $(1 - p/2)$-quantile.

Low sampling rates skip large data stream parts resulting in a high statistical error due to higher information loss. The confidence error rises with decreased sampling rate. On the contrary, low down-sampling rates reduce the data stream volume. The objectives of minimized confidence ϵ and minimized stream volume V conflict with each other. As timeliness and data stream volume are aligned due to higher processing times for large tuple numbers, the objectives of minimal timeliness u and minimal confidence ϵ conflict with each other in the same manner.

The up-sampling ($r_{sa} > 1$) inserts data items into the data stream. For example, a linear data interpolation is executed with the rate $r_{sa} = 2$, which doubles the data stream length. Hence, the up-sampling increases the fraction of interpolated data and has to be tracked by updating the DQ dimension completeness.

Definition: *During up-sampling, the window completeness c_w is divided by the sampling rate $r_{sa} > 1$, such that the completeness is stated as $c'_w = c_w / r_{sa}$.*

The higher the up-sampling rate, the more data items are generated. The lower is the resulting completeness and the higher the data stream volume. Hence, high up-sampling rates have a negative impact both on completeness c and data stream volume V.

As described above, the relation of stream rates has to remain constant for each processing tree level. For example, the up-sampling in data stream th_c increases the down-sampling rate for th_{e1-e4}. The up-sampling has an indirect positive impact on the statistical error in the data quality dimension confidence. The objective of maximized completeness c achieved by low sampling rates r_{sa} contradicts with the minimization of the confidence ϵ resulting from high r_{sa}.

2) Group Size: First and foremost, the group size l constitutes a parameter in the definition of data processing queries. For example during the calculation of the turnover, it is essential whether the sales volumes of one day ($l = 1d$) or one month ($l = 30d$) are summed up. However, during the processing of high-volume data streams situations can arise, where the group size must not be defined strictly.

The group size of aggregation and frequency analysis[1] share the same impact on the optimization objectives, as they summarize stream tuples to a smaller data volume. The larger their group size is defined, the more the data stream volume is reduced.

The more tuples are aggregated or serve as basis for frequency analysis, the larger is the amount of data of outcoming results. As the aggregation reduces the data volume, maximal amount of data goes along with minimal data stream volume. However, the summarization of information has a negative impact on the data stream granularity, as one processing result represents the timeframe spanned by all incoming data tuples in the respective group. The larger the group size, the larger is the resulting timeframe. During the configuration of the group size l positive effects on data stream volume V and amount of data d oppose negative effects on the granularity T.

The objective conflict can be resolved by configuring the data stream rate with the help of sampling and/or interpolation. During group sizes increase, the same stream rate increase leads to consistent amount of data. However, the consistency of optimal stream volume V, amount of data d and granularity T interferes with the objectives of minimized confidence ϵ and maximized completeness c as declared above.

3) Window Size: The definition of the size of data quality windows constitutes a compromise between fine granularity and high volume of transfered data quality information.

The wider the data quality window, the lower is the overhead for the data quality transfer and, thus, the lower is the overall stream volume. On the contrary, the larger the data quality window is defined, the more tuple-wise DQ information have to be aggregated in one window-wise quality value balancing out meaningful data quality peaks.

Therefore, we add the window size ω itself as optimization parameter. The objective function f_ω defining the averaged window size $\bar{\omega}_i$ over all data quality windows and stream attributes normalized with the maximal possible window size $\omega = m$ has to be minimized.

$$f_\omega \quad : \quad min \qquad \frac{1}{n \cdot m} \sum_{i=1}^{n} \bar{\omega}_i \tag{13}$$

As more data quality information have to be transfered and processed, the objective of low window sizes ω contradicts with minimal data stream volume V and maximal timeliness u.

[1] The frequency analysis computes amplitude and phase of all stream inherent frequency bands.

	ϵ	c	d	u	V	T	ω
ϵ	-	r_{sa}		r_{sa}	r_{sa}, l	r_{sa}, l	
c	r_{sa}	-	r_{sa}	r_{sa}	r_{sa}, l	r_{sa}, l	
d		r_{sa}	-			r_{sa}, l	
u	r_{sa}	r_{sa}		-		l	ω
V	r_{sa}, l	r_{sa}, l			-	l	ω
T	r_{sa}, l	r_{sa}, l	r_{sa}, l	l	l	-	
ω				ω	ω		-

TABLE II
OBJECTIVE CONFLICTS

Table II summarizes the conflicts between the optimization objectives. The cells indicate the configuration parameter leading to the respective conflict.

III. Quality-Driven Optimization

In this section, we present the framework architecture for DQ-driven optimization consisting of satisfiability checks and optimization component. To solve the defined problem, we propose the quality-driven evolution strategy.

A. Optimization Framework

The data quality-driven optimization is executed continuously to tune the data stream processing during system runtime. As soon as an optimal parameter set is found and deployed, it has to be checked against the currently processed data stream. The online tuning allows the seamless adaptation to varying stream rates, measurement values and data quality requirements.

First, the system evaluates by means of static information like maximal sensor stream rate or sensor precision, if the user-defined quality requirements can be accomplished or conflicts exclude a realization of all sub-objectives. In the latter case, the conflict is reported to the user. To check the satisfiability of DQ requirements, no access to streaming data is needed.

Heuristic optimization algorithms approximate the optimal problem solution by iteratively improving the achieved fitness. Different solution individuals have to be applied, evaluated and compared.

As the optimization must not interfere with the ongoing data stream processing, it is separated on an independent system component as illustrated in Figure 3. To execute the optimization in parallel with the traditional data stream processing, the processing path with all its operators is copied in the optimization component. In each iteration of the optimization algorithm the evaluated solution individual determines the specific path configuration of sampling and interpolation rates as well as group and window sizes.

Each solution is evaluated by directing a representative data stream partition through the configured processing path. We propose two approaches for the partition selection in Section III-C. As soon as the partition is completely processed, the average data quality result for each dimension, the average granularity and the used data stream volume are computed and returned to the optimization algorithm to calculate the fitness of the tested configuration. If the user-defined quality requirements are not met, the fitness guides the determination of the next solution individual to iteratively improve the achieved fitness.

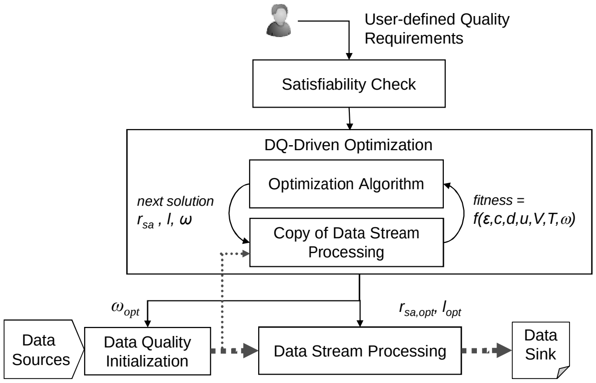

User-defined Quality Requirements
Satisfiability Check
DQ-Driven Optimization
Optimization Algorithm
next solution r_{sa}, l, ω
fitness = f(ε,c,d,u,V,T,ω)
Copy of Data Stream Processing
ω_{opt}
$r_{sa,opt}$, l_{opt}
Data Sources
Data Quality Initialization
Data Stream Processing
Data Sink

Fig. 3. Optimization process

As soon as the requirements are accomplished, the optimization problem is solved. The new parameter setting is applied to the original processing path. The sampling operators are updated with the optimized sampling rates $r_{sa,opt}$. The frequency analyses and aggregations are updated with the determined group sizes l_{opt}. Finally, the data quality initialization at the sensor nodes is re-configured with the new window sizes ω_{opt}. After deploying the found parameter set, the next optimization run is performed to adapt the processing to dynamic streams characteristics.

The logical distinction between optimization and processing enables also the physical separation, for example on a distinct server node. Thus, the optimization task has no negative impact on the performance of the traditional data stream processing.

B. Satisfiability Check

Four quality checks have to be performed to prevent from optimizing against unsatisfiable user-requirements as shown in Algorithm 1.

First, the desired confidence error is evaluated. The best possible confidence is achieved, when no down-sampling is performed. Only initial statistical errors of the sensors reduce the confidence. Thus, the requested confidence objective ϵ_{req} must not be smaller than the square root of all initial confidence errors ϵ_{S_i} (line 1).

Second, only completeness deficiencies due to up-sampling can be reduced by DQ-driven optimization. The completeness objective c_{req} must not exceed the average initial completeness c_{S_i} provided by the sensors (line 2).

The conflicting objectives of minimal data stream volume V, minimal window size ω and maximal amount of data d are compared based on the resulting stream length m. As the amount of data cannot exceed the stream length, it must hold that $d_{req} \leq m$. By applying Equation 8, we derive the satisfiability check shown in line 3.

Finally, the conflict between amount of data d and granularity T is controlled based on the maximal stream rate r_{max} introduced in Section II-B in line 4. For example, the required amount of data of $d_{req} = 100$ and the maximal rate of $r_{max} = 1/ms$ leads to the minimal achievable granularity of $T_{min} = 100ms$.

Algorithm 1: Satisfiability Check

```
Input: $\epsilon_{req}, c_{req}, V_{req}$ user-defined requirements
Output: $sat$=FALSE satisfiability
1 if $\epsilon_{req} \geq \sqrt{\sum_{i=1}^{|S|} \epsilon_{S_i}^2}$
2 $\wedge c_{req} \leq \frac{1}{|S|} \sum_{i=1}^{|S|} c_{S_i}$
3 $\wedge d_{req} \leq \frac{V_{req}}{n+1+\sum_{i=1}^{n} \frac{1}{\omega_{req}} |Q_i|}$
4 $\wedge T_{req} \geq \frac{d_{req}}{r_{max}}$
5 then
6     $sat$ = TRUE;
```

C. Batch vs. Continuous Optimization

The stream partition for optimization constitute a data stream window of ζ data tuples. It may either be selected in batch-mode at the beginning of each optimization run and used in each iteration without changes. At the other hand, the window can be updated for each iteration with current tuples to reflect the dynamic progression of the data stream and allow the continuous optimization.

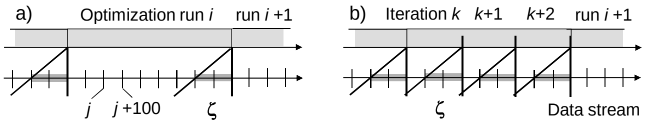


Fig. 4. Batch & continuous optimization

The batch-mode selects a constant stream partition of the last ζ tuples for one complete optimization run as shown in Figure 4a. The parameter ζ depicts the trade-off between representative partition length and duration of one optimization run. Besides, the length is restricted by limited memory capacity in most streaming environments. It only represents a small, static data stream window.

In contrast, the continuous optimization approach follows the dynamic stream behavior by selecting a new stream partition for each iteration of the applied optimization algorithm. As shown in Figure 4b, a larger stream fraction is used. However, the continuous mode exchanges the base data for optimization in each iteration. A good solution derived from the fitness in iteration k on partition k may perform poor if applied to base data $k+1$ in iteration $k+1$. The fitness will not increase monotonically, but may diverge. To guarantee the algorithm termination, additional criteria like allowed duration or number of fitness evaluations have to be defined.

The static optimization in batch-mode guarantees the converging of the fitness function and, thus, the accomplishment of user-defined quality requirements. However, the computed optimal solution only holds for the processed short stream partition, so that a permanent control with subsequent optimization runs is necessary. The continuous approach allows the inclusion of dynamic data stream alterations during the optimization process. Thereby, divergences of the fitness function may arise that must be encountered by supplementary terminations rules.

D. Optimization Classification

The operations research combines mathematics and formal science to find and define methods and algorithms to arrive at optimal or near optimal solutions to complex problems. Optimization problems are classified according to their domain, their objective function and respective problem constraints. In this section, we will range the optimization problem of data quality-driven stream processing in this classification scheme.

The problem analysis revealed seven objectives, which partly conflict with each other and define a multi-objective optimization problem. There exist various strategies to solve these problems. On the one hand, the Pareto optimization allows the finding of a set of optimal compromises, that dominate all other possible solutions. That is, the improvement of one sub-objective is only possible with a decline of one or more of the other sub-objectives. In Section IV we define the multi-objective quality-driven evolution strategy (MO-QES), which optimizes the Pareto front in the problem search domain by solving the objective f_{multi}.

Definition: *The multi-objective fitness function* f_{multi} : $\mathbb{R}^{dim} \mapsto \mathbb{R}^7$ *defines the fitness of a solution individual as the Pareto-dominance of the achieved sub-objective* $f_i(i \in \{\epsilon, c, d, u, V, T, \delta\})$.

On the other hand, sub-objectives can be summarized in one optimization function f_{single}, which will be minimized or maximized by the single-objective quality-driven evolution strategy (SO-QES). Cost weights defined for each objective determine the order and importance of optimization. However, the user-defined weighing may restrict the search space resulting in minor optimization solutions. Multiple optimization runs are required to determine optimal cost weights.

Definition: *The single-objective fitness function* f_{single} : $\mathbb{R}^{dim} \mapsto \mathbb{R}$ *calculates the overall fitness of a solution individual as weighted sum of the obtained sub-objectives, such as*

$$f_{single} = \sum_{i \in \{\epsilon, c, d, u, V, T, \delta\}} c_i \cdot f_i \quad (14)$$

The data quality dimensions completeness, amount of data and timeliness are defined using linear computation functions, so that f_c, f_d and f_u pose linear optimization problems.

The same holds for the data quality window size f_ω and the granularity f_T. Due to the square root definition of the statistical error computation, the objective function f_ϵ gives a non-linear optimization problem. A second non-linear problem is posed by f_V, which minimizes the stream volume based on the window size ω.

While the parameters l and ω as well as fitness of the sub-objectives f_d, f_u, f_V, f_T and f_ω allow only discrete values, the domains of the parameters r_{sa} and the objective functions f_ϵ and f_c propose continuous optimization problems.

Moreover, the search space is restricted by side conditions. To guarantee join partners, the data stream rates in one tree level have to stay in constant relation to each other. For example, the duplication of $r_{th_{e1}}$ (see Figure 2) requires i.a. the duplication of r_{ax}. Either, the rate of the sampling operator $r_{sa,ax}$ has to be doubled, or the group size l_{ax} has to be halved. Thus, the dependence of sampling rates and group sizes defines side constraints for each processing tree level, such that

$$\forall\ level\ l \quad \forall\ r_i, r_j \in l : r_i = r_j. \tag{15}$$

As in multi-objective optimization problems the most complex sub-objective defines the complexity of the overall problem, the quality-driven optimization is defined as follows.

Definition: *The quality-driven process optimization constitutes a multi-objective, non-linear, continuous optimization problem with side conditions.*

There exists no deterministic algorithm to solve optimization problems of this complexity in reasonable time. However, the stream processing optimization shall be executed on-the-fly without interrupting the data flow. It is essential to provide good solutions in an acceptable timeframe. Further, the optimal solution is not required in most cases. Rather, the user or data consuming application defines data quality levels, which have to be met. Heuristic algorithms offer an appropriate answer to such optimization problems. They provide fast results by approximating the optimal solution.

Heuristic optimization algorithms range from simple approaches like the Monte-Carlo-Search over more sophisticated strategies such as Hill Climbing and Simulated Annealing to complex Evolutionary Algorithms. Evolutionary Algorithms comprise genetic algorithms working on binary data and the evolution strategy supporting real parameters and objectives. As only the evolution strategy can solve real-valued multi-objective as well as single-objective problems, we apply this heuristic to solve the DQ-driven optimization.

IV. Evolution Strategy

The evolution strategy constitutes a stochastic, population-based search heuristic inspired by the principles of natural evolution. It can be applied to arbitrary optimization problems and requires nothing but the objective function(s) of the optimization problem to guide its search. A population of possible solutions is iteratively recombined and mutated to select the best population individual as approximated global optima.

To improve the data quality via configuration of the data processing, we have to adapt the generic algorithm structure of the evolution strategy to the defined quality-driven optimization problem. In this section, we present the specification of the quality-driven evolution strategy (QES) including specific functions for recombination, mutation and selection.

Algorithm 2: QES

```
  Input: $domain$ of possible inputs,
  $DQ$ user-defined requirements
  Output: $P$ population of optimal solutions
1 $t$ = 0;
2 initialize($P(t)$, $domain$);
3 while $DQ.notAchieved()$ do
4     $P_c(t)$ = recombine($P(t)$);
5     mutate($P_c(t)$);
6     $P(t+1)$ = selectNextGeneration($P_c(t), P(t)$);
7     $t$ = $t$ + 1;
8 end
```

Algorithm 2 shows the overall structure of QES. The first population $P(0)$ is randomly initialized in line 2 to cover the complete search $domain$. The first step of the repeated iteration process recombines the solution individuals of the current population $P(t)$ in line 4 to build new solution candidates $P_c(t)$. They are randomly mutated to allow new solutions to enter the population in line 5. Finally, they are evaluated with the help of the objective function f_{single} or f_{multi}, respectively, to select the best individuals of $P_c(t)$ to build the next generation $P(t+1)$ in line 6. The quality-driven evolution strategy terminates, when all user-defined data quality requirements are met (line 3). Other termination criteria are the allowed number of performed solution evaluations or the planned execution time.

The *recombination* is designed to hand down and combine positive traits of different solutions individuals. First, parent solutions are chosen randomly from the current generation. Then, the dim parameter configurations of parent pairs are combined. The children's parameters are determined by averaging each of the parents' parameters. As group and window sizes only allow integer values, they are rounded up.

Algorithm 3 illustrates the *mutation* of the recombined solution candidates. Due to different value domains, specific mutation steps sizes are applied to each parameter type: sampling and interpolation rates $\Delta r_{sa/in}$ in line 4, group size Δl (line 6) and window size $\Delta\omega$ (line 8). To ease the application of the QES and to allow for the automatic adaptation, the specific step sizes are considered as additional optimization parameters. The parameter vector is extended by three variables: $dim' = dim + 3$. The step sizes themselves are mutated by as shown in line 10.

The mutation is executed individually for each solution of

```
Algorithm 3: mutate()
   Input: $P$ current population
   Output: $P_m$ mutated population
1  forall $a \in P$ do
2      $index = random(1, dim + 3)$;
3      $oldValue = a$.getParameterAt($index$);
4      if type(index) = sampling || interpolation then
5          $newValue = oldValue \pm \Delta r_{sa/in}$;
6      else if type(index) = groupSize then
7          $newValue = oldValue \pm \Delta l$;
8      else if type(index) = windowSize then
9          $newValue = oldValue \pm \Delta\omega$;
10     else
11         $newValue = oldValue \pm 0.1 \cdot oldValue$ ;
12     $a$.setParameterAt($index$, $newValue$);
13 end
```

the current population. The parameter to mutate is selected randomly from the configuration set (line 1 & 2) and mutated according to the respective step size (lines 3-10). The new solution individual b is created by exchanging the mutated parameter in the current solution a (line 11).

The last algorithm step, the *selection*, evaluates the new solution candidates to form the next population generation. The quality-driven evolution strategy follows the $(\mu + \lambda)$-approach, that produces a monotonically nondecreasing fitness curve. One population consists of μ elements, which are used to produce λ candidates. The fitness of all parent and child solutions is calculated with the help of the objective function. The μ fittest solution individuals build the new generation as starting point for the next algorithm iteration.

The single-objective quality-driven evolution strategy (SO-QES) uses the cost-weighted objective function f_{single}. The optimization problem defined in f_{multi} is solved by the multi-objective evolution strategy (MO-QES). The comparison of optimization results allows conclusions on the impact of different cost settings. Furthermore, the following section evaluates the performance of the two optimization approaches.

V. Evaluation

In this evaluation, we examine data quality-driven optimization of the data stream processing at real-world data streams to empirically answer the following questions.

1) Which impact has the cost weighing on the optimal configuration of the data stream processing?
2) What are the benefits of batch and continuous optimization?
3) How do single- and multi-objective optimization compete with each other?
4) Do the presented algorithms scale for the complex data stream processing with high sensor numbers?

We have implemented the quality-driven optimization described in this paper using the data stream management system PIPES [6] and the Java-based optimization framework JavaEva [7].

A. Experimental Setting

We ran our experiments on a dual-core 2x2GHz Centrino Duo processor with 2GB of main memory, running Microsoft Windows XP Professional 2002. All Java-based systems were executed using JRE Version 6.

We use an artificial dataset to analyze the effects of the optimization modes and test the performance of the designed algorithms. Therefore, we generated data streams subject to the standard normal distribution ($\mu = 0, \sigma = 1$) with randomly varying stream rates in the range of $1/ms \leq r \leq 100/ms$. We simulated queries over 2 to 128 of such generated streams joined in pairs of two. Each data stream is sampled in each query tree level to find one-to-one-join partners; aggregations are spread randomly.

Further, we applied the real-world dataset of contact lens manufacturing available at [8] to analyze the impact of cost weights and to examine the practicability of the presented algorithm. It consists of measurements of the thickness of lens center and edge and the axial difference. As described in Section II-C, the production quality is monitored to predict the optimal maintenance planning for re-calibrating the production line.

For both datasets, we assumed a systematic error of optimistic 1%, while the statistical measurement error was derived from the measurements' variance using the confidence probability $p = 99\%$. To simulate sensor failures in the artificial dataset, we randomly skipped 2% of the data tuples. To initialize the lens data completeness, we identified missing measurements by comparing the recorded timestamps to the planned sensor rates.

B. Impact of Weights

This section answers the first of the above questions at the sample objectives of maximal completeness and minimal confidence (compare Table I). Figure 5 shows the Pareto front of the multi-objective optimization. Minimal statistical confidence errors produced by high sampling rates are only achieved at the expense of high values of incompleteness and vice versa. The optimal compromises represented by the Pareto front can be re-produced by the single-objective optimization using sophisticated weighing.

Points A,B and C in Figure 5 show exemplary single-objective results. The higher the weight was determined, the better the proposed optimal configuration suits the respective sub-objective. If the cost for incomplete data tuples exceeds the confidence weight, low sampling and interpolation rates are proposed (point C). High costs for statistical errors in the DQ dimensions confidence (point A) lead to high sampling rates resulting in less data loss. Similar cost weights result in a well-balanced compromise between completeness and confidence as given in point B.

Figure 6 illustrates the Pareto front and cost impact of the conflicting objectives maximal amount of data and maximal

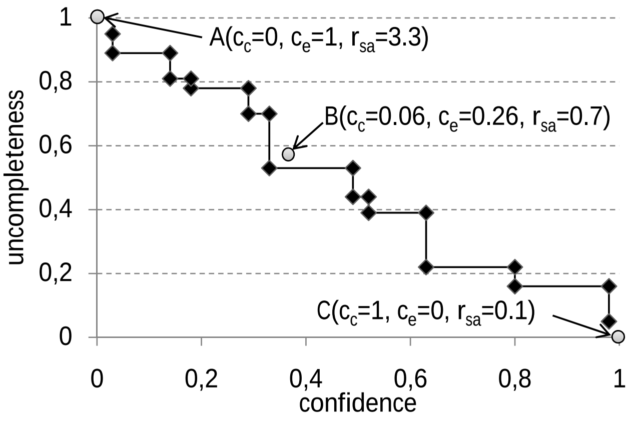


Fig. 5. Confidence vs. completeness

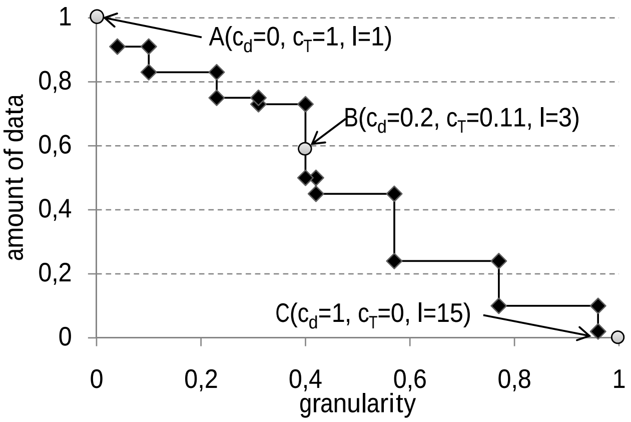


Fig. 6. Amount of data vs. granularity

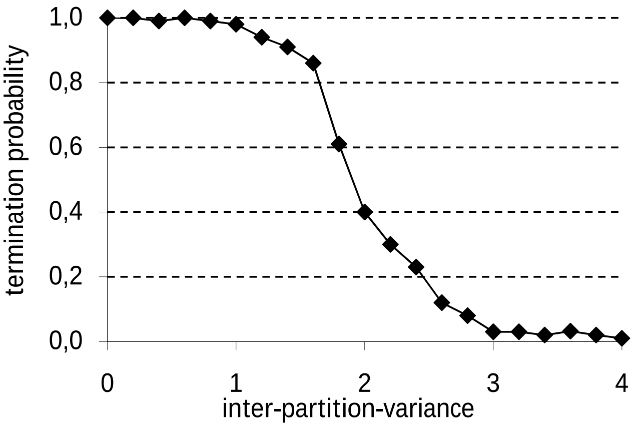


Fig. 7. Termination probability

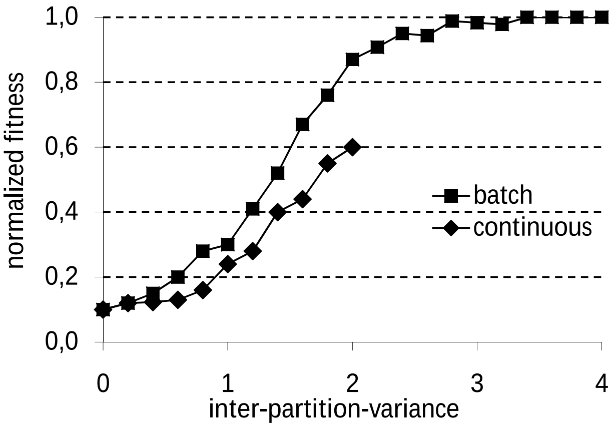


Fig. 8. Normalized fitness

granularity. High amount of data leads long timeframes represented by each tuples, i.e., low granularity. The higher the cost weight for the amount of data exceeds the cost for fine granularity, the higher is the average group size in the proposed optimization solution and vice versa (compare weights of points A, B and C in Figure 6).

Finally, we evaluated the conflicting objectives of minimal data stream volume and minimal data quality window size. Similar to the above evaluations, higher weighing of the stream volume results in low window sizes and vice versa.

C. Comparison of Optimization Modes

This section answers the second question. First, we check the termination probability of the continuous optimization, which selects a new data stream partition as basis for each optimization iteration. Therefore, we applied the artificial data set described above to the single-objective optimization. We kept the intra-partition-variance ($\sigma = 1$), but introduced an inter-partition-variance $\breve{\sigma}$ by modifying the mean μ for each partition.

Figure 7 shows the termination probability of the single- and multi-objective optimization for an increasing inter-partition-variance. For $\breve{\sigma} < 1$, the continuous optimization is likely to terminate successfully. The termination probability decreases for $1 \leq \breve{\sigma} \leq 3$ and converges to 0 for $\breve{\sigma} > 3$. The termination probability is independent from the applied partition length.

To compare the quality of the optimization results provided by batch and continuous mode, we apply the computed optimization results, i.e., the operator configurations, to the ongoing stream and compare the achieved overall quality. Figure 8 shows the normalized fitness for an increasing inter-partition-variance. The batch mode allows good results for low variances $\breve{\sigma} \leq 0.5$. The continuous mode adapts better to changing situations and thus provides appropriate fitness values also for a higher inter-partition-variance. However, the increasing termination probability limits the application of the continuous mode to the bound $\breve{\sigma} \leq 2$. Here, a small DQ improvement is only possible by using batch mode again.

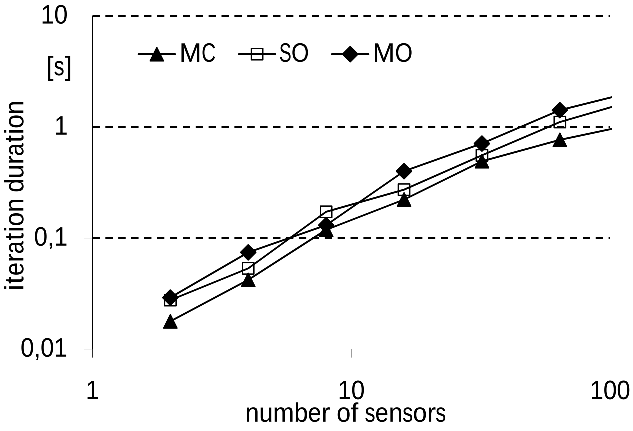


Fig. 9. Iteration duration vs. number of sensors

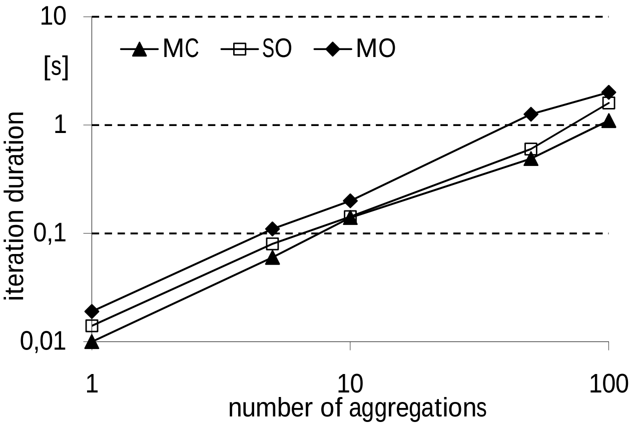


Fig. 10. Iteration duration vs. number of aggregations

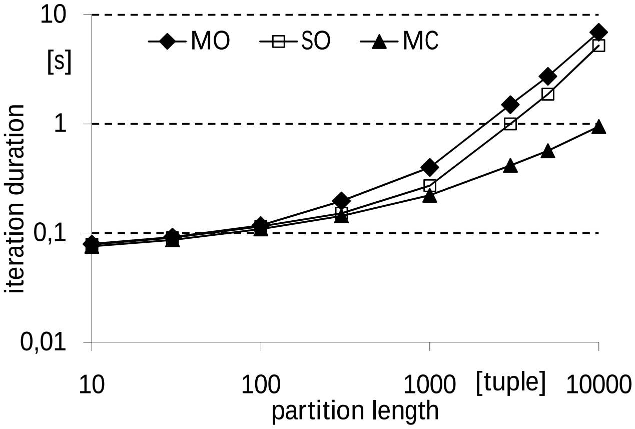


Fig. 11. Iteration duration vs. partition length

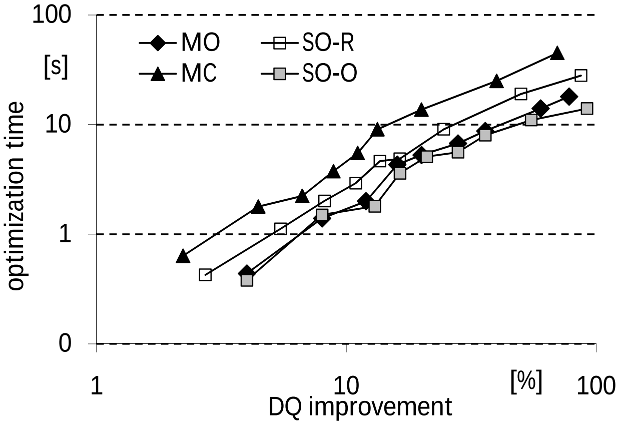


Fig. 12. Duration of DQ improvement

D. Performance Tests

In this section, we show the scalability of the presented algorithms by means of the artificial data set and complex query structure defined in Section V-A. As no inter-partition-variance was introduced, the evaluation tests were performed in batch-mode. We first analyze the performance of the proposed algorithms with regard to increasing numbers of data sources (sensors), applied aggregations and frequency analyses, respectively. As the number of sampling operators is defined by the executed joins and, thus, by the number of sensors, an individual scalability test for that operator class is not required. Then, we evaluate the impact of the length of the data stream partition used for optimization. Finally, we compare the optimization time required by the single- and multi-objective optimization strategy to improve the overall quality.

The Monte-Carlo-Search (MC) performs a random search over the problem domain and serves as reference value defining the lower performance bound [9]. The single-objective optimization is executed with randomly chosen weights (SO-R) as well as optimal weights (SO-O), which determine a well-balanced objective compromise. As the iteration duration of the single-objective optimization does not depend on the weights, these performance test results of SO-R and SO-O have been summarized to SO. Finally, the multi-objective optimization (MO) approximates the Pareto front of all optimal compromises.

Figure 9 illustrates the impact of the sensor number on the time required for one iteration of the respective optimization algorithm executed with a partition length of 1000 tuples. The more complex the algorithm, the longer takes one iteration. The performance difference between single- and multi-objective optimization is caused by the complex Pareto front computation. For all tested algorithms, the iteration duration increases linearly with the sensor number.

Figure 10 shows the scalability for increasing numbers of aggregations and frequency analyses. Again, the time required for one iteration rises linearly.

Figure 11 shows the iteration duration (in seconds) for 16

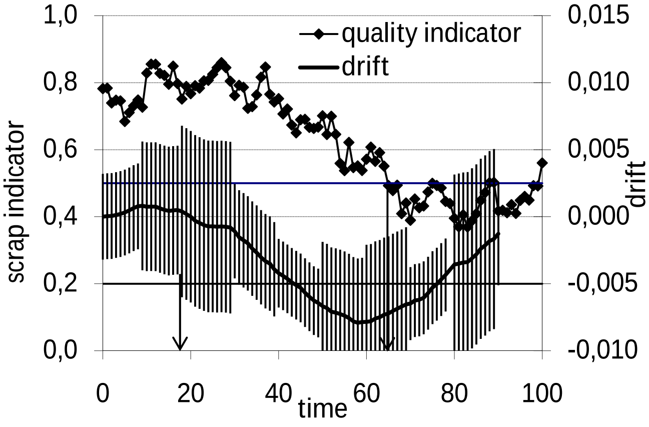


Fig. 13. Lens quality control before optimization

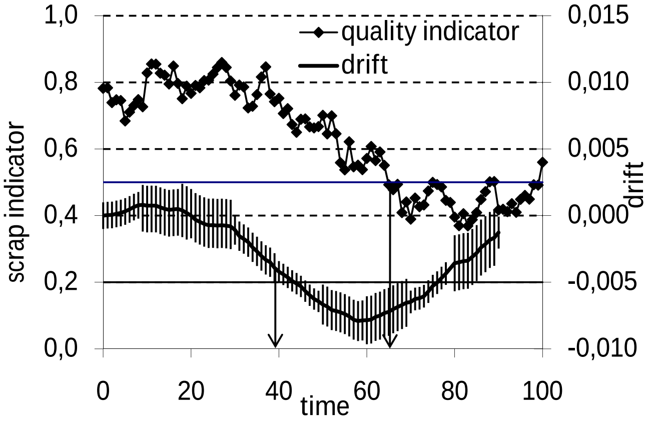


Fig. 14. Lens quality control with MO

sensors and 10 aggregations for increasing partition lengths (in data tuples). The processing time rises nearly linearly for small to medium partition lengths of 100 to 1000 stream tuples. Only for very large stream partitions, the iteration duration exhibits an exponential character.

Figure 12 compares the overall time performance of single- and multi-objective optimization with respect to the achieved quality improvement for 16 sensor data sources and 10 randomly inserted aggregations. The quality improvement is expressed as percentage value $(q_{before} - q_{after})/q_{before}$. The Monte-Carlo-Search performs worth followed by the randomly initiated single-objective optimization (SO-R), requiring 5.2 and 2.9 seconds, respectively, for a DQ improvement of 10%. The single-objective optimization executed with well-balanced weights (SO-O) performs best (1.6s for 10%). However, the definition of these weights requires multiple optimization runs and has to be adapted as soon as stream characteristics or user requirements change. The multi-objective optimization (MO) is a little slower (1.9s) due to the complex computation of the Pareto front. However, the result comprises the complete set of all optimal compromises and no pre-processing to determine optimal weights is necessary.

The evaluation showed, that the designed quality-driven optimization provides good scalability with regard the applied stream partition length as well as increasing complexity of the data stream processing. Data quality and quality of service could be improved within few seconds. Further, we deduce that the single-objective optimization in batch mode is the best choice for constant user requirements and steady data streams. If streaming data values present high fluctuations or user requirements are often adjusted, the multi-objective optimization constitutes the better option. Here, the inter-partition-variance has to be analyzed to estimate the termination probability. While the continuous mode provides better results for medium variances, the batch mode has to be applied for highly varying data streams to prevent from divergent fitness functions.

E. Lens Production Optimization

To approve the suitability of the designed algorithms, we refer to the application scenario of scrap monitoring for predictive maintenance in the contact lens production introduced in Section II-C. Due to numerical errors and missing sensor measurements, the predicted maintenance date deviates from the optimal point in time. The maintenance is scheduled either too early (the calculated quality drift exceeds the true value), or too late (the drift of the computed quality indicator is too low).

To improve the reliability of the determined maintenance planing, we aim to optimize the underlying data stream processing. The first optimization parameters are the sampling rates on the data streams of the lens thicknesses and axial difference. The group sizes of the axial difference summation and sliding average aggregation of the quality identifier determine the compromise of detailed information and outlier balancing. The last optimization criteria is given by the group size of the drift calculation that provides statistically stable short- or long-term variations.

We start the optimization process with arbitrary settings of the optimization parameters. Figure 13 illustrates the scrap indicator of the contact lenses and the derived drift of the production quality. To guarantee correct maintenance activities so that no lens under the scrap threshold of 0.5 remain in the production line, the drift is monitored against the drift threshold of -0,005. As the calculated drift suffers from measurement errors and uncertainties due to sensor failures, the lowest possible bound of the drift must be taken into account. Therefore, the absolute measurement error (the sum of systematic and statistical error) is illustrated as error bars of the drift function. The arrow indicates the resulting maintenance time at $t = 19$. However, due to the high measurement error before the optimization, this maintenance date is too early considering the actual scrap indicator, that falls below 0.5 at $t = 64$.

Figure 14 shows the same situation after 10 iteration of the multi-objective optimization of the data stream processing has been executed. The absolute measurement errors have been decreased significantly, such that the "'lost"' production time could be reduced by starting the maintenance only at $t = 39$.

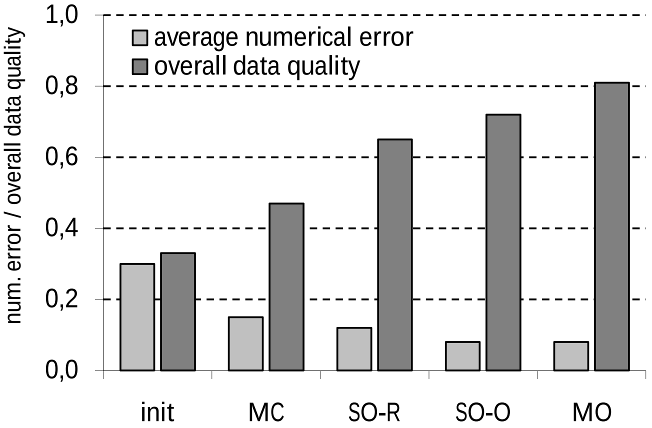


Fig. 15. Improvement of overall DQ & num. error

Figure 15 shows the overall data quality and the numerical error after 100 iterations for each of the presented optimization heuristics, averaged over 10 optimization runs. Already the simple random Monte-Carlo-Search halves the numerical error defining the uncertain range of the quality drift. The randomly initiated single-objective optimization (SO-R) further decreases the error by 30%. The best results are obtained by SO-O and MO. SO-O achieves an overall error reduction of 73%. MO provides a set of optimal compromise solutions: as all sub-objectives shall be improved, Figure 15 illustrates the error reduction (74%) for the well-balanced compromise of weights: $c_c = 0.06, c_\epsilon = 0.26, c_d = 0.2, c_T = 0.11, c_V = 0.2, c_\omega = 0.11$ (compare point B in Figure 5 and 6).

After the data quality-driven optimization, the numerical errors are decreased leading to a narrower uncertainty range. The determined maintenance planning approximates the optimal point in time with higher confidence. Premature maintenance, that unnecessarily interrupts the contact lens production, as well as too late activities, that risk loss of sales due to low production quality, are both prevented.

VI. Related Work

In this section, we will discuss related work in the fields of data quality management and optimization methods, especially focusing on multi-objective optimization. This paper gave an extended view over the quality-driven optimization of sensor data stream processing, that we first published in [2]. Besides detailed definitions of the objective functions as well as optimization parameters, we added the discussion of the batch and continuous optimization strategy as well as the crucial definition of the stream partition length used for the optimization. Further, the evaluation was deepened to show the influences of different optimization techniques as well as the scalability of the overall approach not only at artificial data streams, but also using the real-world example of contact lens production monitoring.

Traditional approaches of query optimization in database and data stream systems aim at minimal processing time and maximal data throughput. The quality of service is improved to provide fast processing results to the user. In this paper, we address the opposite problem: we improve the quality of data. We maximize the data quality of processing results under consideration of restricted system resources, so that the quality of service remains in acceptable ranges.

Multiple publications underline benefits of the data quality management in data warehouses and databases [10][11]. To define the term data quality, different sets of data quality dimensions are discussed i.a. in [12] and [3]. While there are different approaches to structure data quality metadata in databases [13][14], [1] presents the first data quality model suitable for data streaming environments using so called jumping data quality windows.

Data quality improvement in the context of data warehouse and information systems is achieved by data cleaning [15][16]. For example, the Total Data Quality Management (TDQM) provides tools to analyze the data quality in information systems and suggests data cleansing techniques for DQ improvement [17]. However, prior work in this domain suffers from the major drawback, that either an active participation of users or domain experts in data quality improvement is necessary or the presented approaches refer to a (set of) reference data source(s) providing better or optimal data quality. It is obvious that in case of sensor data, the manual subsequent data quality correction for each measurement item is not feasible, and a high-quality reference for comparison is not present.

Instead, the data and data quality processing has to be configured to reduce the error amplification. Based on the classification of the quality-driven optimization problem, we identified the heuristic evolution strategy as appropriate tool to approximate the optimal problem solution in an acceptable timeframe. For the evaluation of different implementations of the evolution strategy we rely on the comprehensive study and experimental analysis provided in prior work and focus on the SPEA as a well-studied algorithm with high rankings in multiple test cases [18][19].

The application to multi-objective optimization problems is described in [20]. Empirical studies showed the practicability and superiority of the multi-objective evolution strategy (MOES) [21][22]. [23] discusses the benefits of parallel execution of evolutionary algorithms, which would further improve the performance of the quality-driven evolution strategy.

VII. Conclusion

In this paper, we presented the quality-driven optimization of sensor stream processing. On the one hand, the data quality of processing results, expressed by the DQ dimensions accuracy, confidence, completeness, amount of data and timeliness, were improved. On the other hand, the quality of service was increased by minimizing the data stream volume to comply with resource constraints in data streaming environments.

To identify candidates for the data quality improvement, we analyzed the operator repository of the quality propagation model presented in [1] and extracted sampling rate and group size as configuration parameters. Furthermore, the size of jumping data quality windows was detected as promising parameter for the data quality optimization. Based on these insights, we defined the multi-objective, non-linear, continuous optimization problem with side conditions.

To solve the problem of quality-driven optimization, we presented the generic optimization framework that can be instantiated with any optimization algorithm. Optimization time and quality were improved by satisfiability checks and two optimization modes for changing stream characteristics. Evolutionary algorithms represent the most promising optimization strategies for the defined complex optimization problem. Thus, we developed the quality-driven evolution strategy QES as sample instantiation of the generic framework.

Finally, we evaluated the proposed optimization strategy with the help of artificial data streams as well as real-world data from the contact lens production control. We showed, that QES solves the optimization problem in a reasonable timeframe and provides good scalability for complex data stream processing queries. The maintenance prediction for contact lens production could be determined more precisely by improving the data quality of the calculated maintenance date.

VIII. Acknowledgment

This work has been partially supported by the European Commission under the contract FP7-ICT-224282 (GINSENG).

References

[1] A. Klein, “Incorporating quality aspects in sensor data streams,” in *Proceedings of the 1st ACM Ph.D. Workshop in CIKM (PIKM)*, 2007, pp. 77–84.

[2] A. Klein and W. Lehner, “How to optimize the quality of sensor data streams,” in *ICCGI ’09: Proceedings of the 2009 Fourth International Multi-Conference on Computing in the Global Information Technology*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 13–19.

[3] R. Y. Wang and D. M. Strong, “Beyond accuracy: What data quality means to data consumers,” *Journal of Management Information Systems*, vol. 12, no. 4, pp. 5–33, 1996.

[4] M.-L. Lee, T. W. Ling, H. Lu, and Y. T. Ko, “Cleansing data for mining and warehousing,” in *DEXA*, 1999, pp. 751–760.

[5] P. J. Haas, “Large-sample and deterministic confidence intervals for online aggregation,” in *SSDM*, 1997, pp. 51–63.

[6] J. Kraemer and B. Seeger, “Pipes - a public infrastructure for processing and exploring streams,” in *SIGMOD*, G. Weikum, A. C. Koenig, and S. Deloch, Eds., 2004, pp. 925–926.

[7] A. Zell, “Javaeva: A java based framework for evolutionary algorithms,” 2009, http://www.ra.cs.uni-tuebingen.de/software/EvA2, last access: 14.06.2010.

[8] R. L. Edgeman and S. B. Athey, “Digidot plots for process surveillance,” *Quality Progress*, vol. 23, no. 5, pp. 66–68, 1990.

[9] N. R. Patel, R. L. Smith, and Z. B. Zabinsky, “Pure adaptive search in monte carlo optimization,” *Mathematical Programing*, vol. 43, no. 3, pp. 317–328, 1989.

[10] C. Batini and M. Scannapieco, *Data Quality: Concepts, Methodologies and Techniques*. Springer-Verlag, 2006.

[11] J. M. Juran, *Juran’s quality control handbook*, F. M. Gryna, Ed. McGraw-Hill, 1988.

[12] F. Naumann and C. Rolker, “Assessment methods for information quality criteria,” in *ICIQ*, 2000, pp. 148–162.

[13] V. C. Storey and R. Y. Wang, “An analysis of quality requirements in database design,” in *ICIQ*, 1998, pp. 64–87.

[14] D. M. Strong, Y. W. Lee, and R. Y. Wang, “Data quality in context,” *Communications of the ACM*, vol. 40, no. 5, pp. 103–110, 1997.

[15] M. A. Hernández and S. J. Stolfo, “Real-world data is dirty: Data cleansing and the merge/purge problem,” *Data Mining Knowledge Discovery*, vol. 2, no. 1, pp. 9–37, 1998.

[16] R. C. Morey, “Estimating and improving the quality of information in a mis,” *Communications of the ACM*, vol. 25, no. 5, pp. 337–342, 1982.

[17] J. M. Pearson, C. S. McCahon, and R. T. Hightower, “Total quality management: are information systems managers ready?” *Information Management*, vol. 29, no. 5, pp. 251–263, 1995.

[18] Z. Michalewicz, *Genetic Algorithms Plus Data Structures Equals Evolution Programs*. Springer-Verlag, 1994.

[19] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1989.

[20] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications (Advanced Information and Knowledge Processing)*. Springer-Verlag, 2005.

[21] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: Empirical results,” *Evolutionary Computing*, vol. 8, no. 2, pp. 173–195, 2000.

[22] T. Hanne, “Global multiobjective optimization using evolutionary algorithms,” *Journal of Heuristics*, vol. 6, no. 3, pp. 347–360, 2000.

[23] D. A. V. Veldhuizen, J. B. Zydallis, and G. B. Lamont, “Issues in parallelizing multiobjective evolutionary algorithms for real world applications,” in *ACM Symposium on Applied Computing*, 2002, pp. 595–602.

Functionality of a Database Kernel for Image Retrieval

Cosmin Stoica Spahiu, Cristian Marian Mihaescu, Liana Stanescu, Dan Burdescu, Marius Brezovan
University of Craiova
Faculty of Automation, Computers and Electronics
Craiova, Romania
{stoica.cosmin, mihaescu, stanescu, burdescu, marius.brezovan}@software.ucv.ro

Abstract—**This article presents a software tool that implements a dedicated multimedia database management server for managing alphanumerical and multimedia data collections from medical domain, along with the file structure used. The server is designed to manage medium sized personal digital collections. The recognized commands and the functionality of the server are also presented. An element of originality is that along with classical operations for databases, it includes a series of methods used for extracting visual information represented by texture and color characteristics. The extracted data are stored in the database in a special data type called IMAGE, with a specific structure that can be used for visual queries. Some clustering algorithms are used to increase the image retrieval speed.**

Keywords - multimedia; database management server; content-based retrieval; clustering; file system.

I. Introduction

Nowadays information is not limited only to strings. In order to facilitate different multimedia processing, advanced database management systems can integrate various data types (such as: images, video, text and non-numeric information) in a single database.

The aim of a database is to offer the user the possibility to use it for the query process. There are two types of queries that can be executed. One type of query process is the classical one that use simple text-based query. It can be used in the following cases:

a) The doctor knows the name of the patient and he wants to find all information about him

b) The doctor knows a certain diagnosis and wants to find all similar cases with the same diagnosis.

Another modern type of query is the content-based retrieval. That means that the search is made using similar characteristics of an image. The images have to be processed and extracted the characteristics. These characteristics will be compared later in order to find the images most similar to the query image. As an immediate effect the traditional information management systems cannot be used on this large collection of various data types [1][2].

The information in raw format is usually useless. The real benefits come when different decisions can be taken based on it, or it can be explored and obtained provenience information.

There are some domain areas were large volumes of information are stored in centralized or distributed databases. Few of these domains, are: digital libraries, image archives, bioinformatics, imagistic medicine, health, finances and investments, production, marketing strategies, telecommunication networks, scientific areas, WWW and biometry.

Usually in these databases the multimedia information (images, video, audio, etc.) is stored either in separate files, or inside the database in BLOB data types. This solution is not suitable all the time because makes it difficult to execute directly the visual content-based retrieval queries.

All these problems presented above are leading to the necessity for developing a multimedia database server that include intelligent data storage and processing methods used for content-based retrieval and automate data classification.

The solution proposed in this paper also uses a clustering method for grouping similar images.

Clustering is the process of grouping a set of physical or abstract objects into classes of similar objects. As a product of clustering process, associations between different actions on the platform can easily be inferred from the logged data. In general, the activities that are present in the same profile tend to be found together in the same session [3].

This paper presents a dedicated database management server (DBMS) that is based on the relational model. The DBMS can be used for managing medium sized databases from medical domain. For clustering process it is used Weka package.

The paper is structured as follows: in Section 2 there are presented similar implementations of well known servers, along their particularities. Section 3 presents the architecture of the server. Section 4 presents the database management and querying operations, and in the Section 5 it is presented a solution for clustering the images.

II. Related Work

Most of the applications that use multimedia information are based on traditional database management systems as MS SQL Server, My SQL or Interbase. Each of them offers partial support for multimedia content.

MYSQL offers only the BLOB data type that can be used to store images. A BLOB is a binary large object that can hold a variable amount of data. BLOB attributes have no character set, and sorting and comparing operations are based on the numeric values of the bytes [1] [4].

MS SQL Server offers a data type called “image”, but with no other support. It is considered a variable-length binary data having the size between 0 and 2^{31}-1

(2,147,483,647) bytes. There are no pre-defined functions that can be used for extracting the characteristics or building content-based queries. More than that, in the MS SQL Server 2008 it is specified that “ntext”, “text” and “image” data types will be removed in a future version of Microsoft SQL Server. The recommendation is to avoid using these data types in new development work, and to plan to modify applications that currently use them. It should be used instead: nvarchar(max), varchar(max), and varbinary(max) [5] [6].

The complete solution is provided by Oracle - the Oracle 10g database server and Intermedia tool that can manage all kind of multimedia data, including DICOM files [7][8] [9].

In addition to the image support offered via the ORDImage object type, in Oracle 10g version interMedia provides support for the first edition of the ISO/IEC 13249-5:2001 SQL/MM Part 5: Still Image Standard. This standard defines object relational types for images and image characteristics. Each object type includes attributes, methods, and SQL functions and procedures. The use of the SQL standard interface may make some applications to be more portable across various vendor databases.

For the clustering algorithms, there are many methods detailed in the literature: partitioning methods, hierarchical methods, density-based methods such as [10], grid-based methods or model-based methods. Hierarchical clustering algorithms like the Single-Link method [11] or OPTICS [12] compute a representation of the possible hierarchical clustering structure of the database in the form of a dendrogram or a reachability plot from which clusters at various resolutions can be extracted.

Because we are dealing with numeric attributes, only the iterative-based clustering is taken into consideration from the existing partitioning methods. The classical k-means algorithm is a very simple method for creating clusters. Firstly, it is specified how many clusters are being thought: this is the parameter k. Then k points are chosen at random as cluster centers. Instances are assigned to their closest cluster center according to the ordinary Euclidean function. Next the centroid, or the mean, of all instances in each cluster is calculated – this is the “means” part. These centroids are taken to be the new center values for their respective clusters. Finally, the whole process is repeated with the new cluster centers. Iteration continues until the same points are assigned to each cluster in consecutive rounds, at each point the cluster centers have stabilized and will remain the same thereafter [10][11][12].

From a different perspective there may be computed the following parameters for a cluster: means, standard deviation and probability (μ, σ and p). The expectation-maximization (EM) algorithm that is employed is a k-means clustering algorithm type. It takes into consideration that we don’t know either parameters. It starts with initial guess for the parameters, it uses them to calculate the cluster probabilities for each instance, it uses these probabilities to estimate the parameters, and repeat. This is called the EM algorithm for “expectation - maximization”. The “expectation” is the first step needed to calculate the cluster probabilities (which are the “expected” class values); the “maximization” of the likelihood of the distributions given is the second step which it is needed to calculate the distribution parameters [12].

The quality of clustering process is measured by computing the likelihood of a test data set given from the obtained model. The goodness-of-fit is measured by computing the logarithm of likelihood, or log-likelihood: the larger this quantity is, the better the model fits the data. Instead of using a single test set it is also possible to compute a cross validation estimate of the log-likelihood.

III. Database Kernel – Overview

The kernel it is a tool that can be used for database creation, maintenance, and executing simple text-based queries or content-based visual queries. The multimedia digital collections from medical domain are used for these queries.

This dedicated MMDBMS permits database creation, table and constraints definition (primary key, foreign keys), inserting images and alphanumerical information, simple text-based queries and content-based queries using color and texture characteristics. The software tool is easy to be used because it respects the SQL standard and has the advantage of low cost. The users do not need advanced informatics knowledge. It is a good alternative for a classical database management system (MS Access, MS SQL Server, Oracle10g Server and Intermedia), which would need higher costs for database server and for designing applications for content-based retrieval [13].

The Figure 1 presents the general architecture of the medical system.

In the first step the client application that uses the server should connect to the database. This way it will be created a communication channel between them. All commands and responses will use this channel to send queries requests and receive answers.

The server has two main modules: kernel engine and database files manager.

The kernel engine includes all functions implemented in the DBMS. It is composed from several sub-modules each of them with specific tasks:

- The main module. It is the module which manages all communications with the client. It is the one that receives all queries requests, check what the type of requested query is, extracts the parameters of the query and calls the specific module to execute it.

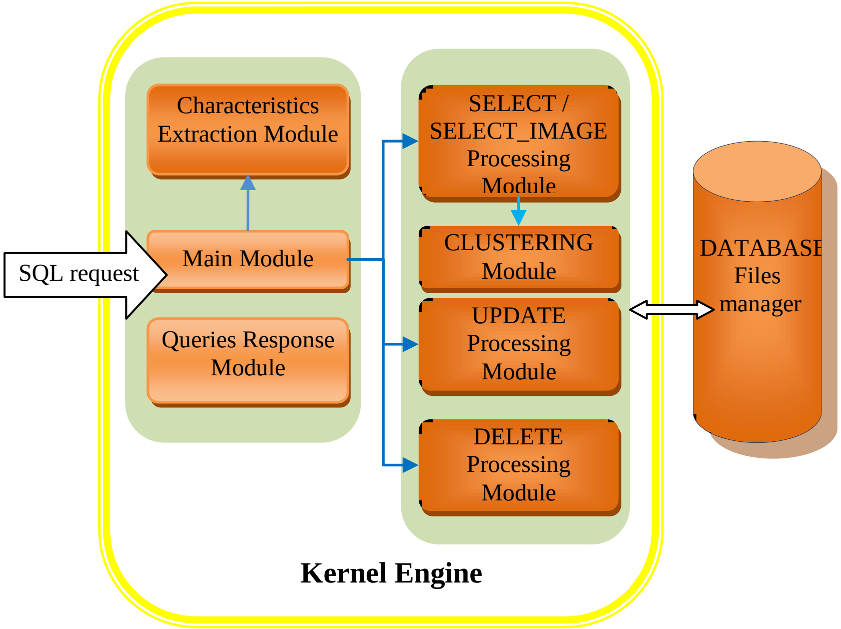


Figure 1. General architecture of the system.

- Queries response module. After the query is executed, the results will be sent to the Queries Response Module. It will compact the result using a standard format and then return it to the client. The client will receive it on the same communication channel used to send the request.
- Select/Select/Image Processing module. If the main module concludes that is a SELECT SQL command, it will call the Select Processing module. This module extracts the parameters from the query and then search in the database files for specific information. If the query is a SELECT IMAGE query, it will use for comparation the similitude of characteristics instead equality of parameters.
- Characteristics Extraction Module. When the main module receives a SELECT IMAGE or a UPDATE query which uses an image that is not already in the database it is needed first to process it. This module is called to extract the color and texture characteristics of the image. The data of the results will be used to initialize a variable of IMAGE data type.
- Clustering Module. The role of this module is to assign each image to a cluster containing similar images.
- Update Processing Module. When the query received from the user is an UPDATE command, it will be called this module to execute it.
- Delete Processing Module. It is called when the user executes a DELETE command. The kernel executes only logic deletes. It never executes physical deletes. The physical deletes are executed only when a "Compact Database" command is sent by the user.

The second main module is the Database Files Manager. It is the only module that has access to the files in the database for reads and writes. It is its job to search for information in the files, to read and write to files and to manage databases locks. There are two kinds of locks: shared locks (read locks) and exclusive locks (write locks). A read lock it is enabled when a client module requests a read from a file (that represents a table in the database). All other read requests will be permitted but no writes will be allowed. If the client module request a write access, it will be enabled a write lock. No other requests will be allowed until the lock is canceled.

The results will always be returned to the module which made the request. The data read or wrote to files is not structured in any way. This module does not modify the data structure in any way. All the results will be raw data, as it is read from the files or received from client modules.

IV. DATABASE MANAGEMENT AND QUERY

This system is a client-server tool that can be used by all kind of applications that work with databases. The client applications can be implemented in C++, Java or any other language that supports TCP/IP communication [1][2].

An element of originality for this tool is that among the classic operations that can be executed (like databases creation, maintenance, simple text-based query) it is included a new module for content-based visual query using color and texture characteristics. This module will search in the database for images with similar texture and color pattern and return all the associated information (the similar image along with the rest of the attributes).

The first step in the communication between client-server is to establish the TCP connection. Only after the client can send query requests. The connection is made using TCP standard and should include a username and a password for security. Each user has defined a series or rights that he can benefits from: create databases, create tables in a specific database, modify tables' structures, insert data into tables and execute queries. Each of these rights is defined at the database level. When a database is created, only users who have defined rights can work with it.

The implemented commands are related to: listing the available databases, create/delete a database, create/delete a table, adding constraints for tables (primary key, foreign key), etc. Another element of originality for the server is that the images can be seen automatically, when browsing the records in the database.

The commands categories recognized by the system are:

A. Managing Databases

1. *create database <database_name>*

eg: create database student

This command creates a new database on the server.

2. *use database <database_name>*

e.g.: use database student

Before being able to execute operations on it, first it should be defined the default database.

3. *create table <table_name> (<atribute1> <data_type1>, <atribute2> <data_type2>, [<atribute3> <data_type3>,…])*

e.g.: create table person (id int, name varchar(20), age double)
create table image (id int, name varchar(20), picture image)

The command will create a new table in the database with the specified attributes. A table there can have any number of attributes. The data types that can be used are:

- integer
- double
- varchar
- image

If it is used the varchar data type, it is needed to be specified the maximum size of the array.

4. *alter table <table_name> add primary key (<attribute_name>)*

e.g.: alter table person add primary key (id)

The command will define <attribute_name> as primary key in <table_name>.

The composed primary keys can be defined by using this command for each attribute in the key.

5. *alter table <modified_table_name> add foreign key (<atribute1_name>) references <table_name2> (<refered_ atribute2_name>)*

e.g.: alter table person add foreign key (id) references picture(id)

The command defines a foreign key between two tables <modified_table_name> and <table_name2>. The key will link the two attributes <atribute1_name > and <refered_ atribute2_name>.

The many-to-many links cannot be represented, as the server respect the relational model. That is why at least one of the attributes which is part of the external key, must be a primary key.

6. *get table keys <table_name>*

e.g.: get table keys person

This command returns all the references defined on all of the attributes existing in the table

B. Inserting Data into the Database

The communication between client application and DBMS is based on messages exchange. All messages represent commands written in SQL language.

1. *insert into table_name values (value1, value2, [value3, …])*

There are two possible cases:

a) *Inserting only text:*

e.g.: insert into PATIENT values("disease", "polyps")

In this case the main module receives the command, and checks the user's rights. If the user has enough rights to insert new values, it calls the Update Processing Module to add data into the database.

b) *Inserting text and images*

e.g.: insert into person values (1,'George O.',20.5)
insert into image values(1, 'George O.','analysis.bmp')

The values between parentheses will be added in the table <table_name>. It is mandatory to add values for each attribute. There should be no NULL attributes. The order of the inserted values should be the same as the one of the attributes in the table.

For this example it is called first the Update Processing Module. It checks the metadata of the table and finds out that one of the attributes has the type "image". It notifies the Main Module that it should receive an image file from the client called "analysis.bmp". After the image is received, it will call the Characteristics Extraction Module to process the image and create the image type value. The record is inserted into database only after this type is created.

In order to increase the retrieval speed for future queries, it is also called the Clustering Module. It will assign the new image to the cluster that contains images with similar characteristics.

C. Simple Text-Based Queries

The text-based queries can be used on any attribute defined in the table. It is used the SQL syntax, having the possibility to specify to return only certain attributes from a record, or all the attributes. There are accepted any number of conditions specified in the WHERE clause, which are combined by AND or OR operators.

The syntax is:

*select * from table_name where attribute1 [</=/>] values [and attribute2 [</=/>] values2 …]*

e.g.: select * from person
select * from person where age>45.5 and id<40 and name= 'Adrian Ionescu'

The command returns all the records along with all the attributes specified in the WHERE clause.

In this case the main module receives the SELECT command and checks the user's rights. If the user has enough rights to execute select commands, calls the SELECT Module to search data into the database.

In the implemented version of the server there is accepted only to retrieve information from a single table. There are also not accepted sub-queries.

D. Visual Content-Based Queries

The MMDBMS makes a series of validations, both in the table design phase and after that, when adding values to the database: unique names for databases, tables and, attributes, checking the constraints, etc. The users have the possibility to build content-based visual queries in a simple manner, to the image level. The elements which can be used to build such a query are: like, select, from, where, threshold, maximum number of returned images. The retrieval operations use two functions for computing the similitude between two images: one for texture (Euclidian distance) and the other for color characteristics (histograms intersection). The user has the possibility to choose if he wishes to use both or only one of them. Based on the user's settings, the system will create a modified SQL command, adapted for this type of query.

A special select command is included which can be used for content-based retrieval operations. When the user sends a SELECT_IMAGE command to the system, the DBMS will try to find all the records in the table that contain similar images. This is the most complex operation that has been implemented. The parameters received are: the attributes that should be returned, the table where to search, methods that should be used for computing the similitude, maximum number of returned images, and the image itself. After receiving this command, the server waits to receive the query image from the client. After the image is transferred, it will be saved temporary on the disc using a name that is unique. This is needed in case several clients send the same image simultaneously to the server. After receiving the image, it is processed first and extracted the characteristics. The system can go further only after this step is finish. To the next step it will search the database for similar images using for similarity the specified method [16].

Before being executed, the command is added to the transaction file.

The syntax of the query is:

SelectImage [/<attribute_name1>, <attribute_name2>, …] FROM <table_name> WHERE <image_column_name> LIKE QueryImage (METHOD: color[,texture] MaxImages <no_of_returned_images>)*

Where:

"attribute_name1" is the attributes that should be returned

"table_name" is the name of the table

"image_column_name" is thename of the column that contains the image itself

"no_of_returned_images" specifies the maximum number of images returned

E.g.:

selectImage name, picture from picture where picture Like QueryImage (method: color maxImages 5)

selectImage * from Patients where age>50 and picture like queryimage

In the first example it is specified that the results are from table "picture". There will be retrieved only the name and picture attributes for the images similar with the query image that is specified by the user. The result will contain only the most similar 5 images.

This type of query can be visually built in two different ways. In the first case the user sends a SELECT ALL query that will retrieve all the records in the table. From these records it selects the image he wishes to be used as a query and then sends a SELECT_IMAGE query with it. The DBMS returns all records containing similar images. The second possibility is when the user has his own image, he calls the Processing Module first and creates the image type value. Then it is sent to the system a SELECT_IMAGE command.

When inserting an image in an Image data type, the Image Processing Module will be automatically called after the image is selected. This will automatically extract the color information and texture (represented by a 12 values vector).

The execution time depends mainly to the number and size of images that will be sent to the client. This is due to the fact that sending the images over the TCP is much slower that processing the images, especially when involves a high number of images.

E. Administration Suport

The MMDBS offer also support for users' management: creating new users, deleting users and managing their access rights.

The user who creates a database will have absolute rights over it. All the other users excepting the administrator will need to receive rights access in order to execute operations over it. There are two kind of rights: general rights (given only by the administrator) and particular rights (for a specified database), given by the administrator or by the database owner.

The general rights refer to:

- Permissions for creating databases
- Permissions for creating other users.

The particular rights that can be given (for each database in part) are:

- Execute select operations
- Execute update/delete operations
- Create Tables
- Modify tables structures
- Delete table/database

The commands are:

1. *create user <user_name> password <password> cd=[0/1] cu=[0/1]*

e.g.: create user cosmin password test cd=1 cu=0

This command will create a new user with the name "cosmin" and the password "test". He will have defined from the beginning the rights to create new databases (cd = 1 for granting this right, or cd = 0 for denying the right) and to create other users (cu = 1 or cu = 0).

For the moment the user doesn't have any right over the existing databases. He cannot execute any commands, as he has no rights granted.

2. *update user rights <nume_user> on <nume_baza_de_date> set ct=1[/0] s=1[/0] u=1[/0] m=1[/0]*

e.g.: update user rights cosmin on person set ct=1 s=1 u=0 m=0

This command is used for granting users rights at the databases level. The rights that can be specified are:

- create table (ct = 1 – granting the right ; ct=0 – denying this right),
- executing select commands(s = 1 – granting the right; s = 0 – denying this right)
- updating information using update/insert (u = 1 – granting the right ; u=0 – denying this right)
- modifying table structure (m = 1 – granting the right ; m=0 – denying this right)

3. *get user rights <user_name> on <database_name>*

e.g.: get user rights cosmin on person

The function is useful to verify the user's rights for a specific database. If only the general rights are it should be specified the "default" database.

4. get table <table_name> metadata

e.g.: get table person metadata

The command returns the structure of a table (defined attributes and their data types). It is useful for a client in order to interpret and display correctly the information received from server. Before executing this type of command it should be executed first USE DATABASE command.

5. *set user password <user_name>,<password>*

e.g.: set user password cosmin, test01

The command permits changing users' passwords. Each user can change his password, but he cannot change other's users' password. The system administration is the only one who can do this.

6. get databases list

It returns to the user the list with all the databases defined in the system. It is very useful for the client's application in order to know where to search for table lists.

7. *get tables list*

It returns to the user the list with all the tables defined in the databases. It is very useful for the client's application in order to know in which table can search for metadata.

In order to increase the retrieval speed it is used the Clustering module. In this case, first it is searched for the cluster that contains images most similar with the query image. After this cluster is located, the system looks for similar images only inside it. In this way the quantity of data needed to be compared is reduced substantially.

V. Clustering the Images

In order to increase the speed of the retrieval process, after an image is inserted in the database, it should be assigned to a cluster containing similar images.

For this operation we chose to use the algorithms provided by Weka package [12]. Weka is a collection of machine learning algorithms for data mining tasks. It contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

The Expectation-Maximization (EM) algorithm from Weka clustering package EM is a statistical model that makes use of the finite Gaussian mixtures model. The algorithm is similar to the K-means procedure in that a set of parameters are re-computed until a desired convergence value is achieved [14][15].

It needs the input data to be in a custom format called arff. Under these circumstances we have developed an offline Java application that queries the database and creates the input data file called activity.arff. This process is automated and is driven by a property file in which there is specified what data will lay in activity.arff file.

The most important step in this procedure is the attribute selection and the granularity of their nominal values. The number of attributes and their meaning has a crucial importance for the whole process, since irrelevant attributes may degrade classification performance in sense of relevance. On the other hand, the more attributes we have the more time the algorithm will take to produce a result. Domain knowledge and of course common sense, are crucial assets for obtaining relevant results.

For an image we may have a very large number of attributes. Still, in our procedure we used only two sets: color characteristics and texture characteristics. Here is how the arff file looks like:

```
@relation images
@texture
        {v1,v2,v3,v4,v5, .. v12}
@ color
        {v1,v2,v3,v4,v5,…,v166}
@data
Texture1 v1,v2,…
Color1 v1,v2, ..
Texture2 v1,v2,…
Color2 v1,v2, ..
….
```

As it can be seen from the definition of the attributes each of them has a set of five nominal values from which only one may be assigned. The values of the attributes are computed for each of the XXXX images and are set in the @data section of the file. For example, the first line says that the image 1 has value v1 for parameter named color1 and has value v1 for parameter named texture2.

The granularity for the nominal values of the attributes can be also increased. In our study we considered only five possible values but we can consider testing the algorithm with more possible values. This should have great impact on the number of the clusters obtained. The time taken by the algorithm to produce results should also increase.

Running the EM algorithm created XX (e.g. three) clusters. The procedure clustered XX (e.g. 91) instances (34%) in cluster 0, 42 instances (16%) in cluster 1 and 135 instances (50%) in cluster 3. The final step is to check how well the model fits the data by computing the likelihood of a set of test data given the model. Weka measures goodness-of-fit by the logarithm of the likelihood, or log-likelihood: and the larger this quantity, the better the model fits the data. Instead of using a single test set, it is also possible to compute a cross validation estimate of the log-likelihood. For our instances the value of the log-likelihood is -2.61092 which represent a promising result in the sense that instances (in our case images) may be classified in three disjoint clusters based on their characteristics.

This function of the database is activated when the user sends an INSERT/UPDATE command and after processing the image. The system calls the Clustering Module in order to include the newly inserted image in a cluster. These clusters will be used for retrieval when user sends a SELECT command. The main problem in this case for the Clustering Module is to determine which is the cluster containing similar images. After this cluster is located, only images contained here are computed in order to find images with the highest similarity to the query image.

VI. Client-Server Communication

The implemented server is a client-server application based on TCP/IP sockets. Each client must have a copy of the client application in order to have access to databases on the server. The client application can be implemented both as a standalone application that can be installed on the client's computer and as an application opened from the internet browser. The second option is recommended because it is not necessary any installation and can be used on any computer with internet access.

The first step is to start the server in order to wait for clients' connections. A condition for the Microsoft Vista operating systems (and future versions based on this architecture) is to execute the server with administrative rights. If not, the server will not be able to receive connections from clients.

The second step is to create a connection between client and server. For this, the client application needs to know the IP address and port where to find the server. The port is specified in the configuration file on the server.

When a client connects to the server, all the communication will be managed on a separated execution thread, independent to all the other connections.

In the third step, the client has to authenticate to the server based on a user and password. The password is defined when the user was added into the system and it is stored in the DEFAULT database. The password is encrypted for safety, on a key based on the user's name. If the user fails to enter the right password for three times, the IP address used to log-in will be blocked for a period of 30 minutes. During this period it will not be accepted any connection from that address.

After the client is authenticated, the server can accept commands. These commands can be only based on text or they can imply sending image files.

The commands that imply only text are:

a) INSERT

If the table where the insert is made does not contain any Image data attribute, the command will need to transfer only text information. The client's command will be received by the central module of the server. It will analyze and recognize that it is only a text-based insert operation and it is not needed to receive anything else from the client (e.g., an image). It will be sent further for execution to the Update module. This module responds with an "OK" message if the insert was successful or with an error message if one of the table's constraints did not permit the insert (e.g., primary key duplication). The message is sent to the "Response" module that returns it to the client.

b) SELECT

If the table used does not contain any image data type attribute, the communication will involve only text. The server receives the command, checks and sees that it is not needed to receive any other information and sends it to the "select" module. This module makes the search and put the result in a special structure that will be returned to the "Response" module. The client receives first the metadata. Only after this it will know how to correctly interpret the following information. On the next step, the server sends the number of records that will be sent to the client. At the end it sends record by record all the information returned by the select.

c) Obtaining databases list, available on the server.

This type of requests will be treated in a similar manner. The server receives the command, checks its type and calls the corresponding module. The response will be a string array.

The commands that might need images transfer are:

a) INSERT

If the table used for insertion has an attribute of image data type, the server will need to exchange text data and image data. The command sent by the client is received first by the main module of the server. For communication with the client it is used a handshaking method: client sends the insert command to the server, the server responds with an OK. It checks and recognizes the insert operation that needs an image. Next, the communication module sends a message to the client to announce that it waits to receive an image. The client responds with the image itself. The server confirms then with an OK receiving the image. If one of them does not receives the confirmation, it will repeat sending the information.

The received image is saved in a temporary folder. The identification of the execution thread will be added to the name of the file in order to ensure that each name is unique.

The "Characteristics extraction" module will process the image and extract the characteristics.

The "Update" module will be called only after the last one finishes processing and all the information needed to make the insertion is available: the text information will be inserted in the table file and the image in the images file.

At the end, the Update Module will return an "OK" message if insertion was successful or an error message if it will be violated one of the table's constraints.

The following figure presents the communication with the client:

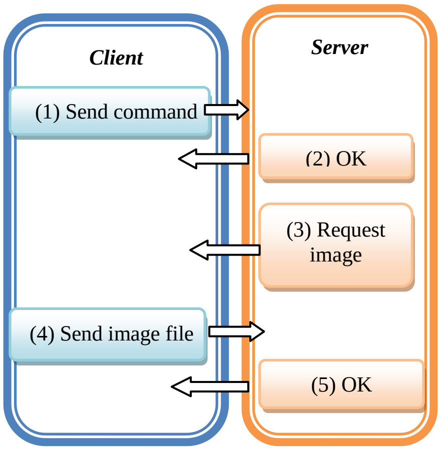


Figure 2. Client-server communication.

The execution time of this command depends on two components: the time needed to send the image to the server (link speed) and the time needed to process the image (processor speed). These components depend mainly on the image size.

VII. Server Files Structure

Because the images are stored directly in the database it could be a problem the files' size on the disc. The size of the images file can grow very much, reaching easily several hundred MB, or even few GB. This is the main reason why it is recommended to use the NTFS file system [2].

The Database File Manager Module is the module that effectively read and writes from/to files. It uses the pagination reading and writing, all information being written in pages of information. A single file can contain several types of information that are written in different type of pages. In order to identify the type of each page, all the pages are indexed in an index file.

The server uses the following types of files:

- Users.ssc – it is the file where there are stored all users created on the server. For each user it is stored two set of rights: general rights and rights detailed for each database. Each of these types of information is stored in a separate page.
- Index_users.ssc – it is the file where the pages used to store users' information are indexed. There can be two types of pages: pages used to store users information and pages used to store users' rights.
- Transaction_log.ssc – it is a file created for each database. It stores the transaction log history. It can be used for database recovery in case of a system error.
- <table_name>.ssc – it is the table file. This file contains two types of data: metadata information and data information.
- Page_Index_<table_name>.ssc – it is the file where all the pages from table file are indexed for a faster retrieval. The pages can be used to store either metadata, either useful information.
- Images.ssc – on the table page there is stored only the characteristics of the image. The image itself it is stored on a special file called images.ssc.
- Index_<table_name>_<key>.ssc – this type of file is optionally and is created by the user if he wishes to index the information in table by one of its columns (key).

The server creates a new folder for each database. The name of the folder is considered to be the name of the database. The user who created the database will have absolute rights over it. All the other users excepting the administrator will need to receive rights access in order to execute operations over it. There are two kind of rights: general rights (given only by the administrator) and particular rights (for a specified database), given by the administrator or by the database owner.

B. Reading and writing information

When the server receives a command for an update, insert or select, it has to store the operation in the transaction log file before doing any other operation. The record will include information about the user who sent the command and about the operation that should be executed. Only after this process is finished, the system goes further to the next step and tries to read or write from files. In order to know how to interpret the information that it will read, it has to find the structure of the table first. It will search in the page_index_table_name file the page containing table metadata and read it. Now, the system can read and "understand" correctly the information from table and can separate correctly each attribute's data. There are two possible cases for reading data. In the first case, the user has already created an index for the specified key using a Create Index command. In this case a file called Index_table_name_key should exist. It will be searched for the page containing the key and read into memory from the table file. Only after that it can be read record by record to find the right information. In the second case, the information is not indexed. In this case each page existing in page_index_table_name file is read in the memory and then read record by record from the memory to find the right information.

The algorithm used for reading is presented next:

```
Procedure Read_info (key)
 Find in Page_Index_table_name metadata page
 Read corresponding page from memory
 If exists index (key)
   Find key in table        Index_table_name_key
       If information found then
                 Read corresponding page to memory
                 Find key in the page
       Else
                 Information not found
 Else
       For each page of information from
       Page_Index_table_name
                 Read corresponding page from memory
                 Find key in the page
```

When the system tries to write information in the file it is searched in the page_index_table_name file the last page containing information. There are two cases: there is enough free space on the page to write the current data (in this case it is written after the last record in the page and then updated the information from page_index_table_name file) or there is not enough space to write in the current page. In this case it is created a new page where the data is written and then writes the details of this page to the page_index_table_name file. If the written information contains a key that is indexed it is called the function update index. The algorithm is:

```
Procedure Write_info (info)
 Find in the page_index_table_name last page with info
 Read corresponding page to memory
 If firstFreePositionInPage + size (info) < pageSize
        Write info to page
        Update page information in Page_Index_table_name
 Else
        Create a new page
        Write info to page
        Add page information in Page_Index_ table_name
 If exists index (info)
        Update index file.
```

If the user sent an update command the system will locate the existing information, update data and then write page back to the disk. For locating the data that should be updated it is used the Read_Info procedure described above. After the information is updated it is possible that the size of the record to be changed. In this case the information from page will be reorganized. It is deleted the record from its current location and added to the end. If the page free space is not big enough to include the current information the page is written back to disk without the modified information and the information will be added to a new page.

The algorithm is described next:

```
Procedure Update_info (key
 Read_info (key)
 Update info
 Delete_info(page, info)
 If firstFreePositionInPage + size (info) < pageSize
        Write info to page
        Update page information in Page_Index_table_name
 Else
        Create a new page
        Write info to page
        Add page information in Page_Index_ table_name
 If exists index (info)
        Update index file.
```

VIII. CONCLUSIONS

The paper presents the organization of an implemented multimedia, relational, database multimedia server kernel that included a new data type, called IMAGE.

It is created for managing and querying medium sized personal digital collections that contain both alphanumerical information and digital images (for examples the ones used in private medical consulting rooms). The software tool allows creating and deleting databases, creating and deleting tables in databases, updating data in tables and querying. The user can use several types of data: integer, char, double and image. There are also implemented the two constraints used in the relational model: primary key and referential integrity.

The software tool can execute both simple text-based queries using one or several criteria connected with logical operators (and, or) and content-based visual queries at image level, taking into consideration the color and texture characteristics. These characteristics are automatically extracted when the images are inserted in the database.

It is presented the algorithms used for reading/writing the information into the database, along with the files used by the server.

Storing the images inside the database can easily lead to very large files that can go beyond several GB. That is why the server administrator has to take care of the available disc space. It is also recommended that the server to be installed on a NTFS file system that has almost no size limit for files, comparing to the FAT file system that accepts files up to 4 GB in size.

This software can be extended in the following directions:

- Adding new traditional and multimedia data types (for example video data type or DICOM type - the main area where this multimedia DBMS is used it is the medical domain. The DICOM type of data is used for storing alphanumerical information along with images existing in a standard DICOM file that is provided by a medical device)
- Studying and implementing indexing algorithms for data inserted in the tables.

The speed of the image retrieval module is increased by using a clustering module that will group the images in clusters. The similarity between a query image and the images in the database will be computed only taking into consideration the images from the cluster with the best similarity.

REFERENCES

[1] C. Stoica Spahiu, C. Mihaescu, L. Stanescu, D.D. Burdescu, and M. Brezovan, “Database Kernel for Image Retrieval”,. Proceedings of The First International Conference on Advances in Multimedia (MMEDIA 2009), Colmar – France, pp. 169-173, 2009

[2] C. Stoica Spahiu, Liana Stanescu, D.D. Burdescu, and M. Brezovan, “File Storage for a Multimedia Database Server for Image Retrieval”, Proceedings of The Fourth International Multi-Conference on Computing in the Global Information Technology (ICCGI 2009), Cannes/ La Bocca – France, France, pp.35-40, 2009

[3] SQL Server 2008 Books Online, January 2009, http://msdn.microsoft.com/en-us/library/ms187993.aspx

[4] MySQL 5.0 Reference Manual, 2009
http://dev.mysql.com/doc/refman/5.0/en/blob.html

[5] A. Chigrik, SQL Server 2000 vs Oracle 9i, 2007
www.mssqlcity.com/Articles/Compare/sql_server_vs_oracle.htm

[6] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” Proc. of the 20th VLDB Conference, pp. 487-499, Santiago, Chile, 1994.

[7] M. Kratochvil, The Move to Store Images In the Database, 2005

http://www.oracle.com/technology/products/intermedia/pdf/why_images_in_database.pdf

[8] Oracle® Database SQL Reference 10g Release 2 (10.2) http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/sql_elements001.htm

[9] New interMedia Features in Oracle10g http://www.oracle.com/technology/products/intermedia/htdocs/imedia _new_features_in_10g.html

[10] M. Ester, H.P.Kriegel, J. Sander, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, Proc. KDD’96, Portland, OR, pp.226-231,1996.

[11] R. Sibson, “SLINK: An Optimally Efficient Algorithm for the Single-link Cluster Method”, The Computer Journal, 16(1): 30-34, 1973.

[12] www.cs.waikato.ac.nz/ml/weka

[13] L. Stanescu, D.D. Burdescu, M. Brezovan, C. Stoica Spahiu, and A. Ion, “A New Software Tool For Managing and Querying the Personal Medical Digital Imagery”, Proceedings of the International Conference on Health Informatics, Porto – Portugal, pp. 199-204, 2009

[14] http://grb.mnsu.edu/grbts/doc/manual/Expectation_Maximization_EM.html

[15] James Foulds. Learning instance weights in multi-instance learning. Master's thesis, Department of Computer Science, University of Waikato, 2008 http://adt.waikato.ac.nz/uploads/approved/adt-uow20080308.150730/public/02whole.pdf

[16] L. Stanescu, M.C. Mihaescu, D.D. Burdescu, E. Georgescu, and L. Florea, “An Improved Platform for Medical E-Learning”, Lecture Notes in Computer Science 4823, Springer, pp.392-403, 2008.

Study of a Secure Backup Network Mechanism for Disaster Recovery and Practical Network Applications

Noriharu Miyaho, Yoichiro Ueno, Shuichi Suzuki,
Kenji Mori
Department of Information Environment,
Tokyo Denki University,
Inzai-shi, Chiba, 270-1382 Japan
e-mail: miyaho@sie.dendai.ac.jp,
ueno@sie.dendai.ac.jp, ssuzuki@sie.dendai.ac.jp,
mori@ine.sie.dendai.ac.jp

Kazuo Ichihara
Net&Logic Inc.,
Setagaya-ward, Tokyo, Japan
e-mail: Ichihara@nogic.net

Abstract—**A practical mechanism for a file-backup system concept is proposed. In this paper, it is demonstrated that a highly secure file backup mechanism can be achieved by combining technologies that implement the following: spatial random scrambling of file data, subsequent random fragmentation of the file, the duplication and encryption of each file fragment using a stream cipher code in each encryption stage, and the corresponding notification of the history data of the encryption key code sequence used in "encryption metadata". If a disaster should occur in the data backup center, prompt data recovery can be easily and securely achieved by making use of a massive number of widely distributed wired PCs, mobile PCs, cellular phones managed by multiple supervisory servers which are secretly diversified but functionally combined. In an experimental evaluation, encryption performance, spatial scrambling performance and the average response time from the Web server have been estimated in terms of the memory load characteristics of the data center. Discussion is also provided on an effective shuffling algorithm to determine the dispersed location sites. Finally this paper describes a system configuration for a practical application, which will be soon commercialized.**

Keywords-disaster recovery; backup; metadata; distributed processing; shuffle algorithm

I. INTRODUCTION

Innovative network technology to guarantee, as far as possible, the security of users' or institutes' massive files of important data from any risks such as an unexpected natural disaster, a cyber-terrorism attack, etc., is becoming more indispensable day by day. To meet this need, technology is required that can be realized with affordable maintenance operation costs and that provides high security.

For this application, Data Grid technology is expected to provide an effective and economical back up system by making use of a very large number of PCs whose resources are not fully utilized. In particular, Data GRID technology using a distributed file data back-up mechanism will be utilized by government and municipal offices, hospitals, insurance companies, etc., to guard against the occurrence of unexpected disasters such as earthquakes, big fires and storms.

However, these methods involve high operation costs, and there are a lot of technical issues to be solved, in particular relating to security and prompt restoration in the event of disasters occurring in multiple geographical locations.

The background leading to the need for the proposed economical backup system and its objectives are shown in Figure 1.

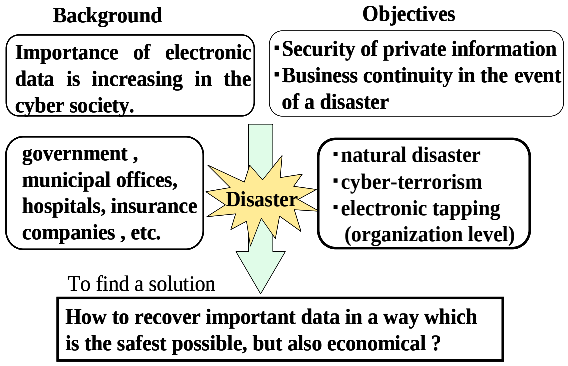

Background
Objectives
Importance of electronic data is increasing in the cyber society.
・Security of private information
・Business continuity in the event of a disaster
government , municipal offices, hospitals, insurance companies , etc.
Disaster
・natural disaster
・cyber-terrorism
・electronic tapping (organization level)
To find a solution
How to recover important data in a way which is the safest possible, but also economical ?

Figure 1. Background and objectives of the proposed system

On the other hand, there is a network infrastructure which can be used to distribute and back-up a great number of data files and a large number of remote office personal computers, cellular phones, and smart phones such as iPhone4 can be utilized for this purpose.

In this paper we propose an practical file back-up concept which makes use of an effective ultra-widely distributed data transfer mechanism and a high-speed encryption technology, based on the assumption that we can use a small portion of the memory of a large number of PCs and cellular phones that are in use in daily life, to efficiently realize safe data backup with an affordable maintenance operation cost [1].

Figure 2 shows a comparison of the proposed method with the conventional method in terms of network utilization. The principal differences of the proposed system are as follows. (1) It does not require the use of expensive leased lines. (2) It only utilizes otherwise unused network resources such as unused network bandwidth, unused memory capacity of PCs, cellular phones and smart phones, etc. (3) It can cipher a number of important data files at the same time using spatial scrambling and random dispatching technology. (4) As the number of user companies increases, the security against being deciphered illegally increases accordingly. (5) The maintenance cost can be drastically reduced. In addition, since it adopts a stream cipher, the speed of encryption of data increases, so it can also be applied to secure streaming video transmission services.

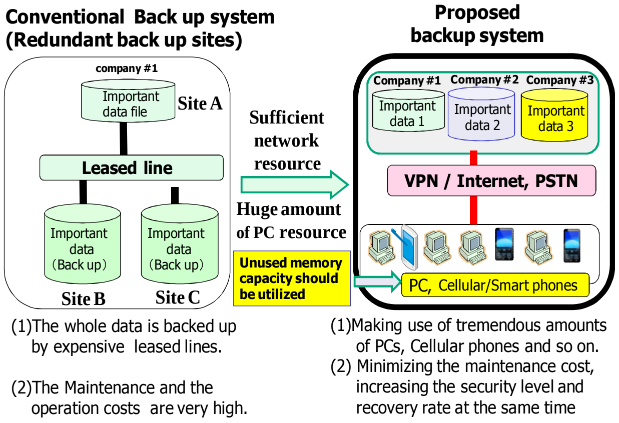

Conventional Back up system
(Redundant back up sites)
company #1
Important data file
Site A
Leased line
Important data (Back up)
Important data (Back up)
Site B
Site C
Sufficient network resource
Huge amount of PC resource
Unused memory capacity should be utilized
Proposed backup system
Company #1
Company #2
Company #3
Important data 1
Important data 2
Important data 3
VPN / Internet, PSTN
PC, Cellular/Smart phones
(1)The whole data is backed up by expensive leased lines.
(2)The Maintenance and the operation costs are very high.
(1)Making use of tremendous amounts of PCs, Cellular phones and so on.
(2) Minimizing the maintenance cost, increasing the security level and recovery rate at the same time

Figure 2. Comparison of the proposed system with a conventional data backup system

In general, encryption technology can use two types of technology, that is, block cipher technology or stream cipher technology.

In case of the block cipher technology, the data is divided into successive blocks and each block is processed separately for point-to-point systems; as a result, the encryption speed becomes quite slow. As the data volume increases, the required processor and memory cost increases in an exponential manner.

On the other hand, in case of the stream cipher, the input data is simply operated on bit by bit using a simple arithmetic operation. Therefore, high-speed processing becomes feasible. These are the fundamental differences between the two cipher technologies [2].

When an ultra-widely distributed file data transfer technology, a file fragmentation technology and an encryption technology are used together, then, quite different characteristics from a point of cipher strength arise. It is possible to combine the use of technologies such as the spatial random scrambling of all data files, the random fragmentation of the data files, the corresponding encryption and duplication of each file fragment using a stream cipher, and, in addition, the corresponding notification of the history data including the encryption key code sequence, which we call "encryption metadata". When these are all combined is it possible to realize an effective, economical and prompt data backup system [3][4].

In this case, PC terminals deployed in remote offices or cellular phones can be used by making effective use of combined stream cipher and distributed data transfer technologies. By making use of the proposed combined technology, illegal data recovery by third party interception becomes almost impossible and an extremely safe data backup system can be realized at reasonable cost.

The proposed technology can increase both of a cipher code strength and a data recovery rate. The economical operation cost achieved by using the proposed backup technology, while high-speed network technology also makes this achievement more practicable.

To realize the proposed disaster recovery mechanism, the following three principal network components are required: (1) a secure data center, (2) several secure supervisory servers, and (3) a number of client nodes such as PCs or cellular phones.

We clarify the relationships between data file capacity, number of file fragments and number of duplications in case of the disaster recovery hereafter [5]-[11].

In this paper we briefly describe related work in Section II, and discuss the basic configuration of the proposed system architecture in Section III, and the basic characteristics of the proposed system in Section IV. The uniformity of the distribution of file fragments is discussed in Section V, the secure and secret decentralization of the proposed system in Section VI, the encryption and spatial scrambling performance in Section VII, user friendly service level assurance in Section VIII, and an example of a substantial system in Section IX. Finally, we provide our conclusions from these studies in Section X.

II. Related Work

Conventionally, leased lines have been adopted as the method of connection between large-scale data centers and the data GRID recovery center for a large-scale file backup [12]. In most cases, all the technologies used are based on duplication, such as duplication of a data center, duplication of an access line or duplication of a transit line, etc.

However, considering that an earthquake may cause fiber cable failures over a wide area and shut down several communication links, and also destroy the backup data, redundant BGP (Border Gateway Protocol) routing needs to be improved in order to divert traffic along redundant backup paths by making use of effective collaboration among different operators [13].

At present, it is not easy to apply such pre-arranged collaboration when an unexpected disaster occurs. Although a reliable peer to peer communication system using Grid computing environments has been proposed, and effective scheduling algorithms have also been proposed, the required efficient and safe encryption mechanisms have not yet been developed [14]-[17]. Previously, the concept of file data fragmentation and dissemination technologies have been investigated [18]-[20]. Other related studies include the concept of a distributed file backup system [21][22].

However, in these studies, the secure and fast uniform distribution of a fragmented data file after the spatial scrambling and the subsequent encryption technology by effectively making use of a series of time dependent metadata have not yet been sufficiently clarified.

Furthermore, the service providers could not make use of the meta-data containing the history of encryption key code sequences for the secure file back up, so far. In addition, a user friendly data backup concept related to the required security strength level and recovery rate parameters has not been taken into consideration at all.

III. Basic Configuration of the Proposed System Architecture and Its Functions

The proposed file backup mechanism consists mainly of three network components and two VPNs as shown in Figure 3. The main functions of the proposed network components which are Data Center, Client Nodes and Supervisory center, are described as follows.

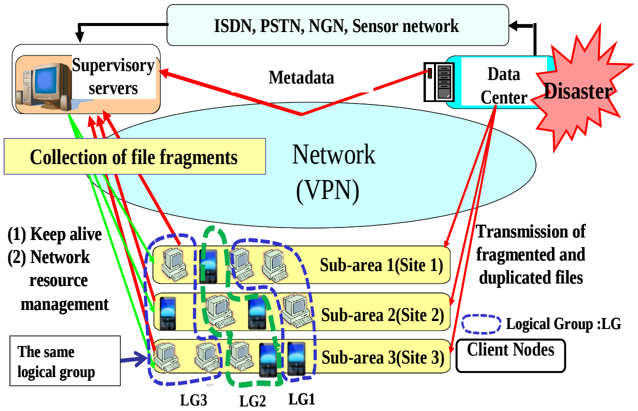


Figure 3. Basic network configuration and interfaces

The client nodes are connected to a data center and to a Supervisory center by virtual private networks (VPNs) and classified into three logical groups, such that copies of the same file fragment files are located redundantly in different geographical areas.

The Supervisory service center acquires the history data composed of the encryption key code sequence (metadata) from the Data Center via the VPN. The precise functions and the corresponding data back-up procedures are described in more detail in section IV. As shown in Figure 3, the basic procedure is as follows.

(1) The Data center sends the fragmented file to some of the sub-areas.

(2) The Data center sends the metadata used for deciphering the series of fragments. Metadata are composed of encryption key sequences, one for each file fragment.

(3) The Supervisory servers collect the required fragments from some of the sub-areas.

(4) After all the required data has been collected, decryption is executed.

Figure 4 shows the proposed system in more detail. The functions of the main components are as follows.

A. Data Center

The main task of data center is the encryption and distribution of important file data. When the data center receives file data, it processes it with the following sequence.

1)1ST ENCRYPTION

The data center encrypts that whole file using a stream cipher code. In addition, the data center records the key of 1st encryption as “encryption metadata”.

2) Spatial random scrambling

After 1st encryption, the data center carries out spatial random scrambling. Spatial random scrambling spreads data words spatially across the whole file.

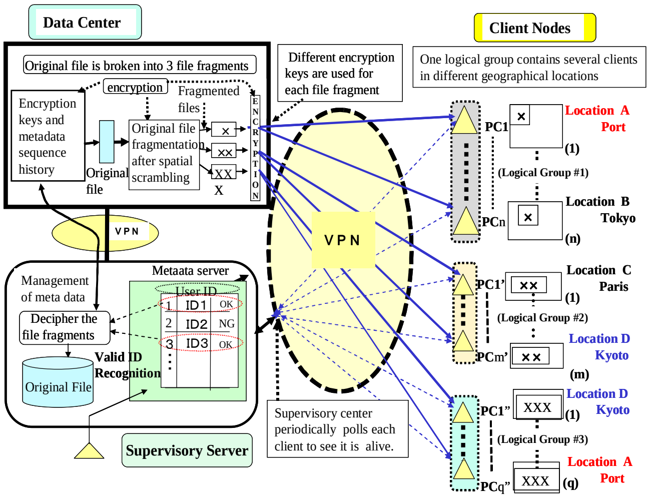


Figure 4. Basic configuration of the proposed system

3)Fragmentation

After spatial random scrambling, the data center divides the file data into a number of fragments. The data center records the size and offset of all fragments as “encryption metadata”.

4)Duplication

Next, the data center makes a number of copies of all fragments.

5)2nd encryption

The data center encrypts all the duplicated fragments with different keys.

So, if there are X fragments, and Y copies are made of each fragment, the number of 2nd encryption keys is X×Y.

The data center records all the 2nd encryption keys as “encryption metadata”

6) Distribution

The data center distributes all the duplicated and encrypted fragments to client nodes in random order. The data center records all relations between duplicated fragments and client nodes as “encryption metadata”.

7) Backup encryption metadata

Finally, the data center sends the “encryption metadata” to supervisory servers, and can then erase all data including original the file data, temporary data and encryption metadata. So, after this process, no one can steal or recover the original file data from the data center.

The original data can be deleted at the data center. It implies that the data center does not hold a copy of the original data at all. In this case, the fragmented data would be the only copy of the data, and would need to be used every time anyone wanted to access the data, not just in the event of a data center being damaged in a disaster.

B. *Client nodes*

Client nodes have four tasks: (1) receiving fragments of fragmented files, (2) deletion of old file fragments, (3) sending keep alive messages, and (4) delivery of stored fragments for transmission.

1) Receiving a file fragment

Client nodes receive duplicated fragments from data center, and store them in their exclusive storage area for this mechanism.

2) Deletion of old file fragments

As a client node's storage area is limited, client nodes have to erase old duplicated fragments.

3) Notifying presence to keep alive

Client nodes periodically send "keep alive" packets to the supervisory servers.

4) Transmission for delivery

If a client node receives a transmission request from a supervisory server and authenticates it successfully, it transmits the specified duplicated fragments to the supervisory server that issued the request.

Generally, the selection of client nodes will be determined by the contract with the service providers or selected among the public organizations or the specific corporation.

C. *Supervisory servers*

Supervisory servers have three tasks, (1) receiving "encryption metadata", (2) receiving keep alive packets, and (3) recovering of the original file.

1) Receiving "encryption metadata"

The data center sends "encryption metadata" to the supervisory servers at the end of the back-up process. The supervisory servers receive such "encryption metadata" and store it in their databases.

2) Receiving "keep alive" packets

The supervisory servers receive "keep alive" packets from client nodes, and store information about which nodes are alive in their databases. The supervisory servers send this information about working nodes to the data center to prevent it from trying to connect to dead client nodes. Moreover, the supervisory servers use this information to select appropriate client nodes at the time of data recovery.

3) Recovering of original file

If a user needs to recover back-up file data, he sends a request for file recovery to the supervisory servers. Every requests are sent to all supervisory servers. The supervisory servers send transmission requests to appropriate client nodes, according to the information on which nodes are alive. When the supervisory servers have received all the requested fragments, they start to decrypt the data using the "encryption metadata" in their database. When the data center should back up a series of a user's updated data files, which are composed of several generation files for example, the supervisory servers are applied to recover the distributed files of the corresponding generation even when the disaster has not happened in order to constantly ascertain the corresponding file's safety.

In the backup mechanism, the three network components are connected to each other via VPNs. The VPN between the data center and client nodes, and that between client nodes and supervisory servers can use the Internet. However, the VPN between the data center and the supervisory servers should use a leased line for security reasons.

IV. Basic characteristics of the proposed system

A. *Overview of the required functions*

This section discusses the characteristic functions in the proposed backup mechanism as follows.

1) Spatial random scrambling

An example of an algorithm of spatial random scrambling is shown in Fig. 5. The operator "⊕" indicated in Figure 5 means a reversible operation, such as exclusive-or, addition, or subtraction.

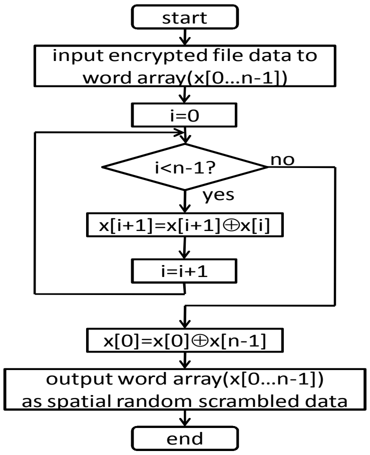

start
input encrypted file data to word array(x[0...n-1])
i=0
i<n-1?
no
yes
x[i+1]=x[i+1]⊕x[i]
i=i+1
x[0]=x[0]⊕x[n-1]
output word array(x[0...n-1]) as spatial random scrambled data
end

Figure 5. Algorithm of spatial random scrambling

In this case, a reversible operation (three kinds of operation being exclusive-or, binary addition or subtraction) for each successive word is executed for the entire file. This computation process should be repeated several times. The kind of operation can also be changed each time. The selection of operations and the number of repetitions are kept as metadata. It is strongly recommended to repeat this process several times. To de-scramble, it is only necessary to perform the same operations in the reverse order. By introducing the above mentioned spatial random scrambling technology, deciphering by a third party by comparing the encrypted fragments and the original plain text becomes almost impossible. That is, deciphering the original data become almost impossible, because of the random distribution of fragments introduced by this spatial random scrambling.

2) Fragmentation

One of the innovative ideas of this backup mechanism is combination of fragmentation and distribution in random order. Even if a cracker captured all raw packets between data center and client nodes, it would be tremendously hard to assemble all the packets in the correct order, because it would be necessary to try about (no. of fragments)! attempts.

Furthermore, the proposed backup mechanism duplicates each fragment and encrypts all fragments with different encryption keys. Therefore, no one can identify even one encrypted fragment and it is not possible to identify the other encrypted fragments. Crackers would require innumerable attempts to decipher. Since in a block cipher, the data is divided into a number of blocks, the processor and memory cost increase exponentially when the data volume increases. However, in a stream cipher, all input data are operated on bit by bit using a simple arithmetical operation, and high-speed processing can be easily achieved. Figure 6 explains the qualitative cost/performance comparison with the conventional encryption system.

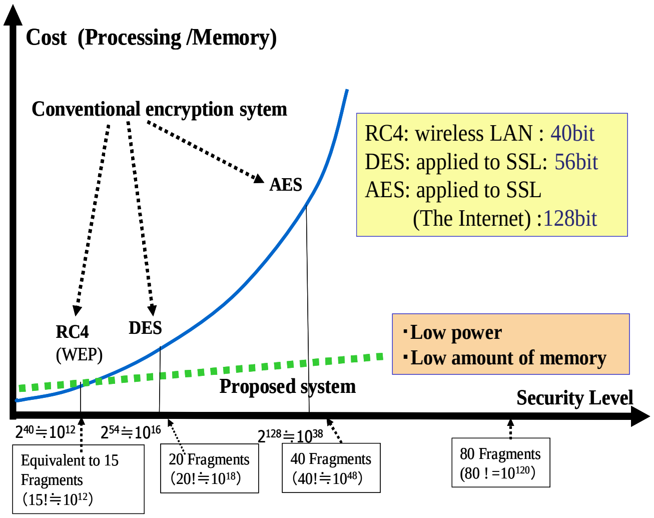


Figure 6. Cost/performance comparison with conventional encryption system

3) Duplication

The main purpose of duplication is to make possible to improve the probability of recovery. The probability of recovery rate can be estimated from the following equation. Here, P is the failure rate of each client node, n is the degree of duplication of each fragment, and m is the number of fragments.

$$\text{Probability of recovery} = (1-P^n)^m \cong 1-mP^n$$

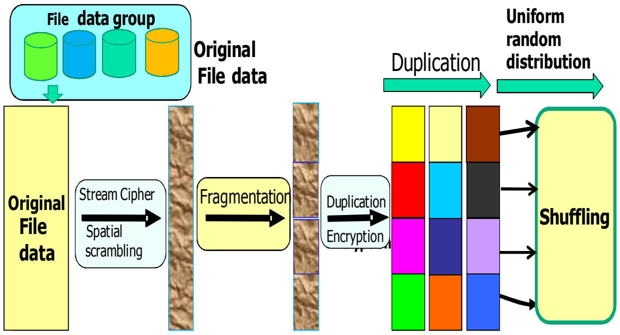


Figure 7. Sequence of a series of data processing

As mentioned above, the degree of duplication is quite important for ensuring a rigid security level, since the recovery failure rate is dependent on P^n. After sequential random scrambling, a data center divides the file data into a number of fragments. In addition, it records the data size and offset of all fragments in "encryption meta data".

Figure 7 shows the sequence of processing such as "encryption by stream cipher, fragmentation, duplication and second encryption".

As shown in Figure 8, whenever the original data is processed in the data center, then the corresponding items of metadata (Metadata#1, Metadata#2, Metadata#3) are produced sequentially. The size of this metadata is much smaller than the original data, in the order of 10^{-4} or 10^{-5} times smaller. However, it needs to be secretly and accurately transferred to the several supervisory centers in order to decipher the original data in preparation for the situation where a disaster occurs.

Figure 9 shows the calculation results of the file recovery rate, on the condition that each file fragment's failure rate P is assumed to be 0.2. For example, if the original file is divided into 30 fragments, and 30 copies are made of each fragment, then, the recovery failure rate becomes less than 10^{-19}, which is commercially available high reliability.

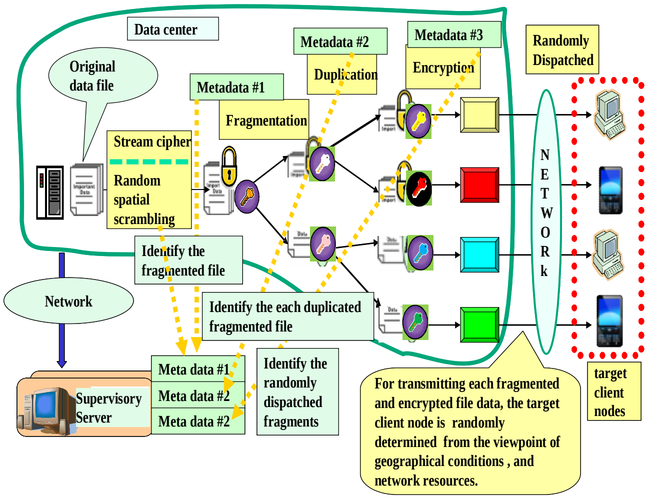

Data center
Original data file
Metadata #1
Metadata #2
Metadata #3
Fragmentation
Duplication
Encryption
Randomly Dispatched
Stream cipher
Random spatial scrambling
N E T W O R k
Identify the fragmented file
Network
Identify the each duplicated fragmented file
Meta data #1
Meta data #2
Meta data #2
Supervisory Server
Identify the randomly dispatched fragments
For transmitting each fragmented and encrypted file data, the target client node is randomly determined from the viewpoint of geographical conditions , and network resources.
target client nodes

Figure 8. Metadata transfer to the supervisory centers

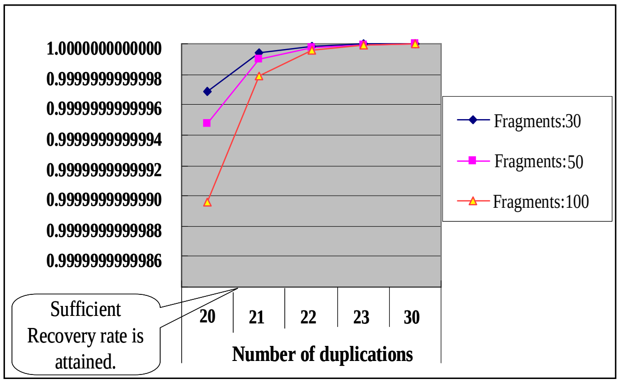


Figure 9. File recovery rate characteristics

4) Distribution

From a disaster recovery point-of-view, the duplicate copies of fragments are distributed over a wide area, in preparation for the case where a disaster is concentrated in a small area. To achieve this, the proposed file backup mechanism classifies client nodes into certain logical groups. The same logical group contains several client nodes in different geographical locations. One set of duplicated fragments which have been derived from the same fragments have to be distributed to several client nodes in some logical groups since each fragment is sent to more than one client node in each logical group because of the risk of some nodes not being alive when recovery is required. This rule ensures a secure geographical distribution, as far as possible. It is very important to ensure the uniformity of distribution, and the details of this are discussed in Section V.

D. *Characterisutics of the required funcitionss*

In terms of file data fragmentation, a rough estimate of the security level when the proposed data backup technology is used may be obtained as follows.

Case (1)

If a data file is divided into 40 fragments which are distributed to various client nodes, the number of combinations to be checked in order to reassemble the file is about $40! \fallingdotseq 2^{160}$.

This is equivalent to the AES (128 bits) cipher code.

Case (2)

If a data file is divided into 80 fragments which are distributed to various client nodes, the number of combinations to be checked in order to reassemble the file is about $80! \fallingdotseq 2^{400}$.

This is more secure than the current AES (256 bits) cipher code and such reassembly cannot be effectively realized at present.

Case (3)

When a data file is divided into 80 fragments which are duplicated to 10 copies, the number of combinations to be checked in order to recompose is about ${}_{800}P_{80} \fallingdotseq 2^{770}$. This is more secure than with the AES (512 bits) cipher code and such reassembly cannot effectively be realized in the near future. On the other hand, the number of duplications should be carefully determined from the viewpoint of the data recovery rate and the client node failure rate and the required recovery rate of each file fragment.

In terms of file data duplications, a rough estimate of the recovery rate when the proposed data backup technology is used may be obtained as follows.

Case (1)

If an acceptable file capacity for each client node is assumed to be 5 Mbytes, the original 100 Mbytes file needs to be fragmented into at least 20 files. If the duplication degree of each file fragment is 10 and each client node's failure rate is 0.2 at worst, then the probability that the original file can be recovered is assumed to be about 0.999998, which is a very high recovery probability.

Case (2)

Let us consider the case that the file data is fragmented into 80 files. When the duplication degree of each file fragment is 10 and each client node’s failure rate is 0.2 at the worst, then the probability is assumed to be about 0.999992.

V. Uniformity of Distribution Table

A. *Background*

When we distribute the data of the disaster recovery system to widely dispersed sites, an appropriate distribution table must be calculated. When we calculate the great number of the table, if the entire set of the distribution tables is biased, there exist weak points in the corresponding recovery system. To avoid these problems we examined two shuffle algorithms and two random number generators.

B. *Shuffle algorithms*

At first, we use the shuffling method "Simple shuffle"; here RNG means some random number generator.

```
for j := 0 to m - 1 do d[j]:=j;
for j :=0  to m-1 do
begin
  a:=RNG mod m;
  b:=j mod m;
  x:=d[a];
  d[a]:=d[b];
  d[b]:=x;
end;
```

Next, we use the shuffling method "Fisher-Yates shuffle" as follows [23][24]. The Fisher--Yates shuffle is unbiased, so that every permutation is equally likely.

```
for j := 0 to m - 1 do d[j]:=j;
for j := m-1 downto 0 do
begin
  a:=RNG mod (j+1);
  b:=d[j];
  d[j]:=d[a];
  d[a]:=b;
end;
```

C. *Pseudo random number generators*

We selected two random number generators. One is MT (Mersenne twister), and the other is the additive random number generator as follows.

```
arrn:=100;
adata:array[0..arrn-1] of LongWord;
rct:integer;
procedure addRandomize;
var
  i:integer; x:LongWord;
begin
  Randomize;
  rct:=0;
  for i := 0 to arrn-1 do
    adata[i]:=Random(65536)+(Random(65536)shl 16);
end;

function addRandom(m:LongWord): LongWord;
var
  a:integer;

begin
  rct:=rct mod arrn;
  a:=(arrn+rct-1) mod arrn;
  adata[rct]:=adata[rct]+adata[a];
  Result:=adata[rct] mod m;
  inc(rct);
end;
```

Note that we can realize fast implementation for the additive random number generator in if $arrn = 2^k$.

D. *Goodness of fit testing*

By using MT or the additive number generator as RNG, and in addition, by using Fisher-Yates shuffle or Simple shuffle as shuffling algorithms, and setting the division number to be 100, we calculated the distribution table 10^8 times for each method. We counted the frequency of the array index which stored the sampling number 50from 100 samples, for example.

For example, under the method of MT and Fisher--Yates shuffle of 3 cycles with division number 100, we observe the sampling number 50, and count up the frequency distribution as follows.

999517,1001965,999602,999831,1000230,999741,1000156,999623,999613,999519,1001023,999375,999689,999980,997614,1002169,1000105,999576,1000608,1001074,1000407,1000744,999558,998516,1002049,1000106,999233,1000582,999225,999697,999109,999216,999520,1000540,100056,999752,999012,1002717,1000808,999358,999477,998880,1000632,999688,999596,999843,998754,998856,999658,1000298,999671,999637,1000480,1000065,1001297,1001273,1001072,999980,1000616,1000838,1000892,999637,997561,999888,1000670,1001821,1001103,1000546,999836,1000647,998212,1000111,998973,999074,1000025,998321,1000037,1000016,999397,999210,1000867,998826,1000179,1001797,1000146,1000740,1000043,1000651,1001629,999275,999771,999762,999028,1000959,999904,999932,999305,999523,1000119,999236.

The null hypothesis for goodness of fit testing is set as follows.

H_0: These frequency tables have uniform distribution.

When we divide the data to be backed up into 100 fragments, the results of the significance level are shown in TABLE I. Here, "MTFYS3r" means that RNG is Mersenne Twister and shuffle is Fisher-Yates with 3 rounds, and "addSimp3r100" denotes that RNG is additive, shuffle is Simple with 3 rounds with the condition that arrn=100.

TABle I. Significance level comparison

MTFYS3r	MTSimp3r	MTFYS1r
0.8416	0.6746	0.4617
addFYS3r100	addSimp3r100	addFYS1r256
0.842	0.03372	0.2877

In calculating a distribution table for the disaster recovery system, we should use Mersenne Twister as the RNG and the Fisher-Yates shuffle as the shuffling algorithm. If we use the additive random number generator, we should use the Fisher-Yates shuffle with 3 rounds.

VI. Secret decentralization of supervisory center

As only supervisory servers contain the “encryption metadata”, the supervisor servers must also take account of an unexpected natural disaster, a cyber-terrorism attack, or information leakage.

One useful solution is to set up several supervisory servers in different geographical locations, and to ensure that all of the supervisory servers have the same copy of the “encryption metadata”.

This solution is acceptable for protection from a natural disaster, but it increases the possibility of security incidents. For instance, all the backup data could be stolen if only one supervisory server was cracked.

As an alternative, the proposed solution for the distributed file data back-up mechanism has the following distinctive features as follows.

1) The separation of the “encryption metadata” database (DB) and the alive/valid information DB.

2) The introduction of a secret sharing scheme in the data center.

The supervisory server for maintaining the keep alive or valid information DB has to be Internet reachable and must always be waiting for keep alive packets from all client nodes to ascertain that some of them are valid..

On the other hand, the supervisory server for “encryption metadata” DB usually communicates only with the data center.

Therefore, the selective and effective separation of several DB servers is reasonable.

We should clearly take the different logical functions assigned to each DB into account. For instance, alive information DB servers only have lists of client nodes, and crackers are unable to decrypt the backup data in client nodes.

In addition, since "encryption metadata" DB servers are usually not Internet reachable, no cracker can intrude directly, except for an insider.

The "encryption metadata" size is about m×n×(length of encryption keys), when a data file is divided into m fragments and n copies of it are made. This means that the resource cost and maintenance cost of realizing the secret sharing scheme with "encryption metadata" will be quite low.

As a result, it is appropriate that the proposed system introduces the secret sharing scheme in the data center. Before sending "encryption metadata" to the supervisory servers, the data center processes the secret sharing scheme and creates some functional shares. Then, the data center sends each shared information file to several different supervisory servers. After the distribution of duplicated fragments and the sending of shared information files, it is quite difficult to find out a series of "encryption metadata" by itself in the proposed system.

From a disaster recovery point of view, the secret sharing scheme with some appropriate thresholds should be introduced in the proposed system. If the system uses a (3, 5) -threshold scheme, the system needs five supervisory servers, and the system can tolerate the simultaneous failure of two servers.

On the other hand, from a cyber terrorism point of view, if the system uses a (3, 5) -threshold scheme, a cracker has to intrude at least three "encryption metadata" servers and one alive/valid information server at the same time.

The configuration of the secret decentralized supervisory servers and data center is shown in Figure 10. As the proposed system uses a (2, 3)-threshold scheme, there are three "encryption metadata" share servers.

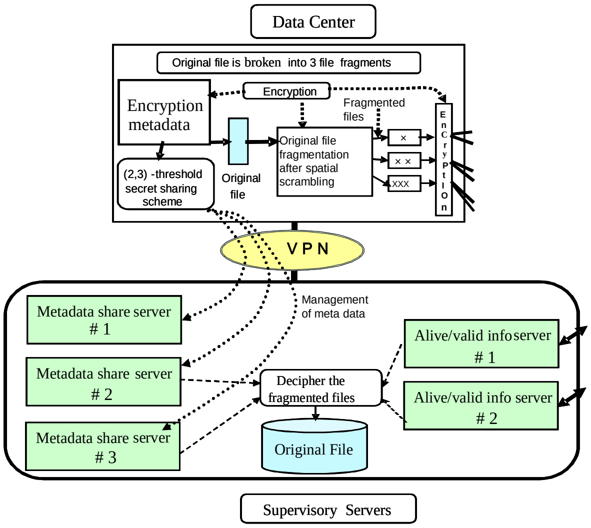

Data Center
Original file is broken into 3 file fragments
Encryption metadata
Encryption
Fragmented files
Original file fragmentation after spatial scrambling
×
× ×
xxx
EnCryPtIOn
(2,3) -threshold secret sharing scheme
Original file
VPN
Management of meta data
Metadata share server # 1
Metadata share server # 2
Metadata share server # 3
Decipher the fragmented files
Original File
Alive/valid infoserver # 1
Alive/valid info server # 2
Supervisory Servers

Figure 10. Configuration of secretly decentralized supervisory servers

As a result, there are two "keep alive" information servers. It is also possible that the Data center adopts the function of the secret sharing scheme. This unit creates some shares from the "encryption metadata". Anyone who has to recover the backup-up file data must request the corresponding recovery to the administrators of the appropriate supervisory servers. As shown in Figure 10, at least two administrators of "encryption metadata" share servers, and one administrator of "keep alive" information servers should be required at the same time. When the administrators receive the recovery request, they gather their shared DBs and alive and valid information at the same time. If the number of shares exceeds the threshold, they can reconstruct the "encryption metadata". Then, the normal recovery process starts.

VII. Performance Evaluation for Realizing Required Functions in the Proposed System

This section discusses the encryption performance and the spatial scrambling performance of the proposed disaster recovery system. We examined three systems for performance evaluation. TABLE II describes the test environment of each system.

TABLE II. SYSTEM ENVIRONMENTS

	Computer A	Computer B	Computer C
CPU	Core2 Quad Q6600 2.40GHz		Atom Z530 1.6GHz
memory	2GB(DDR2-800)		1GB(DDR2-533)
HDD	SATA 250GB 7200rpm	RAID 0(striping) SATA 500GB 7200rpm x4	IDE 40GB 4200rpm
OS	Fedora 10		
Kernel	2.6.27.5-117.fc10.i686.PAE		
gcc	gcc (GCC) 4.3.2 20081105 (Red Hat 4.3.2-7)		
libc	glibc-2.9-2.i686		

On these three systems, we tested the following data processing sequence.

(1) File I/O read process
Test program reads the original file from disk.

(2) Encryption process
It encrypts the whole of the read data using a stream cipher in memory.

(3) Spatial random scrambling process
It performs spatial random scrambling on the encrypted data, six times in memory.

(4) File I/O write process
This program writes the encrypted and scrambled data to disk.

We examined five file sizes, namely 64MB, 128MB, 256MB, 512MB and 1024MB.

Exceptionally, we omitted the 1024MB file from Computer C, because of main memory size restriction. We carried out the execution of the test program five times on each file, and thereafter evaluated the mean processing time to provide the results.

TABLE III shows the performance of encryption and spatial random scrambling.

On Computer A and Computer B, the software based encryption achieved better performance than Gigabit Ethernet class throughput. When the corresponding backup data is processed, it can be sent/received effectively by the Gigabit Ethernet class speed media. We confirmed that we can realize software based encryption while receiving the original data via a Gigabit level network interface. It was also ascertained that the low-end Atom system was able to achieve about half the performance of Gigabit Ethernet throughput with our software based encryption.

TABLE III. ENCRYPTION AND SPATIAL RANDOM SCRAMBLING SPEED

File size	Computer A [MB/sec]	Computer B [MB/sec]	Computer C [MB/sec]
64MB	203.7	203.0	73.20
128MB	203.2	202.9	72.84
256MB	203.4	203.1	73.00
512MB	203.2	203.0	74.63
1024MB	203.3	202.5	

TABLE IV and Figure 11 show the evaluation results for examining the above mentioned four individual processing steps for Computer A. In the TABLE IV, the different rows correspond to different original file sizes, and each column shows the time taken for each processing step.

In the Fig. 11, the X-axis of the graph shows the five file sizes, and Y axis means the total elapsed time including of four processing steps. The each elapsed time for different step can be identified by four colored areas. The blue colored area in the bottom stands for the file read, the red colored area, the 2nd from bottom, stands for the encryption, the green colored area, the 3rd from the bottom stands for the spatial random scrambling, and the purple colored area, on the above, stands for the file write.

TABLE IV. PERFORMANCE EVALUATION FOR COMPUTER A

Steps / File size	file read [sec]	Encryption [sec]	Scrambling [sec]	file write [sec]
64MB	0.1102	0.04898	0.2652	0.1463
128MB	0.2130	0.09848	0.5315	1.386
256MB	0.4129	0.1961	1.063	3.418
512MB	0.8257	0.3922	2.127	8.138
1024MB	3.178	0.7846	4.252	17.66

It is ascertained that in case of Computer A, the disk I/O consumes about 80 percent of total elapsed time. In particular, the file write processing step consumes most of processing time. It is also confirmed that the file write performance of the 1024MB file is about 58 MB/sec. This result is much the same as the consumer electronics HDD average performance. Therefore, it is assumed to be better to introduce an SAS drive that has a rotational speed higher than 10k rpm, RAID-0 (striping), or Solid State Disk Drive. TABLE V and Figure 12 show the evaluation results for Computer B. Computer B in which the HDD is RAID 0 configuration, showed a performance improved over that of computer A.

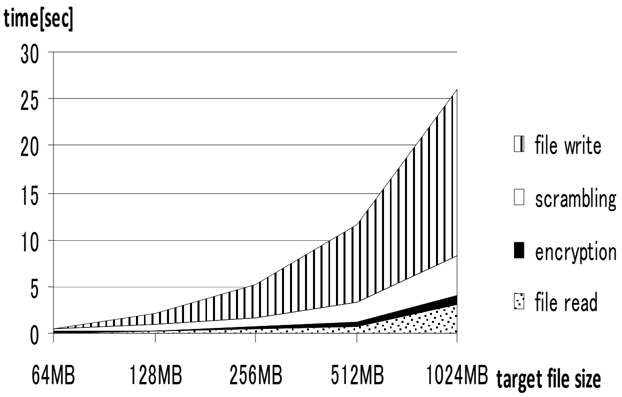


Figure 11. Processing time (v. file size) on Computer A

In TABLE V, the file write performance of the 1024MB file is about 350 MB/sec, that is about six times faster than the HDD of Computer A.

TABLE V. PERFORMANCE EVALUATION FOR COMPUTER B

Steps / File size	file read [sec]	Encryption [sec]	Scrambling [sec]	file write [sec]
64MB	0.1090	0.04914	0.2661	0.1659
128MB	0.2274	0.09878	0.5321	0.3365
256MB	0.4841	0.1967	1.064	0.6694
512MB	1.311	0.3946	2.128	1.385
1024MB	4.022	0.7982	4.259	2.955

Considering these results, the Disk I/O performance is the most important factor in the software based encryption processing system. If we should encrypt a huge file such as one larger than 100GB, it would be difficult to process the encryption and spatial random scrambling in memory. In such cases, the encryption and the spatial random scrambling would have to be done after buffering and making use of multiple disk read/write operations.

TABLE VI shows the results of total encryption processes time with multiple I/O operations within a single disk and with single I/O operation in a disk for both Computer A and Computer B. In the case of the multiple I/O operations, in the encryption processing step, the whole file is read and written as is the case with a single spatial random scrambling step process. Therefore, the total elapsed time for the encryption with multiple I/O operations within a single disk includes 1 encryption time, six spatial random scrambling times, and seven disk read and write times.

In the TABLE VI, the columns of multiple I/O operations within a single disk show the total encryption

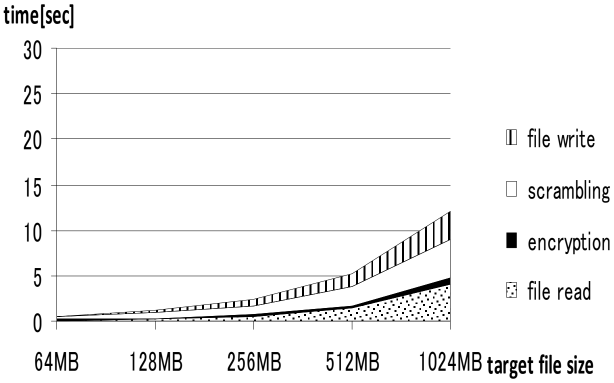


Figure 12. Processing time (v. file size) of Computer B

time using a 1MB buffer. The columns of single I/O operations in a disk show the total encryption time, read from TABLES IV and V. The total encryption time for multiple I/O operations in a single disk is five to six times greater than the total encryption time using a single Disk I/O operations on Computer A.

The total encryption time using multiple I/O operations in a single disk on Computer B is clearly improved over that for Computer A. However, the total encryption time using multiple I/O operations in a single disk for Computer B is still two to three times longer than that using a single disk I/O operations.

TABLE VI. TOTAL TIME TAKEN FOR ENCRYPTION PROCESSES

	Computer A		Computer B	
File size	Multiple I/O operations within a Single Disk	Single I/O operations	Multiple I/O operations within a Single Disk	Single I/O operations
64MB	5.375(s)	0.5707(s)	1.367(s)	0.5901(s)
128MB	14.25(s)	2.220(s)	3.151(s)	1.195(s)
256MB	30.17(s)	5.090(s)	6.515(s)	2.414(s)
512MB	62.92(s)	11.48(s)	12.48(s)	5.218(s)
1024MB	124.8(s)	25.88(s)	33.77(s)	12.03(s)

From these results, it is apparent that we should reduce the disk I/O overhead especially for the encryption of the very large file. One of the effective solutions for the reduction of the disk I/O is pre-fragmentation.

The function of pre-fragmentation is to divide a huge file into some smaller size files and so to reduce the file size appropriately so that it can be handled in the main memory before the encryption and the spatial random scrambling stages. In this way, the encryption and the spatial random scrambling function can process each part of a large file in memory and with a single disk I/O. The following results can be ascertained.

(1) The software based encryption and spatial random scrambling process is very fast if we provide a sufficiently powerful CPU and enough memory.

(2) We need to reduce the disk I/O overhead when handling very large back-up files. Pre-fragmentation of the original file is assumed to be one of the realistic solutions for reducing the disk I/O overhead. Single disk I/O is unavoidable, since we have to save the original file or encrypted file to disk temporarily.

We need to take these technical points into account, when we commercialize the proposed disaster recovery system for practical use.

VIII. USER-FRIENDLY SERVICE LEVEL ASSURANCE

It is very important to provide a prompt response to the various service levels demands from users which will be changed often to ensure reliable and economical backup systems. It is desirable that the backup systems are able to provide the versatility which is demanded by changes in service level requirements, such as security strength and/or guaranteed recovery rate, by effectively utilizing the available network resources. In this case it is necessary to provide a user-friendly Web interface to realize the above mentioned requirements.

We propose an appropriate network architecture to offer the appropriate service levels that will incorporate both security strength level and recovery rate parameters as shown in the Figure 13.

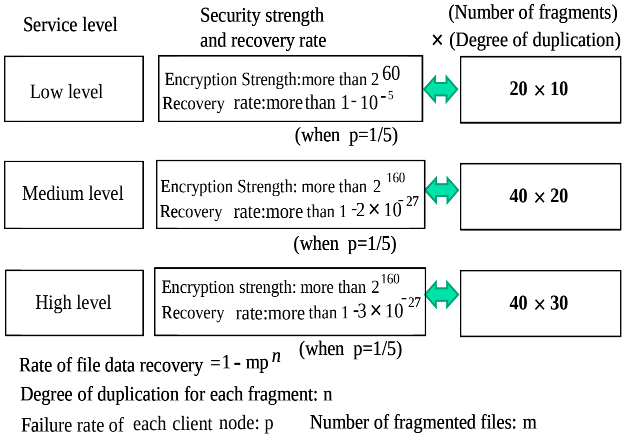

Service level
Security strength and recovery rate
(Number of fragments) × (Degree of duplication)
Low level
Encryption Strength:more than 2^{60}
Recovery rate:more than $1 - 10^{-5}$
(when p=1/5)
20 × 10
Medium level
Encryption Strength: more than 2^{160}
Recovery rate:more than $1 - 2 \times 10^{-27}$
(when p=1/5)
40 × 20
High level
Encryption strength: more than 2^{160}
Recovery rate:more than $1 - 3 \times 10^{-27}$
(when p=1/5)
40 × 30
Rate of file data recovery $= 1 - mp^n$
Degree of duplication for each fragment: n
Failure rate of each client node: p
Number of fragmented files: m

Figure 13. Disaster recovery service levels

The configuration of the service level control system is shown in Figure 14. The system consists of an RFID tag reader, the backup data storage PC of the user agent, a Web server which accepts user demands regarding the backup service level, and the Data Center which distributes the encrypted and fragmented backup file data to widely distributed client nodes including mobile PCs [25].

As the practical and highly secure means of managing user-friendly service levels requests, the RFID tag system including user ID, passwords and service levels which will be provided by the service providers is utilized effectively. In addition, a Web server which accepts the use requests and certifies user identification before informing the corresponding user requests to the Data center. It can specify one of several service levels.

Compared with the conventional method, whereby the user ID and password are input directly to the Web server by users, and which is also vulnerable to hacking by, for example, SQL injection, the above mentioned proposed access control method can safely realize ID certification using RFID tag information even if a lot of users access the Web server at the same time.

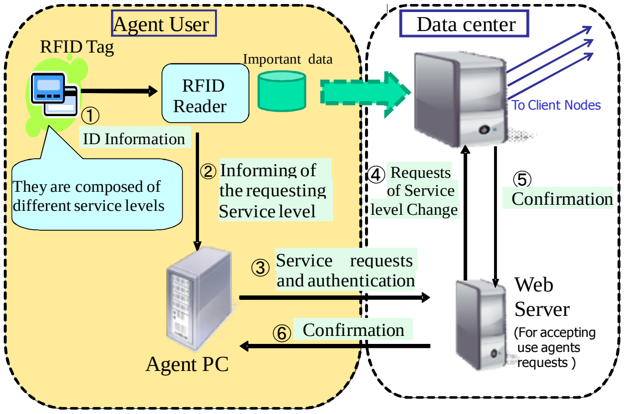

Agent User
RFID Tag
Important data
RFID Reader
①
ID Information
They are composed of different service levels
② Informing of the requesting Service level
③ Service requests and authentication
⑥ Confirmation
Agent PC
Data center
To Client Nodes
④ Requests of Service level Change
⑤ Confirmation
Web Server
(For accepting use agents requests)

Figure 14. Disaster recovery service level control architecture

In this case it is important to evaluate the average response time from the WEB server to a request from a user agent by taking both the number of simultaneous access attempts and the memory usage rate in the data center into account. The Web server frequently accesses the data center to ascertain whether the corresponding user requiring service level change is acceptable or not by considering available network resources. Generally the data center handles functions such as data encryption, fragmentation, distribution, metadata generation, etc. and so requires a much higher memory usage rate than the Web server, so we first evaluated the memory load conditions of the data center as the parameter to be studied, as shown in Figure 15.

In this case, we used an Athlon™ 643200 as the processor in the data center with 1GBytes of RAM. For the Web server, we used Celeron(R)(1.7GHz) with 512MBytes of RAM.

We evaluated the average response time per message of the Web server to ensure the user's convenience when the memory of the Data center is highly loaded (from about 94% - 95%). Consider the case where the number of user agents using the file data back up service is 1000, and 10% of users request a change in service level at the same time, as the worst cases. Taking these conditions into account, we assumed that the required request message handling capacity will be sufficient to provide 100 messages/s in both the Web server and the data center server. The experimental user request message generation rate (50~200 messages/s) was based on these conditions.

Figure 15 shows the evaluation result of the average response time of the Web server when 50~200 messages per second need to be handled at both the Web Server and the data center server. It was confirmed that the average response time of the Web server increases rapidly when the memory usage rate of the data center is 94.4% under all request message generation conditions. According to the experimental data, we found that the memory processing load in the data center should be less than 94% to ensure a prompt response time and corresponding user convenience.

Both servers can be logically separated, but can also be physically integrated into an unified server. In this case the above mentioned experimental results will still apply. From the viewpoint of security, the servers should be deployed in geographically different sites.

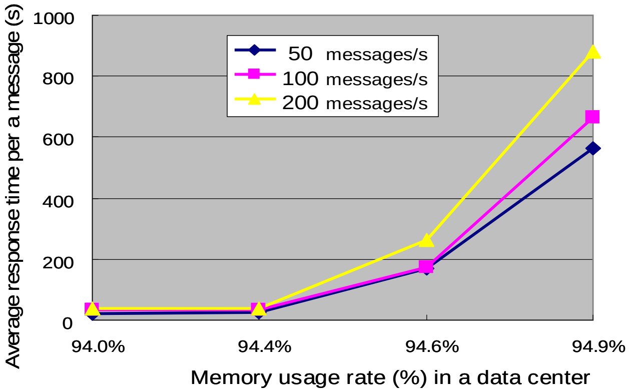


Figure 15. Average response time per message

IX. Example of a substantial sytem

The proposed system architecture can be used in a real-time secure video data monitoring service such as that of a security service company. The main features of the prototype system which we produced are as follows.

The prototype system is mainly composed of two parts as shown in Figure 16. One includes the monitoring camera (SANYO VCC-P450, H. 264), the DRT Processor (Atom N270, 1.6GHz, 1Gbyte RAM), which includes the functions of data division, spatial scrambling, encryption, and distribution and the gateway to the Internet, all deployed in the observed site. Here, DRT stands for the proposed "Disaster Recovery Technology". The interface speed between the camera and the DRT processor is 500kb/s at maximum and the gateway has a 100 Mb/s optical fiber interface. The other part has the functions of data collection, decryption and storing which are implemented in several storage servers deployed in the several surveillance centers. In the minimum configuration at the surveillance center, the DRT processor can also include the function of storage server.

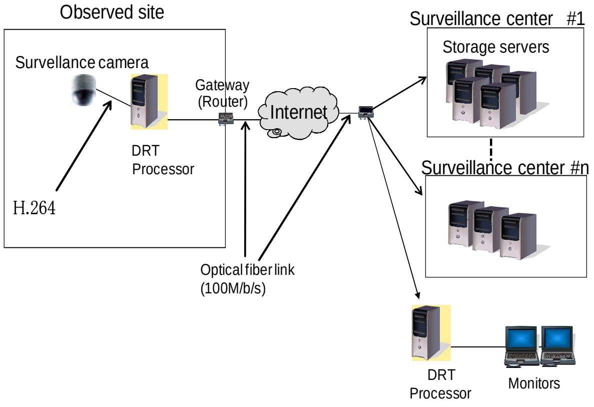


Figure 16. Example of a introduction in a practical system

The DRT processor at the observed site takes five pictures per second from the camera and transmits them continuously to the surveillance center via the Internet in real time. The size of compressed pictures in the surveillance center is about 40 K Bytes in VGA size. For efficient processing and transmission in the Internet, we fragmented the 40~50Kbytes data into 64 fragments, each of about 1 Kbyte. These fragmented data can be distributed to several storage servers in the different surveillance centers.

In Figure 17, the original data is fragmented into 64(=8 x 8) fragments by adopting 2 stages of fragmentation. Actually, in this case of the prototype DRT processor, it fragments the corresponding pictures into eight fragments and executes spatial scrambling and thereafter, sends them to 64 logically separated ports (storage servers) after an additional eight times fragmentation of each fragment.

After the spatial scrambling of the set of eight data file fragments, each file can be uploaded to eight different servers after the additional further fragmentation. In the prototype system, the corresponding fragments can be uploaded to at most 64 different servers. In the prototype described here, we did not produce duplicated fragments, for reasons of network control simplicity and to increase real time performance.

In addition, each fragment has a delivery ID which corresponds to the secret key and is created by the combination of the specific DRT processor number and IP address. The delivery ID is commonly used between the surveillance center and the observed site. In the supervisory center, the original data can be recovered by making use of the delivery ID and the corresponding password. In order to determine the destination server number to which each fragment should be uploaded, we temporarily used the random number generation “rand( )” function. The function rand( ) is called with initial value for the pointed object. Ideally, we should use the method explained in the previous Section IV. This method will be introduced in the commercial products.

When the number of users and corresponding cameras and video information increase, the number of servers should be increased and the number of fragments should be increased to ensure the firm and rigid security. This will be a topic for future study.

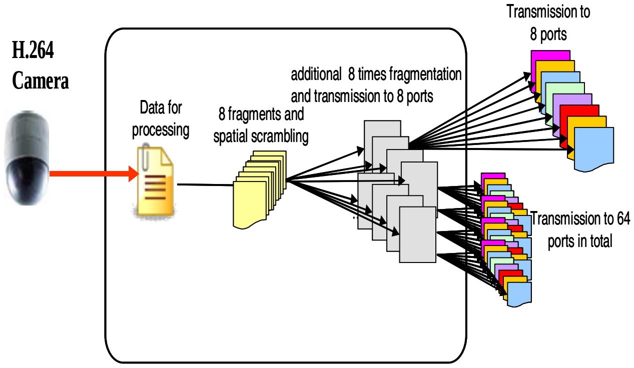

H.264 Camera
Data for processing
8 fragments and spatial scrambling
additional 8 times fragmentation and transmission to 8 ports
Transmission to 8 ports
Transmission to 64 ports in total

Figure 17. Detailed configuration of the proposed system

X. CONCLUSION

In this paper, we have proposed an ultra-widely distributed file backup mechanism implemented by effectively combining a series of technologies [KF It seems unnecessary to call all of these ‘technologies’ every time, so I have simplified this here.]including an effective stream cipher code in each encryption and distribution stage, spatial file scrambling, the random distribution of file fragments and the deployment of several secure supervisory servers . By making use of widely spread PCs, PDAs and cellular phones, a system for prompt and secure file backup can be realized economically. By making use of the proposed mechanism, illegal data recovery by tapping by a third party becomes almost impossible and an extremely safe and economically viable data back up system can be realized.

We clarified that we should use Mersenne twister as the RNG and the Fisher-Yates shuffle as the shuffling algorithm. If we use the additive random number generator, we should use the Fisher-Yates shuffle with 3 cycles.

We clarified that the proposed software based encryption achieves better performance than Giga-bit Ethernet class speed, and that a low-end Atom system is able to achieve about half the performance of Giga-bit Ethernet throughput. It has been found that reducing the disk I/O overhead for the encryption of very large files is very important and an effective solution for the reduction of disk I/O is pre-fragmentation.

We have also proposed a network architecture which can realize a user-friendly service level control mechanism using an RFID tag-reader and a corresponding Web server interface. We have demonstrated secure protocols to change a user’s required service levels. In addition, we have evaluated the average response time from the Web server by taking the memory usage rate of the data center and an appropriate number of simultaneous request messages per second into account.

We have implemented a practical disaster recovery system by using the proposed technology and clarified the system specification.

As for future research we should take into account narrow bandwidth and unreliable connections with the huge amount of communication devices in order to avoid possible inconsistency of gathered fragmented data, considering that the backup data can be dynamically updated. And in addition, an optimum network utilization technology should be introduced.

We are planning to verify the key features to fully utilize the network resources for ideal commercialized disaster recovery systems.

ACKNOWLEDGMENT

This work has been partially supported by the JST Innovation Bridge (Issue number: 08030256), Grants-in-Aid for Scientific Research (Issue number: 21560414), and Tokyo Denki University Research Center for Advanced Technologies (Q07J-13). This work was also supported in part by a grant of Strategic Research Foundation Grant-aided Project for Private Universities from Ministry of Education, Culture, Sport, Science, and Technology, Japan (MEXT), 2007-2011(07H012).

We also greatly appreciate the continuous support of all the staffs of Tokyo Denki University Center for Research Collaboration (TLO accredited).

REFERENCES

[1] N. Miyaho, Y. Ueno, S. Suzuki, K. Mori and K. Ichihara, “Study on a Disaster Recovery Network Mechanism Using Widely Distributed PC Client Nodes,” ICSNC2009, pp.217-223, Sep., 2009.

[2] S. Suzuki, “Additive cryptosystem and world wide master key,” IEICE Technical Report, 101(403): pp.39 ‑ 46, ISEC2001-84.Nov., 2001.

[3] N. Miyaho, S. Suzuki, Y. Ueno, A. Takubo, Y. Wada, and R. Shibata, “Disaster recovery equipments, programs, and system,” Patent publication 2007/3/6 (Japan), PCT Patent :No.4296304, Apr., 2009.

[4] N. Miyaho, M.Onuki and S.Kurokawa, ”Network data distributed system” JP2007/054234, Japan Patent 2007-326288 (pending), Dec., 2007.

[5] Y.Ueno, N.Miyaho, and S.Suzuki, ”Disaster Recovery Mechanism using Widely Distributed Networking and Secure Metadata Handling Technology,” Proceedings of the 4th edition of the UPGRADE-CN workshop, Session II: Networking, 45-48, June, 2009.

[6] K. Kokubun, N. Miyaho, S. Suzuki and Y. Ueno, “Disaster recovery system performance evaluation by making use of Grid computing technology”, IEICE Network system research paper, Dec., 2007.

[7] K. Kokubun, N. Miyaho, S. Suzuki and Y. Ueno, “The evaluation of the disaster recovery system using Grid computing technology”, IEICE Technical Report NS2007-106 Dec., 2007.

[8] K. Kokubun, N. Miyaho, S. Suzuki, Y. Ueno and Y. Ohno, “Study on a disaster recovery system using Grid computing technology,” IEICE Communication Society Symposium, BS-5-10, 2007.

[9] Y. Ohno, T. Iwamoto, N. Miyaho, K. Kokubun, S. Suzuki and R. Shibata, “Study on a Disaster recovery system using Grid Computing Technology,” IEICE National Convention, BS-6, 2007.

[10] K. Kokubun, T. Iwamoto, N. Miyaho, S. Suzuki, Y. Ueno and R. Shibata, “Study on a disaster recovery system using Grid computing technology,” IEICE National Convention, BS-7-193, 2007.

[11] Y. Someya, Y. Ueno, S. Suzuki, K. Mori and N. Miyaho, ” Study on a disaster recovery system by making use of GRID computing technology,” 2008 IEICE National convention, Communication Society Symposium, BS-7-1, 2008.

[12] NTT-East “Wide area disaster recovery services”, <http://www.ntt-east.co.jp/business/solution/security/dr/>06.01.2010.

[13] Y. Kitamura, Y. Lee, R. Sakiyama and K. Okamura, ”Experience with Restoration of Asia Pacific Network Failures from Taiwan Earthquake”, IEICE TRANS.COMMUN.,VOL .E90-B, NO.11, pp.3095-3103, Nov., 2007.

[14] S. Kurokawa and N. Miyaho, “Study on the Distributed Data Sharing Mechanism with a Mutual Authentication and Meta Database Technology”, APCC2007, Oct., 2007.

[15] J. Yamamoto, M. Kan and Y. Kikuchi, “Storage based data protection for disaster recovery”, *J. IEICE*, 89(9):801‑805, Sep., 2006.

[16] A. Farley, S. Zhao, and V. Lo, “Result verification and trust based scheduling in peer to peer grids”, IEEE International Conference on peer -to-peer Computing, pp.31-38, Aug., 2005.

[17] K. Sagara, K. Nishiki and M. Koizumi, “A Distributed Authentication Platform Architecture for Peer-to-Peer Applications”, IEICE Trans.Commun.,Vol..E88-B, No.3, pp. 865-872, Mar., 2005.

[18] Y. Deswarte, L. Blain, and J. Fabre, “ Intrusion Tolerance in Distributed Computing Systems”, “Intrusion Tolerance in Distributed Computing Systems”, â IEEE Symposium on Security and Privacy, pp. 110-121, 1991.

[19] Jay J. Wylie, M. Bakkaloglu, V.Panurangan, M. W.Bigrigg, S. Oguz, K. Tew, C. Williams, G. R.Ganger, P. K. Khosla, “ Selecting the Right Data Distributed Scheme for a Survivable Storage System”, CMU-CS-01-120, School of Computer Science, Carnegie Mellon University, Pittsbugh, PA 15213, May, 2001.

[20] J.Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W.Weimer, C. Weels, and B. Zhao, “OceanStore: An Architecture for Global-Scale Persistent Storage”, <http://oceanstore.cs.berkeley.edu/(2010.6.26), and http://oceanstore.cs.berkeley.edu/publications/papers/pdf , University of California, Berkeley, ASPLOS 2000, Nov., 2000.

[21] S. Tezuka, R. Uda, A. Inoue and Y. Matsushita, “A Secure Virtual File Server with P2P Connection to a Large-Scale Network”, IASTED International Conference NCS2006, pp.310-315, 2006.

[22] R. Uda, A. Inoue, M. Ito, S. Ichimura, K. Tago, T. Hoshi,, “ Development of file distributed back up system,” Tokyo University of Technology, Technical report, No.3, pp. 31-38, Mar., 2008.

[23] Fisher, R.A. and Yates, F. [1938]. Statistical tables for biological, agricultural and medical research. London, 1948.

[24] Knuth, Donald E. The Art of Computer Programming, Volume 2: Seminumerical algorithms., 3^{rd} edition, Addison Wesley. pp.142-146, 1998.

[25] N. Miyaho, Y. Ueno, S. Suzuki, K. Mori and A. Takubo, “Security level network control system” Japan Patent 2008-262704 (pending), 2008.

A Multipath Approach for Improving Performance of Remote Desktop Transmission

Cao Lethanhman, Hiromi Isokawa, and Takatoshi Kato
Systems Development Laboratory, Hitachi Ltd.
Ohzenji 1099, Asao-ku, Kawasaki-shi, Kanagawa 215-0013 Japan
{lethanhman.cao.eq, hiromi.isokawa.yt, takatoshi.kato.bb}@hitachi.com

Abstract—Remote desktop systems enable business users to access critical data located in the office from outside it with a mobile thin client or any other portable ubiquitous devices without having to worry about data leakage. However, high latency and low bandwidth in wireless-network environments may degrade the performance of remote desktop clients, such as causing slow responses to keyboard/mouse input and low speed in playing animated presentations. We therefore propose a method of transmitting data to improve the way business applications are perceived in wireless thin clients. It utilizes multiple wireless-network paths to achieve short latency in transmitting input signals and receiving responses as well as increases the throughput of large desktop images. We evaluated the performance of the proposed method using a network simulator and a real network environment. The experimental results revealed that the round trip time of a small data packet was shorter and the throughput of large amounts of data was higher than those with single network paths or existing multipath transmission algorithms.

Keywords - remote desktop; multipath transmission; wireless network; thin client

I. INTRODUCTION

All data and applications are centralized in a server, which provides a remote desktop service in remote desktop systems, such as the Citrix XenDesktop [1], Microsoft Remote desktop service [2], X-Window System [5] and AT&T VNC [7]. The remote desktop client sends input signals from the keyboard/mouse to the server and receives differences in desktop images from the server. The remote desktop client requires neither a large amount of CPU power nor large amounts of data storage, and is, therefore, usually a thin client, i.e., a resource-poor computer whose components are limited to network interfaces and input/output devices. Because the thin client does not have any confidential data stored on its local hard disk, business users can use it outside the office without having to worry about losing important data. As information leakage has been an increasingly serious problem for many enterprises, more thin clients and remote desktop systems are currently rapidly emerging.

Thin clients depend heavily on the performance of networks and remote desktop protocols have been designed for reliable networks. However, a business user who works outside the office would use a mobile thin client in a high latency, high packet loss ratio and low bandwidth wireless network environment. In these cases, there are often gaps between the requirements of remote desktop protocols and the capabilities of wireless-network services. The result is that wireless thin clients perform badly and may not reach a sufficiently acceptable level of service to be used for fast typing or presenting an animated presentation.

We improved the quality of data transmission performed by a wireless thin client using a method of multipath data transmission, where the thin client simultaneously deploys two or more wireless-network access attempts to communicate with the server. Unlike existing multipath transmission algorithms that can only be used for reducing transmission latency [13] (duplicate-and-forward algorithm) or aggregating network bandwidths [14, 15] (divide-and-forward algorithm), our proposed algorithm, called the data-aware multipath transmission (DAMT) algorithm, dynamically changes from duplicate-and-forward to divide-and-forward according to the size of the forwarded data. The DAMT algorithm can accomplish short latency in transmitting input signals and their responses and simultaneously increase the throughput of large desktop images.

We evaluated the DAMT algorithm in many network conditions created by a network simulator. The results revealed that when the thin client leveraged two network connections using the DAMT algorithm, the round trip time of a small data packet is smaller and the throughput of data is higher in comparison with single network paths or existing multipath transmission algorithms.

We also evaluated the DAMT algorithm in a real network environment. We found that a thin client equipped with one Worldwide Interoperability for Microwave Access (WiMAX) and one High Speed Downlink Packet Access (HSPA) data card could respond to key input faster while a video file was being played with a lower dropped-frame rate when it transmitted multipath data using the DAMT algorithm, compared with when only a WiMAX or an HSPA data card was used.

The remainder of this paper is organized as follows. Section 2 explains limitations with wireless thin clients. Section 3 discusses some related research. We then introduce the DAMT algorithm in Section 4 and discuss its implementation in Section 5. Sections 6 and 7 present the results from evaluating the DAMT algorithm. Section 8 concludes the paper and mentions future work.

II. LIMITATIONS WITH WIRELESS THIN CLIENTS

This section first discusses the characteristics of data that are sent and received by a thin client and it then points out the reason for the degradation in performance of thin clients in wireless network environments.

A. Characteristics of traffic in thin clients

We first investigated the characteristics of data sent and received by thin clients in remote desktop systems. Figure 1 shows the size of data that was received by a thin client when it used Microsoft's Remote Desktop Protocol (RDP) to download desktop information from a remote desktop server. As we can see, the data includes two types of data blocks, small and large. The small blocks occur when a user is typing and the large ones occur when he/she is playing a presentation. That reason for this is that when the user presses a key on the keyboard, the change in the desktop image is much smaller than when he/she plays a presentation on the remote desktop server. The desktop server only sends differences in desktop images to thin clients, instead of the entire image. Thus, the data received by the thin client in the remote desktop system can be in the form of both small and large data blocks.

We next examined the data that a thin client sends to the remote desktop server while the user is operating the thin client. As shown in Figure 2, the data only compounds small blocks, regardless of the kinds of applications the user is using. This is because the thin client only sends the input signals from the keyboard/mouse to the server.

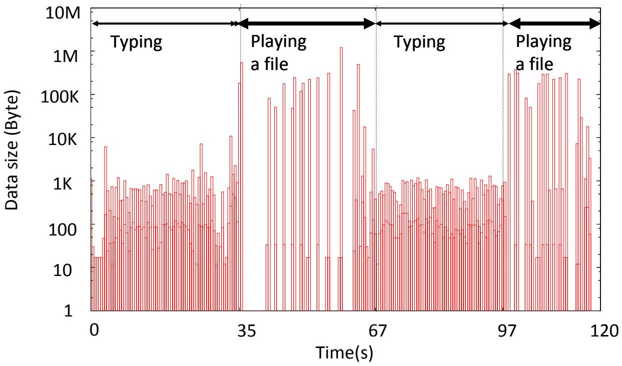


Figure 1: Data received by thin client when using RDP

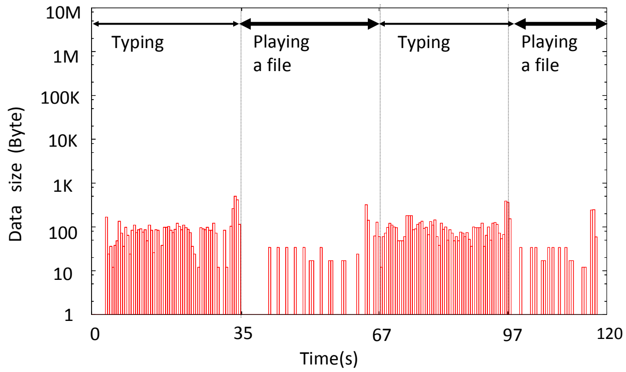


Figure 2: Data sent by thin client when using RDP

From the above-mentioned results, we can conclude that the data that are sent and received by thin clients generally include two different-sized data blocks.

1. Small data blocks (several KB): These kinds of data often appear with keyboard/mouse input. The thin client sends small-capacity keyboard/mouse input signals to the server. The server then sends small differences in desktop images (that occur when the user is typing) to the thin client as the responses to the input.
2. Large data blocks (several MB): These kinds of data appear when desktop images change dramatically, such as when animated presentations are played, desktops are scrolled, or new documents are opened.

B. Limitations with wireless thin clients

Thin clients have two main limitations with above-mentioned two characteristics of transmission data, in wireless-network environments with long latency and narrow bandwidth:

- Long latency in the transmission of small data blocks. This causes extended responses to keyboard/mouse input. The user will feel difficult typing a document because the stress because the key pressed does not appear on the screen immediately.
- Low throughput of large data blocks. This causes long delays in the time it takes to download desktop images and therefore decreases the speed at which animated presentations are played. The user of thin client will find difficult browsing web pages with animated pictures.

We will next discuss some related research on improving how applications are perceived in thin clients.

III. RELATED RESEARCH

A great deal of research related to improving the QoS of remote desktop systems has been conducted thus far. Some of this work [8, 9] has introduced the concept of localization within the context of the thin client computing model and its ability to be applied to wireless and mobile environments. Their work focused on decreasing the number of short yet frequent exchanges of communication between the thin client and the server to solve the problem of high-latency networks. However, it did not deal with the problem of narrow-bandwidth networks that occurs when the thin client transmits large data blocks. Other researchers [10, 11] have proposed QoS-aware mechanisms in remote desktop protocols. Their solution was able to compensate for temporary decreases in network performance or disruptions in transmission between a client and a server. Optimizing the use of resources by making remote desktop protocols QoS-aware, introduces efficient control over differentiated application-level traffic. Their work was similar to Distortion-based Packet Marking [12], an on-the-fly relocation of network resources in a shared link for multiple data flows with different QoS requirements. However, both the mechanisms worked on a single network path and therefore could not exceed the physical limitations of the network.

Various multipath approaches have thus far been proposed to achieve higher rates of data transmission than those in a single network. Some studies [12] have improved the robustness of data transmission by sending the same data on multiple network paths. This approach is effective in networks with high packet-loss ratios. Others have increased the throughput of data transmission by sending different parts of the data over different network path simultaneously [14, 15]. However, as we will explain in the next section, existing multipath algorithms are not suitable for data transmission in remote desktop systems that have the characteristics described in Section 2.

We proposed a multipath transmission algorithm for remote desktop systems and evaluated the performance using network simulator in [6]. This paper describes the algorithm in more details and provides some experiment results in the real network environment.

IV. Proposed Method of Data Transmission

We introduce a multipath approach to improve the data transmission of remote desktop systems. This section first introduces a TCP proxy that transmits multipath data in a remote desktop system. We next discuss our evaluations of the existing multipath transmission algorithm and then introduce our proposed algorithm.

A. Overview

We inserted a TCP proxy between the thin client and the server on both sides. The proxy programs were called multipath communication control proxies (MCCPs). The MCCPs received the data from remote desktop software and forwarded them over multiple wireless-network connections.

Figure 3 outlines an example of the placement of MCCPs in remote desktop systems. The MCCPs do not require any changes in the OS components of the server, thin client, or the remote desktop protocol. The remote desktop server passes the data to the MCCP in the server (server-side MCCP) and then the server-side MCCP sends the data over Network A and Network B to the MCCP thin client (client-side MCCP). The Client-side MCCP then passes the data it received to the remote desktop client. Thus, the MCCPs simultaneously deploy two networks (networks A and B) to send and receive data exchanged between the thin client and server.

The server-side MCCP can be installed outside the remote desktop server. As seen in Figure 4, it can work in a gateway. The gateway receives data from the server and forwards the data to the thin client over two networks using a multipath data-transmission protocol. Here, no software needs to be installed in the server while the thin client can perform multipath data transmission.

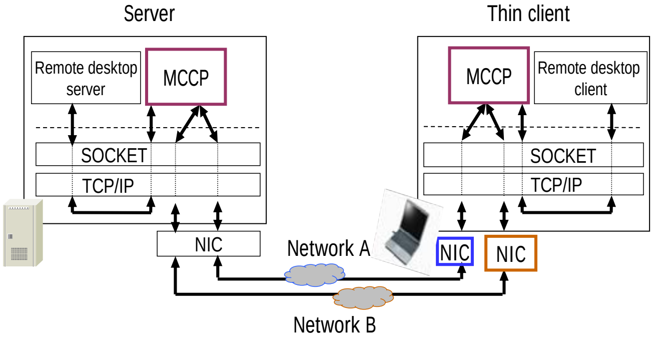


Figure 3: Placement of MCCPs

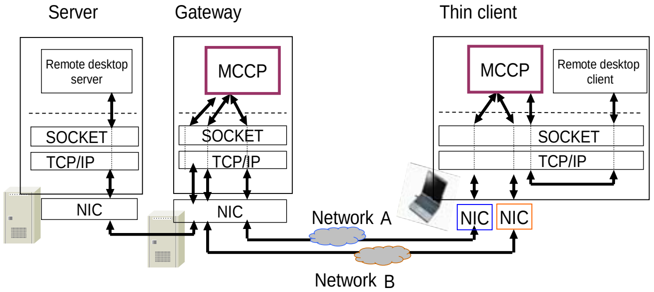


Figure 4: MCCPs with gateway

The client-side MCCPs can also be installed outside the thin client. For example, the client-side MCCP can work in a gateway that redirects the data flow from the thin client to the server. Here, the gateway will transmit the multipath data through the server instead of the thin client. It needs two network interfaces that are connected to two different networks.

B. Existing algorithms for multipath data transmission

We next discuss the algorithms for the MCCPs to send and receive data over multiple network connections. The existing algorithms in the domain of multipath communication can be divided in two types. The first decreases transmission delay and increases the robustness of transmission [13]. Figure 5 outlines the flow of data when the MCCPs deploy this algorithm. When the server-side MCCP receives data from a remote desktop's server program, it duplicates the received data and then forwards them to both networks, A and B.

The client-side MCCP forwards data that arrive sooner to the remote desktop client and disposes of the later. We called this the duplicate-and-forward algorithm. It could transmit a small data block through a network with lower latency but could not improve the throughput especially for a large data block.

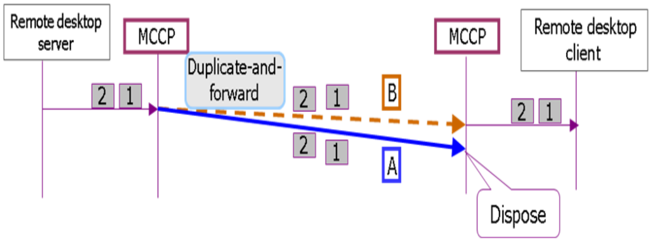


Figure 5: Duplicate-and-forward algorithm

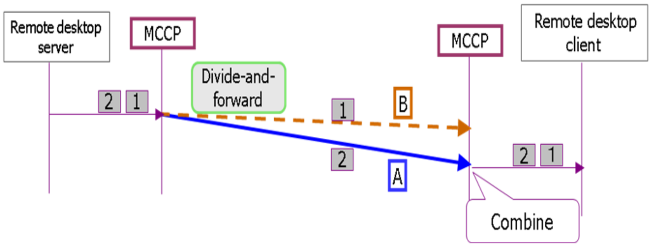


Figure 6: Divide-and-forward algorithm

The second algorithm aggregates the bandwidths of multiple network connections to increase the throughput of data transmission [14, 15]. Figure 6 shows the flow of data when the MCCPs deploy the algorithm. When the server-side MCCP receives data from the remote desktop server program, it divides the received data and then forwards different parts of the data to networks A and B. The client-side MCCP combines the data received and then forwards them to the remote desktop client. We called this the divide-and-forward algorithm. It could improve throughput especially for a large data block but did not decrease the transmission latency of a small data block.

As existing multipath transmission algorithms can only optimize transmission delay or aggregate bandwidths, they cannot be used to solve the above-mentioned two limitations with wireless thin clients.

C. Data-aware multipath transmission algorithm

We propose an algorithm called the data–aware multipath transmission (DAMT) algorithm that improves the efficiency of thin clients in wireless environments. The algorithm combines the two existing transmission algorithms: duplicate-and-forward and divide-and-forward. The DAMT algorithm can transmit small data blocks with short latency and large data blocks with high throughput. First, when a data block arrives, an MCCP runs the duplicate-and-forward algorithm. While transmitting the data, the MCCP also monitors the access network of the server. If the access network becomes congested, the MCCP determines that the block that is being transmitted is large. The MCCP then changes the algorithm to divide-and-forward. Thus, the DAMT algorithm dynamically changes from duplicate-and-forward to divide-and-forward according to the size of the data being transmitted. The details on the MCCP algorithm are described in what follows.

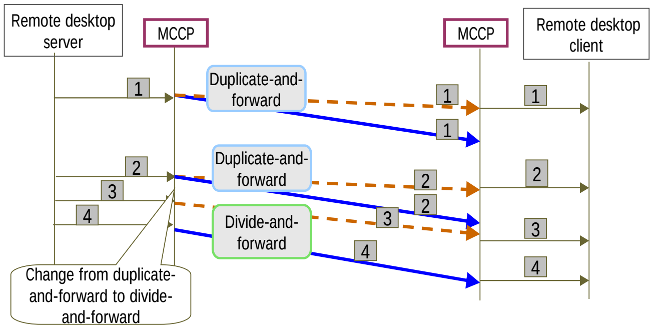


Figure 7: DAMT algorithm

Figure 7 shows the flow of data when the DAMT algorithm is used. First, the server-side MCCP receives data 1 from the remote desktop's server program, duplicates them and forwards them to both networks, A and B. Next, when a large block of data that includes data 2, 3, and 4 arrives, the server-side MCCP also duplicates data 2 and forwards them to networks A and B. Suppose that when data 2 are being transmitted, the server-side MCCP finds that the buffers for the TCP connections are full. The server-side MCCP then switches from duplicate-and-forward to divide-and-forward and performs the latter on data 3 and 4

Sender-side MCCP:

a) Wait until data that need to be transmitted arrive. Go to step *b)* when the data arrive.

b) Perform duplicate-and-forward tranmission while monitoring the buffer for TCP connections (e.g., by checking the return value of the send() function). If the buffer is full, go to step *c)*. Go to step *a)* if data transmission has been completed.

c) Perfom divide-and-forward tranmission on the remaining part of the data block and then go to step *a).*

Receiver-side MCCP:

a) Wait until data have arrived. Go to step b) when the data arrive.

b) Receive the data. Combine the divided data. Delete the duplicated data. Go to step a) when no more data arrive after a certain interval.

Thus, the DAMT algorithm uses duplicate-and-forward transmission for small-capacity data and divide-and-forward transmission for large-capacity data. This means that interactive data, which are always small capacity, are transmitted with short latency while large desktop images caused by animated presentations are transmitted at high throughput.

V. Implementation Issues

This section discusses some issues concerning the implementation of MCCPs. Our prototype was implemented in the Windows XP environment using Winsock for data transmission. We will first explain the sequence for the remote desktop client and remote desktop server to establish/close the TCP connections for transmitting desktop information. We then explain the functions of the main components in MCCPs.

A. *Establishing/closing sequence for TCP sessions*

The flow chart in Figure 8 outlines the sequence for establishing TCP sessions before the remote desktop client can exchange data with the remote desktop server. Before the TCP sessions are started, the server-side MCCP and client-side MCCP must wait in certain ports for the connection requests. First, the remote desktop client establishes a TCP session with the client-side MCCP (Step 1). If the client-side is installed at the thin client, the remote desktop client connects to a local port where the MCCP is listening. Second, the client-side MCCP establishes two TCP sessions with the server-side MCCP. In Step 2, the client-side MCCP uses network interface NIC A to connect to port X on the server and in Step 3, it uses network interface NIC B to connect to port Y. Thus, the client-side MCCP uses different network interfaces for each TCP session. In Windows, the metric parameter in the routing table of the two network interfaces A and B must have the same value, otherwise only an interface with a smaller metric will be deployed for both TCP sessions because the two sessions will have the same destination.

In Step 4, the server-side MCCP establishes a session (TCP session 4) with the remote desktop server after establishing a session with the client-side MCCP. The server-side MCCP needs to know the port where the remote desktop server is listening. This will be the connection to a local port if the server-side MCCP is located on the same machine as the remote desktop server. After TCP session 4 has been established, the MCCPs start forwarding the data exchanged by the remote desktop client and server using the DAMT algorithm.

The closing sequence for TCP sessions is as follows. When the user finishes work on the thin client and signs out, the remote desktop server will close session 4 in Step 5. The send-side MCCP then closes sessions 2 and 3 right after session 4 has closed. The receive-side MCCP will also close session 1 if one of two sessions, 2 or 3, has closed. The closing sequence will occur in the reserve direction if the user closes session 1 before signing out, e.g., by closing the remote desktop client.

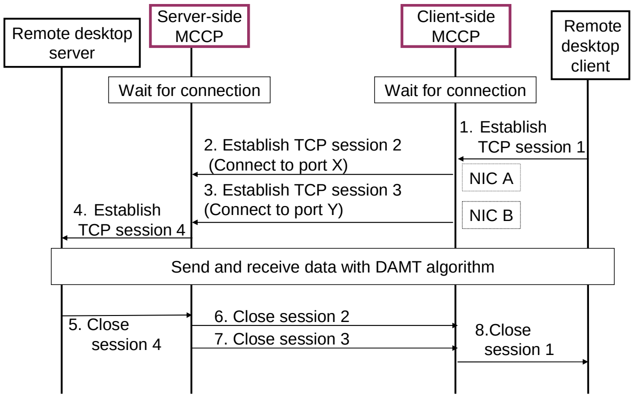


Figure 8: Sequence for establishing TCP session

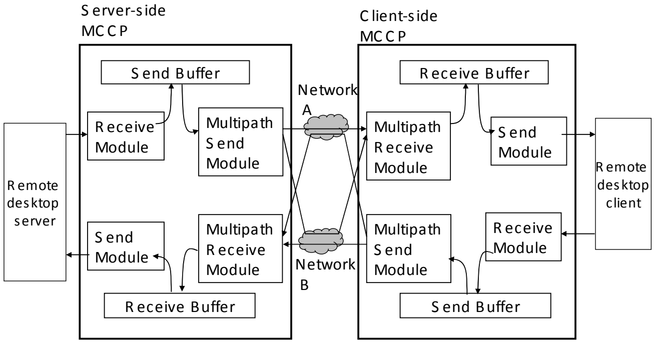


Figure 9: Structure of MCCPs

B. *Structure of MCCP*

We will next explain the components of MCCP and give details on the data processing flow of the components.

The MCCPs on the client-side and server-side have exactly the same functions. These MCCPs have five main components.

1. Data buffers

The MCCP has two buffers for storing data before it forwards them to the networks (Send Buffer) or to the server/client (Receive Buffer).

2. Receive Module

The Receive Module of the server-side MCCP receives data from the remote desktop server and saves the data in the Send Buffer. The Receive Module of the client-side MCCP receives data from the remote desktop client. If the Send Buffer is full, it stops reading data from the socket of the TCP session and waits until some space clears.

3. Send Module

The Send Module of the server-side MCCP sends the data in the Receive Buffer to the remote desktop server. The Send Module of the client-side MCCP sends the data in the Receive Buffer to the remote desktop client.

4. Multipath Send Module

The Multipath Send Module runs the DAMT algorithm. It sends the data in the Receive Buffer to the remote desktop server or remote desktop client over multipath network paths. It will not act if there are no data in the Receive Buffer.

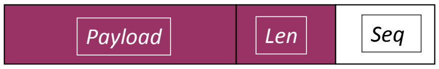


Figure 9: Program data unit (PDU)

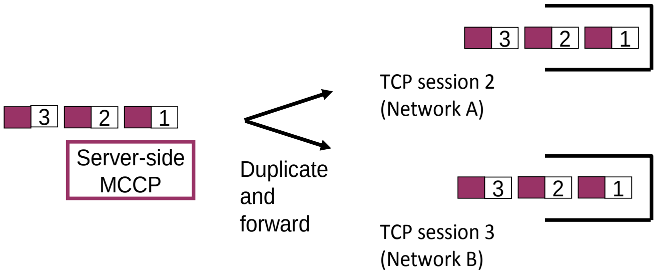


Figure 10: PDU allocation in duplicate-and-forward mode

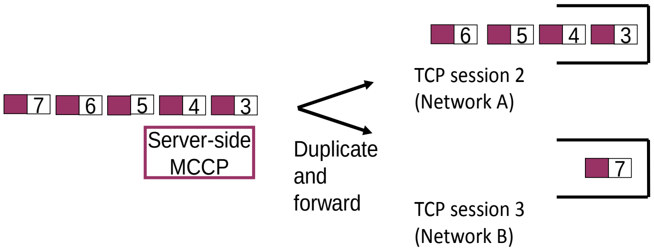


Figure 11: PDU allocation in duplicate-and-forward mode

The Multipath Send Module divides the data in the Send Buffer into small data blocks called Program Data Units (PDUs). Figure 9 outlines the structure of a PDU. Each PDU includes a sequence number (*Seq*), the length of the data (*Len*), and the data (*Payload*). The maximum size of the Payload is 500 KB. The Payload may be smaller than 500 KB if the amount of data remaining in the Send Buffer is less than 500 KB. Here, the module creates one PDU from all the data remaining in the buffer.

The Send Buffer Module continuously monitors the buffer of the two TCP sessions, 2 and 3, using select() commands. When the sessions are available for transmitting more data, the module chooses a PDU in the Send Buffer and sends it over TCP sessions 2 and 3. As previously mentioned, the Send Buffer Module first sends the PDU in duplicate-and-forward mode while monitoring the returned value of the send() function. It will change to the divide-and-forward mode if the send() function return value is smaller than the size of the PDU that the module requested to be sent, because this is the sign indicating that the TCP session's buffer is full. We will now explain how the Send Buffer Module chooses a PDU in the Send Buffer to send to the network.

In the duplicate-and-forward mode, the Send Buffer Module selects the PDU with the following sequence number: N = R + 1, where R is the largest number of PDUs that have already been sent at that time. The module sends the PDU twice, once over network session 2 and the second over session 3 (Figure 10).

However, in the divide-and-forward mode for session 2, the next PDU to be sent to the network is:
N = R + K + 1, where R is the largest *Seq* of PDUs sent at that time; K is the number of PDUs that is sent by session 3 when session 2 sends one PDU. The value of K reflects the transmission speed of networks A and B. Similarly, the PDU that session 3 will send is N = R+K'+1, where K' is the number of PDUs that is sent by session 3 when session 2 sends one PDU. Thus, the module sends data with larger *Seq* in the sessions that have low transmission speed. By doing so, the receiver can receive the data in the right order, saving time in buffering and in sorting disordered data. Figure 11 shows an example where network A is four times faster than network B. Transmitting one PDU on network B (with session 3) takes the same amount of time as transmitting four PDUs on network A. Here, the Send Buffer Module sends PDUs 3, 4, 5, and 6 in session 2 and PDU 7 in session 3. The PDUs will then arrive at the receiver in the right order.

5. Multipath Receive Module

The Multipath Receive Module receives PDUs from networks A and network B. It stores received data into the Receive Buffer. If the Receive Buffer is full, the module stops reading data from the socket of sessions 2 and 3; it will wait until the buffer becomes available for more data.

The Multipath Receive Module monitors TCP sessions 2 and 3 using the command select() of Winsock. Whenever a PDU arrives, the module first checks if the PDU is a duplicate of any PDU that has previously arrived. If the PDU is a duplicate, then the PDU will be disposed of. Otherwise, the PDU will be stored in the Receive Buffer (Figure 12). Note that the processing flow in the Multipath Receive Module does not need to change when the Multipath Send Module changes from duplicate-and-forward to divide-and-forward.

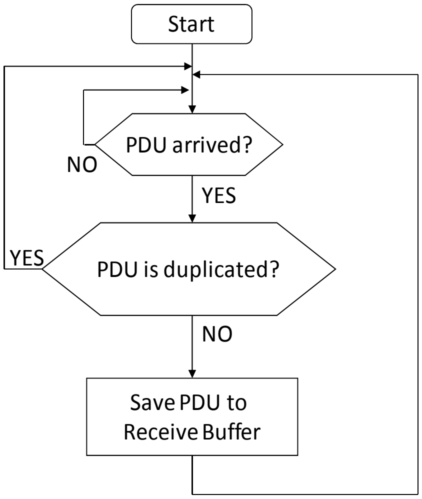


Figure 12: Algorithm for Multipath Receiver Module

VI. EVALUATION WITH NETWORK SIMULATOR

This section discusses our evaluation of the proposed algorithm in a laboratory network environment. We implemented MCCPs and installed programs on a thin client as well as a remote desktop server, as shown in Fig. 13. The simulator was a FreeBSD [3] machine with a Dummynet program [4] installed. There were two networks (A and B) in the simulator. The thin client communicated with the server through the two networks. The delay and bandwidth of the networks varied in each experiment.

A. *Evaluation of tranmission delay*

We first checked the round trip time (RTT) of a probe packet traversing between the thin client and the server when the DAMT algorithm was applied. We also investigated a case where the thin client only used network A or B to compare them. The propagation delay of networks A and B varied in a range from 100 ms to 300 ms. The bandwidth of both networks was set to 100 Mbps; this value was sufficiently large that the bandwidth did not cause additional delay in transmitting the probe packets. The settings for networks A and B are listed in Table I. The propagation delays of the two networks were set to the values shown in Figure 14.

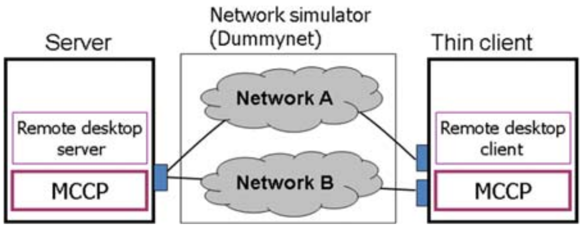


Figure 13: Network topology

TABLE I. SETTINGS FOR NETWORK METRICS

	Propagation delay (one way)	Bandwidth
Network A	Varies within range from 50 to 150 ms	100 Mbps
Network B	Varies within range from 50 to 150ms	100 Mbps

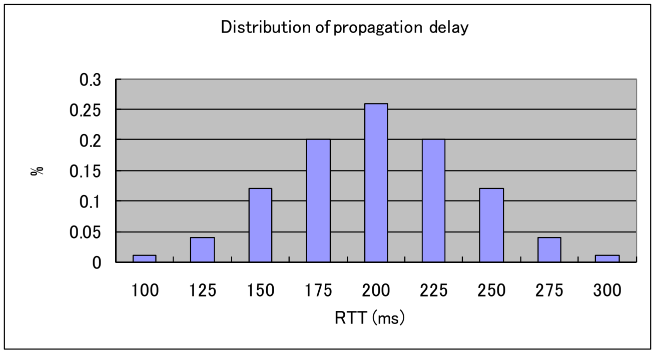


Figure 14: Distribution of propagation delay (round-trip)

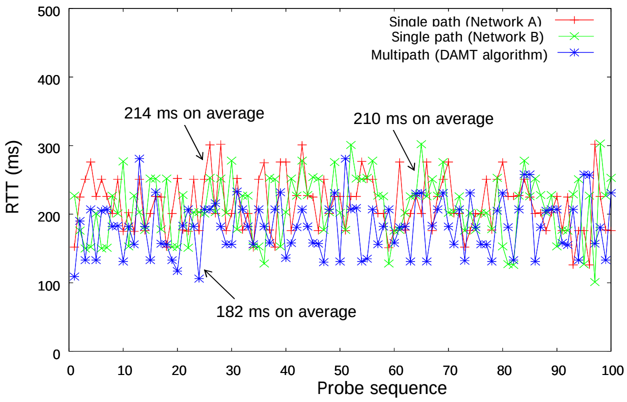


Figure 15: RTT-measurement results

The probe packet was 30 bytes and was sent from the thin client to the server and then immediately sent back from the server to the client.

Figure 15 shows the RTT values of the probe packet when it traversed network A, network B, and when it was forwarded by MCCPs with the DAMT algorithms. The RTT of the probe packets on average was 182 ms when using the multipath control algorithm, 214 ms when using network A and 210 when using network B. This means that DAMT could successfully reduce delay in the transmission of small packets.

B. *Evaluation of data-transmission throughput*

We evaluated the throughput of transmitted data in an experiment. The propagation delay for networks A and B was kept constant at 2 ms. The bandwidths of both networks were set to various values such as 500 Kbps, 1 Mbps, 1.5 Mbps and 2 Mbps (See Table II). For each value of bandwidth, we compared the throughput for bulk data transmitted between the client and the server when a single path (network A or B) was used and when multipath algorithms were used.

TABLE II. SETTINGS FOR NETWORK METRICS

	Propagation delay (one way)	Bandwidth
Network A	2 ms	500 Kbps, 1 Mbps, 1.5 Mbps, 2 Mbps
Network B	2 ms	500 Kbps, 1 Mbps, 1.5 Mbps, 2 Mbps

The results obtained from measuring throughputs are in the bar chart in Figure 16. The DAMT algorithm always outperformed the single paths (A and B) by up to 1.9 times. We can also see the throughput for multipath transmission when existing algorithms were used. We can see that DAMT achieved the same performance as the divide-and-forward algorithm, which is proposed for an aggregate bandwidth for multiple network connections.

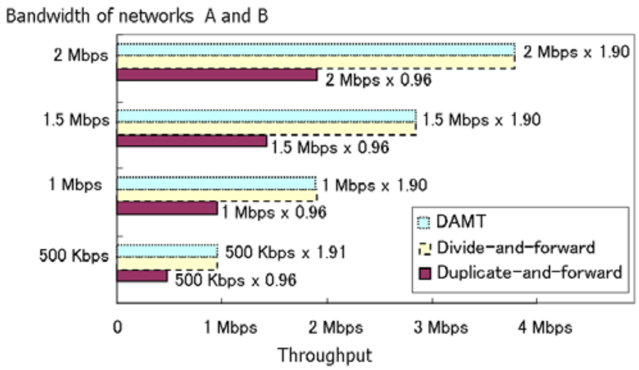


Figure 16: Throughput-measurement results

C. *Performance of thin client when using PowerPoint*

We next evaluated the performance of the thin client when Microsoft's PowerPoint program, which is a popular business application, was used. The settings for networks A and B are listed in Table III. The settings were determined according to the measured characteristics of two popular commercial mobile networks in Japan in May 2008. The remote desktop protocol used was RDP [12].

Figure 17 shows the PowerPoint file used in the experiment. The playing time of the file is about 4(s), the amount of data that was transmitted in the network when the file was being played was a total of 1626 KB.

Figure 18 is a bar chart of the average delay in response to keyboard input. This was calculated as the time from when the packet (including the input signal) left the thin client to when the first response packet returned back to it. We can see that the DAMT algorithm responded as quickly as the duplicate-and-forward algorithm.

TABLE III. SETTINGS FOR NETWORK METRICS

	Propagation time (one way)	Bandwidth
Network A	52 ms	380 Kbps
Network B	150 ms	600 Kbps

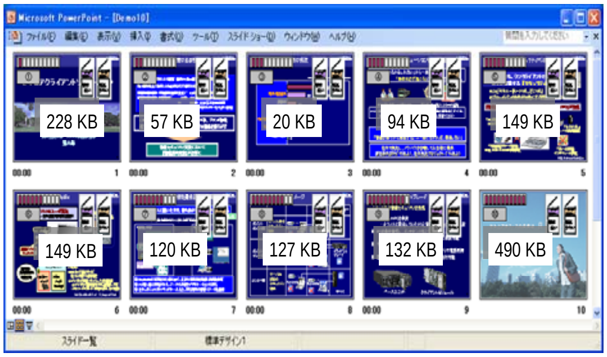


Figure 17: PowerPoint slides

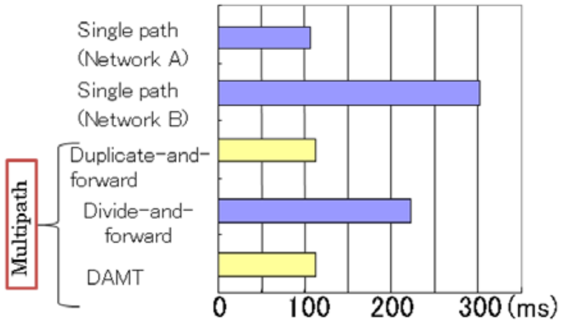


Figure 18: Response time to keyboard input

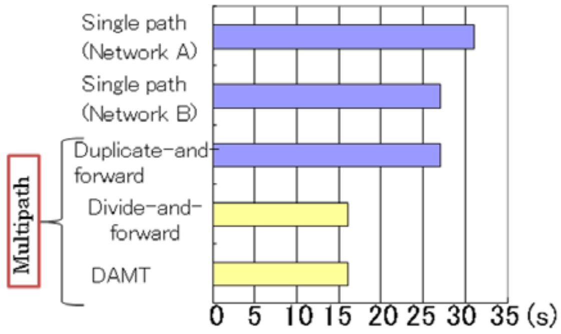


Figure 19: Time to play presentation file

Figure 19 is a bar chart that shows the time it takes to play a presentation file. The speed at which the file is played depends on the throughput of the desktop images transmitted from the server to the client. We can see that the DAMT algorithm takes the same time as the divide-and-forward algorithm. And the time is far shorter than the playing time when only network A or B is deployed.

The DAMT algorithm thus shortens the response to keyboard input to that of the duplicate-and-forward algorithm, and the speed at which the presentation file is played is as high as that of the divide-and-forward algorithm.

VII. EVALUATION IN REAL NETWORK ENVIRONMENT

We next evaluated the performance of the proposed method DAMT in a real network environment. The network used in the experiment is outlined in Figure 20. A thin client equipped with two high-speed wireless network access cards was used. The two cards had different data-transmission methods and different access networks. The first used WiMAX and the other used HSPA. The thin client was located in an office on the top floor of a six-storey office building. The thin client accessed an RDP server located in a data center through a gateway located in the same local network as the server. A server-side MCCP was installed in this gateway. The data center was connected to a WAN by a high-speed network (20 Mbps).

We first evaluated the time it took for a TCP packet to traverse from the thin client to the server and back again. This time indicated how fast a letter appeared on the screen of the thin client when the user pressed a key.

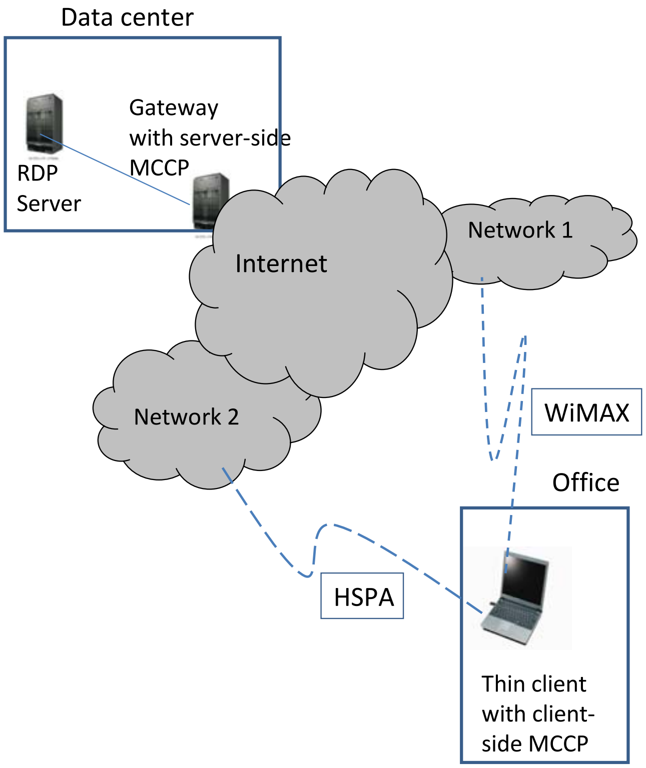

Data center
Gateway with server-side MCCP
RDP Server
Internet
Network 1
Network 2
WiMAX
Office
HSPA
Thin client with client-side MCCP

Figure 20: Experimental network

We also evaluated the amount of RDP traffic transmitted from the RDP server to the thin client when the user watched a 15-s video. This amount of data reflected the quality of the video shown on the screen of the thin client. If the amount of data was small then the video had many too lost frames. We checked the amount of data when WiMAX access and HSPA access were used singly and when DAMT was used.

Figure 21 has bar charts of the average value (in 50 experiments) of the round trip time of a TCP data packet when it used network 1 (through WiMAX access), and network 2 (through HSPA access). The figure also indicates the round trip time when the multipath algorithm was deployed. We can see that when the thin client used HSPA access, the response time was about 159 ms, and the WiMAX access was 201 ms. However, the response time when the proposed algorithm DAMT was deployed was only 139 ms. The results attained when using WiMAX access and HSPA access may change if we carry out the experiments in other places. However, as the DAMT algorithm will choose the shortest access for all data packets, we can rest assured that DAMT will introduce a shortened response time in any environment.

We next checked the amount of data that the thin client received during the time it took to play a 15-second video file. In a LAN environment, the thin client played the video with almost no difference to playing it directly. Here, the thin client received about 20 MB of data for the video file. Table 4 lists the amount of data when WiMAX, HSPA, and the proposed multipath data transmission algorithm were used.

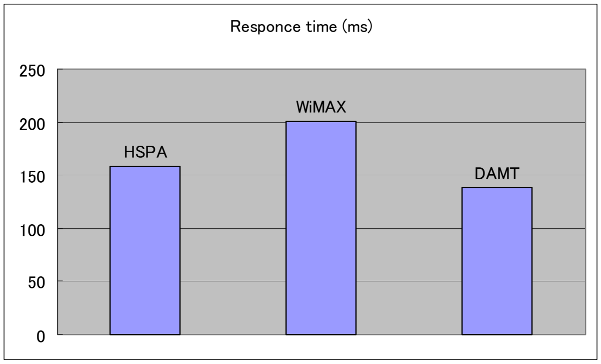


Figure 21: Average value of response time

As seen in Table IV, when using HSPA access, the thin client could receive only 21.5% of the data in comparison with the LAN environment. In WiMAX access, the ratio was 23.8%. However, the DAMT algorithm performed better when the ratio was 43.8%. The reason for the higher throughput of DAMT is that, the thin client can use the bandwidth of both wireless networks to transmit the desktop information (Figure 22). From Figure 22, we can see that DAMT deployed both network interfaces to receive data from the HSPA and WiMAX networks.

We could test and confirm the superior performance of the proposed algorithm through a real-network environment. Similar to the results we obtained in the laboratory, the DAMT algorithm could introduce short response times for small packets such as the RDP packets that occur when users are typing. As they also increased the bandwidth of the network path between the thin client and RDP server, video files could play more smoothly than when using a single wireless network.

TABLE IV. AMOUNT OF DATA RECEIVED BY THIN CLIENT

	Received data (bytes)	Compared with LAN
Network 1 (WiMAX)	4,554,858	21.5%
Network 2 (HSPA)	5,040,438	23.8%
Multipath (DAMT)	9,046,273	42.8%
LAN	21,134,323	100%

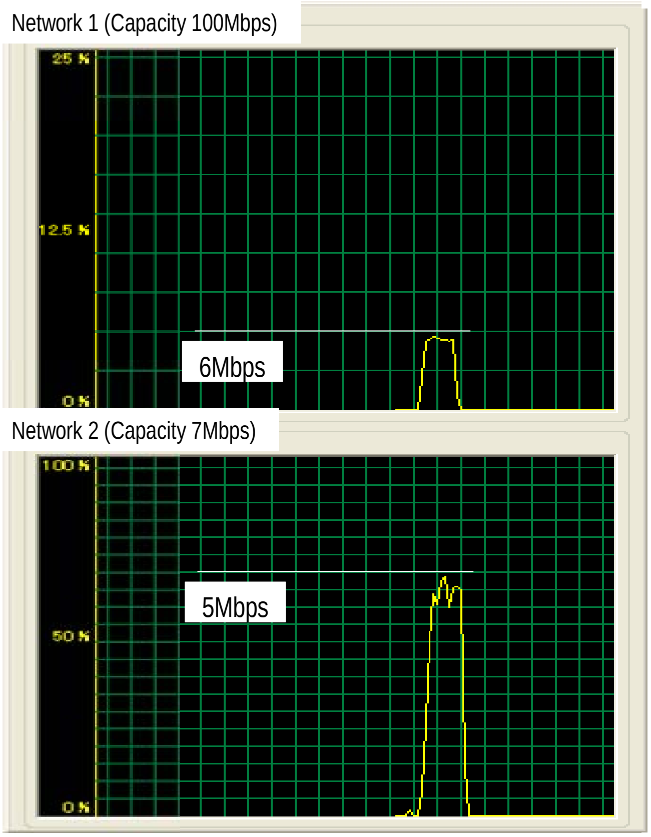


Figure 22: Data transmission on network interfaces of thin client

Also note that the thin client in the experiment was not located in the best place to achieve the best performance with WiMAX and HSPA access. The aims of the experiment were not to evaluate the performance of these wireless access networks, but just to confirm that DAMT could work well in a real-network environment. We also found that DAMT could outperform single network access by using the multipath approach.

VIII. Conclusion and Future Work

We proposed an algorithm called DAMT that leveraged multiple wireless network connections to attain short latency in the transmission of input signals and its responses as well as to increase the throughput of large desktop images.

Due to the large range of experiments in both the simulator and in a real network, we found that the thin client could quickly respond to keyboard input with the DAMT algorithm and could quickly play animated presentations.

However, some future work still needs to be done in this respect. First, we intend to carry out more experiments in real networks to find what additional benefits DAMT can provide. We intend to assess the performance of thin clients in places other than offices, such as in trains or cafeterias. Implementing MCCPs that can work over three or more wireless networks is also within the scope of future work.

Indication of Trademark Registration

XenDesktop(TM) is trademarks of Citrix Systems, Inc.
Remote Desktop(TM) is a trademark of Microsoft Corporation.
X Window System(TM) is a trademark of The Open Group.
VNC is a trademark of AT&T Laboratories Cambridge.
FreeBSD is a trademark of the FreeBSD Foundation.
PowerPoint is a registered trademark of Microsoft Corp. in the U.S. and other countries.
Windows is a registered trademark of Microsoft Corp. in the U.S. and other countries.

References

[1] Citrix Systems, http://www.citrix.com 05.05.2009

[2] Remote Desktop Protocol, http://msdn.microsoft.com/en-us/library/aa383015.aspx 05.05.2009

[3] FreeBSD, http://www.freebsd.org 05.05.2009

[4] Luigi Rizzo, Dummynet, http://info.iet.unipi.it/~luigi/ip_dummynet/ 05.05.2009

Article in a journal:

[5] R. W. Scheifler and J. Gettys, "The X Window Systems", ACM Transactions on Graphics, 5(2), 1986.

Article in a conference proceedings:

[6] ---------,"Multipath Data Tranmission for Wireless Thin Clients", Proc. UBICOMM 2009.

[7] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper, "Virtual Network Computing", Proc. IEEE Internet Computing, 1999.

[8] S. Ramamurthy and A. Helal, "Optimising Thin Client for Wireless Computing via Localization of Keyboard Activity", Proc. IEEE Conference on Performance, Computing, and Communications, pp. 249-252, 2001.

[9] C. Aksoy and S. Helal, "Optimising Thin Clients for Wireless Active-media Applications", Proc. Third IEEE Workshop on Mobile Computing Systems and Applications, pp. 151-160, 2000.

[10] M. Lubonski, V. Gay, and A. Simmonds, "An Adaptation Architecture to Improve User-Perceived QoS of Multimedia Service for Enterprise Remote Desktop Protocols", Proc. Next Gerenation Internet Networks (NGI 2005), 2005.

[11] M. Lubonski, V. Gay, and A. Simmonds, "A Conceptual Architecture for Adaptation in Remote Desktop Systems Driven by the User Perception of Multimedia", Proc. Asia-Pacific Conference on Communications, pp. 891-895, 2005.

[12] J. C. De Martin and D. Quaglia, "Distortion-based Packet Marking for MPEG Video Transmission over DiffServ Networks", Proc. IEEE International Conference on Multimedia and Expo, 2001.

[13] J. Chen, K. Xu and M. Gerla, "Multipath TCP in Lossy Wireless Environment,"Proc. IFIP Third Annual Mediterranean Ad Hoc Networking Workshop, 2004.

[14] H. Hsieh and R. Sivakumar, "A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts", Proc. ACM MobiCom, 2002.

[15] K. Park, Y. Choi, D. Kim, and D. Park, "MTCP: A Transmission Control Protocol for Multi-Provider Environment",Proc. IEEE CCNC, 2006.

Content and Type as Orthogonal Modeling Features: a Study on User Interest Awareness in Entity Subscription Services

George Giannakopoulos, Themis Palpanas
DISI
University of Trento
Trento, Italy
ggianna@disi.unitn.it, themis@disi.unitn.eu

***Abstract*—Real-word entities can be mapped to unique entity identifiers through an Entity Name System (ENS), to systematically support the re-use of these identifiers and disambiguate references to real world entities in the Web. An entity subscription service informs subscribed users of changes in the descriptive data of an entity, which is a set of free-form attribute name-value pairs. We study the design, implementation and application of an adaptable, push-policy subscription service, within a large-scale ENS. The subscription system aims to deliver ranked descriptions of the changes on entities, following user preferences through a feedback-driven adaptation process. Within this paper, we offer a novel approach based on the discrimination between the content (descriptive aspect) and the type (directly quantifiable or binary aspects) of information instances (i.e., entity changes in our case). We study and evaluate two different approaches of adaptation to user interests: one that only manages constant user preferences and one that adapts to interest shifts of the users. We evaluate the learning curve of the variations of the system, the utility of the content-type discrimination, as well as the effectiveness of modeling user behavior for random interest shifts of the user. The experiments demonstrate good results, especially in the system's content-aware adaptation aspect, which takes into account user interest shifts. We also extract a set of useful conclusions concerning the detection of user interest shifts, based on feedback, as well as the relation between the user interest shift frequency and the optimality of learning memory window size.**

I. Introduction

An Entity Name System (ENS) [2], [3], [4] is a system that aims to handle the process of assigning and managing identifiers for entities (e.g., people, locations) in the World Wide Web (WWW). These identifiers are global, with the purpose of consistently identifying a specific entity across system boundaries, regardless of the place in which references to this entity may appear.

In the *ENS* we are interested in the minimal amount of information based on which we are able to uniquely identify each entity in the system. This information is imported both automatically or manually to the system, but its update is based mostly on collaborative effort — which is also apparent, e.g., in Wikipedia.

The volume and volatility of the data contained within the *ENS* offer a challenge concerning both the maintenance and accessibility of the information. The maintenance of the entity data would be a rather expensive and cumbersome task, was it not designed as a joined effort with the *ENS* user-community. By relying on the collective wisdom and collaboration of users, we aim to induce and use their feedback in order to achieve high data quality. The premise is that users will be willing to "adopt" some entities, and make sure that their descriptions in the *ENS* are always accurate.

In support of this functionality, we are proposing and describing the adaptive aspect of an *Adaptive Entity Subscription System*, which allows a user to *subscribe* to entities of interest (e.g., one's own entity, or entity of one's home town) and get informed about *changes* on these entities. The subscription system helps clients follow the changes of entities in the *ENS* at the time they occur through asynchronous messages (e.g., via e-mail).

The personalized aspect of AESS takes into account what kinds of change are interesting for a given subscription client. The system ranks the information on changes in every sent message, according to user preferences. It further adapts to the individual use-patterns of each client and to user interest shifts, based on the information of the feedback. The aim is that the information flow to the user remains aligned to user interest. This kind of personalization provides ease of use and enhances the user's experience and interaction with the system, thus facilitating the update of and access to *ENS* data.

In the following paragraphs we elaborate on ENS and the subscription service notions to define clearly the setting of our study.

A. The Entity Name System

The *ENS* has a (distributed and replicated) repository for storing entity identifiers along with some small amount of descriptive information for each entity. The purpose of storing this information is to use it for discriminating among entities, not exhaustively describing them. Entities are described by a number of attribute-value pairs, where the attribute names and the potential values are user-defined (arbitrary) strings.

The ENS supports, through an *Access Services* layer, the search for an entity or the update of entities. The search allows the ENS clients to identify an entity based on cross-validated and updated data. Updates on the entities of the ENS repository can be performed either by inserting a new entity in the system or by changing some of the attributes of an existing entity. System administrators can even merge, split or delete entities.

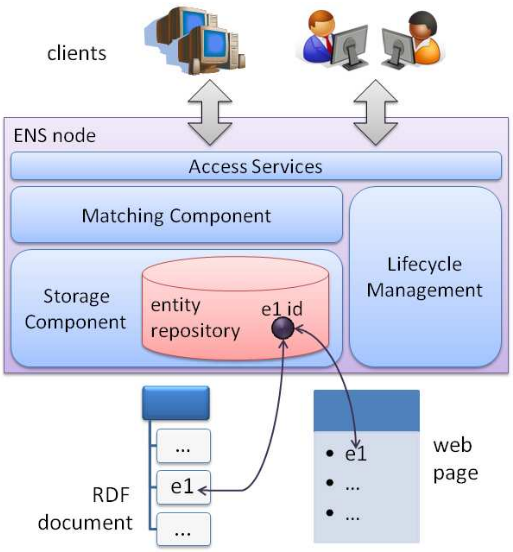


Figure 1. Schematic of the *ENS* and its interactions.

As shown in Figure 1, the end result is that all instances of the same entity (i.e., mentioned in different systems, ontologies, web pages, etc.) are assigned the same identifier. Therefore, joining these documents and merging their information becomes a much more simple and effective process than before.

Within this context, a client can be interested in receiving information on *changes* on the identifying information of a specific entity in the ENS. To this end, a client can *subscribe* to entities that are of some interest and get informed about changes on those entities. This subscription-based information service is described in the next paragraphs.

B. The Subscription Service

The administration and checking of entity information is a difficult task in itself, considering the scalability of an ENS. To alleviate the burden of such a task and to handle the information need of non-administration clients, we propose the use of a *change subscription system*.

The change subscription system helps clients follow the changes of entities in the ENS in the time they occur. The subscription system informs users through asynchronous messages (e.g., via e-mail) on sets of changes concerning entities the users have selected. Examples of subscription include subscribing to changes of the "current population" attribute of one's home town, or of the "phone" attribute of a former colleague.

The system we propose ranks the information on changes in the sent message according to user preferences on what is important, and is able to adapt to the individual use-patterns of each client. This ranking is based on a user model, which is created through machine learning and a feedback-based user modeling methodology. The system we propose further adapts to user interest shifts, based on the information of the feedback.

Within this work, we further study the relation between size of the system's memory window — which takes into account the last k instances of user feedback to create the user model — and the performance of the system. We conclude that:

- The performance of the system is highly correlated to the average frequency of user interest shifts.
- There exists a near-optimal operation range for the window memory size parameter, which is computed based on the estimated interest shift frequency.

The experimental results show that the proposed approach can successfully adapt to usage patterns, delivering to each user change notifications ranked according to one's interest. Note that we use ENS as a concrete example for ease of exposition, but our techniques can be applied to any adaptation process, where there are data that are hybrid in nature, i.e., they can be represented by a vector *and* a string representation of content.

In the following sections, we present related work (Section II), we define the problem tackled herein (Section III) and elaborate on our proposed methodology (Section III-A). Then, we present the performed experiments (Section V), followed by related work (Section II). We then conclude and present future work (Section VI).

II. Related Work

User-modeling has been used for several years (e.g., [5]) and applied to a variety of settings, from information retrieval [6] to filtering tasks [7], [8] and recommendations [9]. In this work, the information retrieved is defined a priori and is always sent to the user without any filtering, because the user is subscribed to specific entities. However, what is considered unimportant based on estimation of importance can easily be discarded, if needed, providing a filtering mechanism.

Feedback-based approaches have been proposed in information retrieval since the early 1970's [10]. These feedback methodologies have relied both on the vector space model [11], [6] or Bayesian modeling [12] to represent user needs. Our method is a content-based method, as far as the content vs. collaborative [13] perspective of the system is concerned. The novelty of our work lies in the fact that we express information instances taking into account their dual nature: *type* and *content*. For us, the *type* represents the directly quantifiable (or simply binary) aspects of a piece of information, e.g., the number of words of a text, or whether a car is blue or not. The *type* of our instances, which are pieces of change information, is identified by vector features engineered for entity change subscription. The *content* represents the descriptive part of a piece of information, e.g., the name of a person, or the content of a web page. To represent what we define as content, we use an intermediate representation: the n-gram graph, which takes into account fuzzy matching of strings[1]. However, our methodology allows any kind of similarity metric to be applied between the content part of any two information instances. The function of similarity allows to add the *content* as a set of new dimensions to the feature space describing a change. Therefore, the representation of the user model is different in semantics from the representation used in most of the other approaches, because of the intermediate representation of the n-gram graphs.

Within this work we use incremental, explicit user feedback. Incremental feedback has been used in retrieval tasks [15] and in the context that we are applying it, which is actually a push-based information flow, it is the most appropriate approach. However, in our case there are no user queries to improve, like for example in [16], because the information delivered to the user is based on the changes of entities pre-selected by the user. There is also no relevant document set that we can use to expand our knowledge, in contrast to retrieval tasks. Therefore, we only take into account the user's history of feedback to corresponding instances. We take into account individual user needs and not "canonical" (i.e., group or community) user needs [5], [17].

Our adaptive subscription service aims to *rank* the information retrieved. Ranking has been in the user modeling community since the 90's [18] and has given rise to the recommender systems domain (see [13] for a survey). In this work, the ranking of information is treated as a *regression problem*, based on a mapping of qualitatively described user interest to real values (similar to the notion of utility [13]), so as to be generically applicable. However, we do not search for a function $PREF(u, v)$ that indicates the certainty of whether u should be ranked before that v, as, e.g., is described in [18]: the ranking in our case comes as an emergent result of the regression method. Our approach evaluation takes into account cases where the user considers instances equally important, because we estimate error in a way that accounts for *acceptable error* in the regression. A base difference that exists between our system and recommender systems is that we only need to rank pieces of information that are designated by the subscription of the user; we do not recommend *new* information instances, relative to user preferences as is the case in some recommender systems [19], nor are we interested in *popular* recommendations, as in [20].

The importance of term coexistence as a model for user preferences in content filtering, as well as the ability of the system to adapt to user shifts, has also been studied through an analogy to the immune system in Nootopia [21]. There, a network (or graph) representation of terms, as well as their coexistence in interesting content, form the representation of what we consider the *content* of user preferences. The terms, represented as weighted nodes of the network, are connected through a weighted edge, if they appear in a given window of words within a text. The weights on the edges are a function of the frequency and distance of co-occurence of the connected terms. In Nootopia however, the interesting — non-interesting distinction is performed through a self-organizing model, that rejects unimportant pieces of information, which does not allow several levels of interest. Furthermore, the system uses whole words as the basic unit of analysis, whereas we use sub-word items. The advantage of the sub-word items is that one can decide on similarity between different word forms, without the need of preprocessing like stemming, lemmatization, etc. The process of checking against the preference graph in Nootopia is performed through a network activation process, which increases complexity, when compared to our Normalized Value Similarity algorithm. Finally, we take into account other aspects of the preference, represented as the *type* portion of the change and the final estimation of the graded interest is estimated in the vector space and not the space of graphs.

In this work, we also study the effect of *"memory" window*, i.e., the number of remembered feedback instances, to the performance of the system, also in the presence of *user interest shifts*. Contrary to [15], in the presence of user interest shift, taking into account only a limited number of feedback instances does better than using all available feedback data, because an interest shift can heavily alter the estimator function's efficacy. Our study indicates that there seems to be a steady functional relation that exists between the user expected interest shift and the memory window, in order to achieve near-optimal results. A highly related study, even in a similar task, has been performed in [22]. However, we do not use classes of documents to relate the user to class-based interest, as done in [22], and we do not use a binary decision of correctness as feedback. Furthermore, our information instances, i.e., changes, have some dimensions concerning their *type* and the model of the content is not based on bag-of-words. Finally, we do not study *how* to detect an interest shift, but the relation between an estimated interest shift frequency and the "memory" window. The detection of the interest shift can use any method from time-series analysis or any probabilistic model (e.g., Hidden Markov Model), but is not the focus of this work.

[1] The n-gram graphs can actually represent any type of information where neighbourhood can be defined between the features of a represented instance. For more details, see [14].

In the next section we formulate the problem of adaptive subscription as a problem of interest estimation, based on user-feedback.

III. Problem Formulation

The subscription service is a supplement to the usability of the ENS, tackling the problem of keeping consumers of information concerning specific entities updated. The system disseminates information to consumers[2] about changes on entities to which the consumers have subscribed.

The sending of information can be either synchronous, based on the time a change takes place, or asynchronous. Furthermore, the update can be sent for every individual change or for a set of changes, e.g., periodically per day. In our case we use an asynchronous model for the update, that takes place periodically and contains all the information the user has subscribed to. It should be noted that a user subscribes to an *entity*, which in turns implicitly defines the set of changes that will reach the user.

It is clear that a subscription service for changes on entities is not very different from any other kind of subscription service. Nevertheless, the *attributes* of the information on changes partially define the adaptive approach we present within this work (see Section IV-A). As we elaborate later in this article, changes can have various degrees of importance, based on their *type*. For example, a "delete" change can be more important than an "insert" change (also see Section IV-A). Then, the *content* of a change is a completely different aspect from its *type*. We consider the content of the change to be the difference between the attributes of the original state of an entity and the attributes of the new state. We argue that this *content* can provide more information than the *type* of the change alone (also see Section V).

A. Subscription Feedback and Adaptive Content Ranking

The problem we tackle within this work is bringing the information on changes to consumers, in such a way that will best suit individual consumers' needs. The main scenario we face wants a user-consumer to have subscribed to a set of entities from the ENS, in order to get informed about changes in these entities. However, each user may be interested in different kinds of changes. This problem needs an adaptable system to support the modeling and application of user needs and preferences. The system we propose is aware of user feedback in order to improve the users' experience and optimize the flow of information to each individual user.

The problem can be formalized as follows:
Given

- a set of users, i.e., user models $\mathbb{U}$
- a set of change descriptions $\mathbb{D} = \{< T, C >, T \in \mathbb{T}, C \in \mathbb{C}\}$ of type T and content C, where $\mathbb{T}$ is the set of possible types and $\mathbb{C}$ the set of possible contents
- a set of feedback indications $\mathbb{F} = \{< U, D, I >, U \in \mathbb{U}, D \in \mathbb{D}, I \in \mathbb{I}\}$, where $\mathbb{I}$ a set of importance indicators, either categorical (nominal-scale) or real

we need the system to optimize the estimation function of importance for a new description of a change D_n, $f(U, D_n|\mathbb{F}) : \mathbb{U} \times \mathbb{D} \rightarrow \mathbb{I}$, for our criterion J (described below) that aims at minimizing the following absolute difference: $|f(U, D_n|\mathbb{F}) - I_0(U, D_n)|$, where $I_0(U, D_n) \in \mathbb{I}$ is the user assigned interest value.

In this work the $\mathbb{I}$ set will be considered to be the set of real numbers, i.e., $\mathbb{I} \equiv \mathbb{R}$. This way we can provide a partial ordering of descriptions $D_i \in \mathbb{D}$. Our criterion J is based on whether the ordering of the set of descriptions a user is entitled to get (due to a subscription) is the one *expected by the user*. We do not consider that there is a unique ordering that will suit the user's needs. Instead, we measure the performance of the system based on a devised measure, called Ratio of Acceptable Errors (RAE), defined in Section V.

IV. Proposed Approach: Adaptive Subscription-Based Updates

The architecture we have devised consists of the *Change Queue*, the *User Profile Database*, the *Subscription Information Broker* and the *Adaptive Information Control* (see Figure 2).

The *Change Queue* is a change log or repository of changes retaining information on the time stamp of a change. The *User Profile Database*, matches a user-consumer to her profile and holds such information as the time stamp of the last update the user has received. The database can be distributed over several servers and a login-based approach allows mapping each user-consumer to a corresponding profile and the set of entities she has subscribed to. On the other hand, the sustem is designed in a way that there is no need to transfer profile information across different modules of the system, thus seriously reducing privacy concerns for this information.

The *Subscription Information Broker* takes change information and stores them into the *Change Queue*. The broker is also the component that sends the information to the consumers. To do that, the broker requests from the *Adaptive Information Control* component the set of data that need to be sent to active users, on a per user basis. Furthermore, the *Subscription Information Broker* reports the consumer feedback to the *Adaptive Information Control* when the consumer uses the feedback mechanism, to allow for adaptation to consumer needs.

The *Adaptive Information Control* matches each consumer to a user profile, requests from the *Change Queue* the

[2] The terms *client*, *user* and *consumer* will be used interchangeably throughout the text.

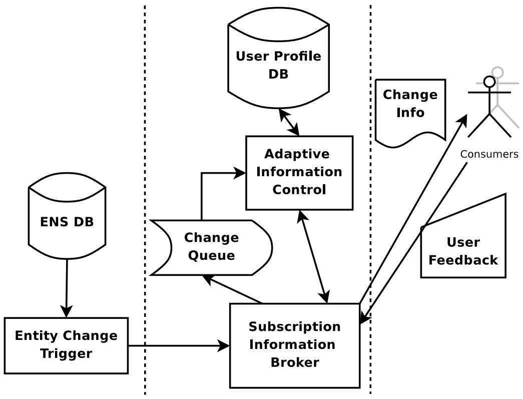


Figure 2. Schematic of the *Subscription Service* and its interactions.

changes that have occurred since the user's last update and ranks and groups the information that will be sent to the user. This information is then passed on to the *Subscription Information Broker* to be disseminated. The *Adaptive Information Control* also uses user feedback provided by the *Subscription Information Broker* to recompute and update the user profiles in the *User Profile Database*.

The asynchronous nature of the overall process, as well as the fact that only *Subscription Information Broker* interacts with external messages, allows for off-line and distributed processing, facilitating scalability. This is why no external access is granted to the subscription service database directly. Furthermore, the *Adaptive Information Control* module is a required medium between the feedback mechanism and the user model database, because of the required calculations for the update of the user model, especially as far as it concerns the content of a change (see Section IV-A2).

The adaptation of the aforementioned architecture to user needs relies on two main aspects of the *Adaptive Information Control*: the first is how changes' descriptions are represented to form a user model and the second is how the system adapts to user feedback.

A. Modeling Change and Information Importance

We define the description D of a change in a dual way, concerning the *type* and the *content* of the change. For each of these aspects we use a different kind of representation to address their individuality.

1) Type of a Change: The *type* T of a description can be viewed as a point in a multi-dimensional space, where the dimensions are:

- One dimension for each alternative of *entity change*: *deletion*, *splitting* of an an entity into two, *merging* of two entities in one, and *entity update*. A deletion change signals the removal of an entity from the ENS. A splitting change indicates the splitting of a single entity into two individual entities, when an entity is found to have been erroneously created with data from two different entities. A merging change describes the merging of two entities into a new one, when the original entities actually referred to the same real entity. An update indicates any change in the individual attributes of an entity. The value for each dimension can be either 1.0, indicating the corresponding type of change, or 0.0 otherwise.
  We have decided to use a different dimension for every change alternative, because enumerating the alternatives over a single real axis would imply a partial ordering between alternatives, which does not stand. An example case of a merging change will have the value 1.0 at the merging dimension and all the other alternatives' dimensions, i.e., deletion, splitting and update, set to zero.
- One dimension for each alternative of *attribute change*, which is the result of an *entity update* change: *deletion*, *insertion*, *update*. A deletion of an attribute means removing the attribute name-value pair as a whole from an entity. An insertion indicates the insertion of a new attribute name-value pair into an entity. An update is a change of a given attribute, using a new value. The update is further elaborated using the following dimension of *normality*.
- The normality of a type of change should indicate, in a quantitative manner, whether a given update appears to be expected. This normality can be judged by such processes as type checking for new values, or by whether a value holds similar qualities to other values for the same attribute. For example a grammar model for a given attribute can indicate whether the new value is normal or not. We use normality as a real value, normalized between 0.0 and 1.0, where 0.0 indicates maximum abnormality and 1.0 perfect normality.

We can now represent a type T of change in a vector space, but the content C of the change is a completely different challenge. In an ENS system, the state of an entity can be described as a set of attribute name-value pairs. Thus, C is determined as the difference between the two states.

Ideally, we want to be able to identify such preferences as "I am interested in changes that have to do with my entities' name or telephone number", or "I am interested about when the *inProduction* field of proteins has a value of *true*". To identify such preferences one must act on the string level and create a model for attribute names and attribute values that are of interest for the consumer. The string representation of the content is the attribute-value pair that was changed, in its changed version if such a pair is applicable to the change. Otherwise, the representation is an empty string.

2) Content of a Change: To model interesting attribute name-value pairs, we use the paradigm of character n-gram graphs [23], which can take into account substring matches and offer a set of operators that allow for an updatable model [14].

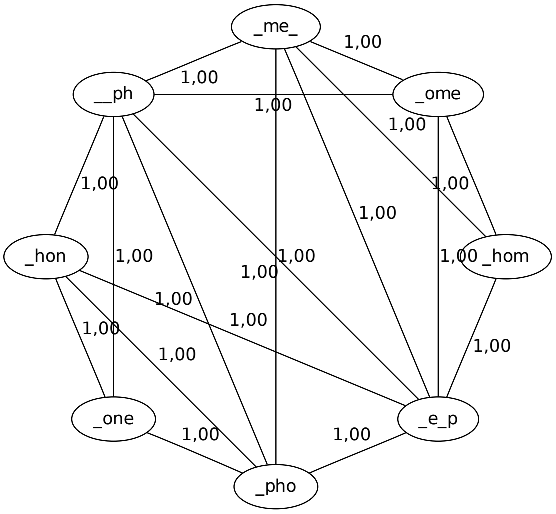
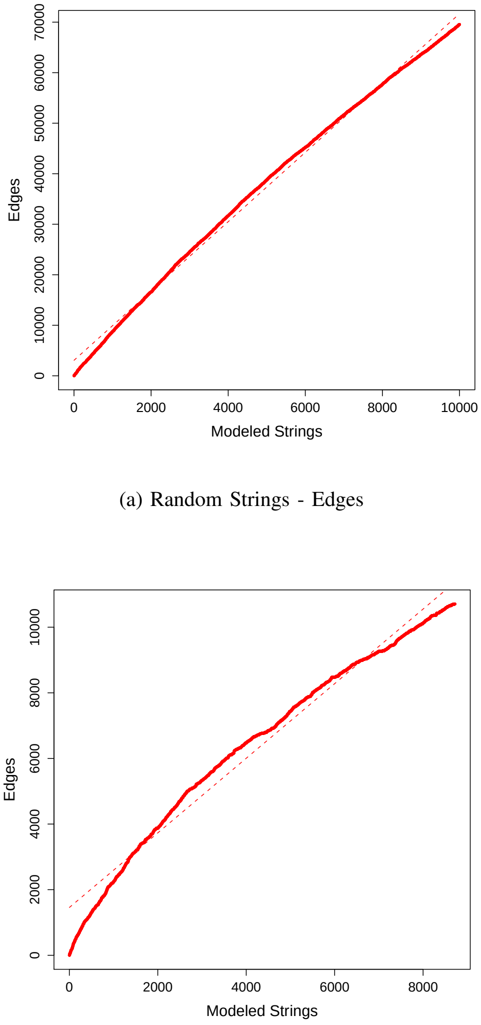


Figure 3. Sample N-gram Graph Representing the String “home_phone”

A *character* n-gram S^n contained in a text T can be any substring of length n of the original text. The *n-gram graph* is a graph $G = \{V^G, E^G, L, W\}$, where V^G is the set of vertexes, E^G is the set of edges, L is a function assigning a label to each vertex and edge, and W is a function assigning a weight to every edge. N-grams label the vertexes $v^G \in V^G$ of the graph. The (directed) edges are labeled by the concatenation of the labels of the vertexes they connect in the direction of the connection.

The graph is simply constructed by a running window over a given string, that analyzes the string into overlapping character n-grams and records information about which n-grams are neighbours within the window. The edges $e^G \in E^G$ (the superscript G will be omitted where easily assumed) connecting the n-grams indicate proximity of these n-grams in the text within a given window D_{win} of the original text [23]. The edges are weighted by measuring the number of co-occurrences of the vertexes’ n-grams within D_{win}.

A graph sample, generated for 3-grams with a maximum neighbouring distance of 3 can be seen in Figure 3. In Figure 4a we also depict the growth of graph size, in terms of edge count, based on the number of represented *random* strings, which shows that the graph’s edge count grows linearly to the number of modeled string instances. For *normal* text[3] the growth is more logarithmic than linear, as shown in Figure 4b. The logarithmic behaviour is more apparent when the vertices, i.e., possible n-grams, are used to indicate the growth of the n-gram graph (see Figures 5a, 5b).



(a) Random Strings - Edges

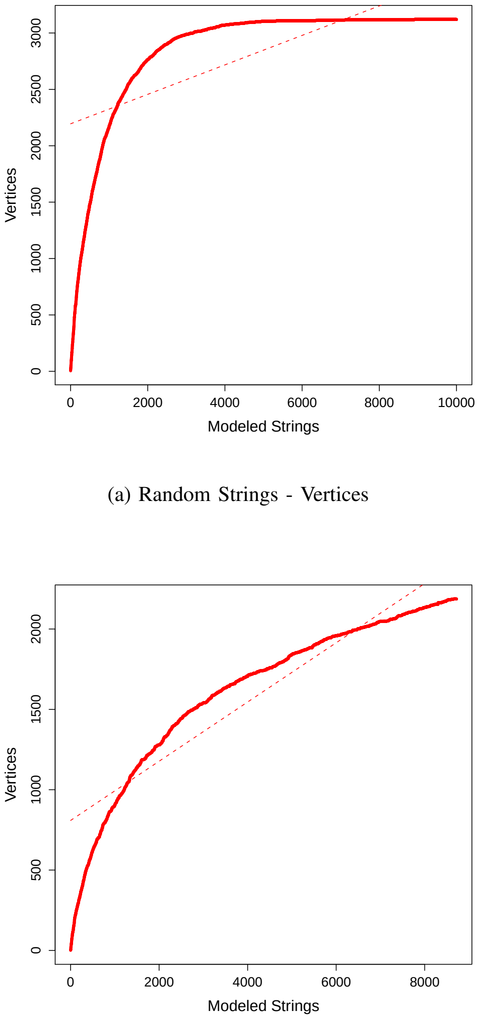
(b) Actual Strings - Edges

Figure 4. Edges of N-gram Graph Over the Number of Modeled Strings

We use the n-gram graphs within this work due to several of their traits like language neutrality and the fact that, when used for matching between strings, they offer a graded normalized indication of similarity. We also use the updatability of n-gram graphs, when applied as language models [14]. In other words, if we judge a set of attribute names-values as indicative of importance, we can create an n-gram graph that models the whole set and, thus, avoid keeping all the attribute names-values for matching. Furthermore, the model offers fuzzy matching and substring matching, which helps in open domains of attribute names and values, as is the case of an ENS.

To model the content C of changes a user is interested in, we create for each training instance C_i, given by the feedback process, a corresponding n-gram graph G_{C_i}. The

[3] The text used was the first book of “The Book of the Aeneis”, by Virgil, found online at http://www.ilt.columbia.edu/publications/Projects/digitexts/vergil/aeneid/book01.html.



(a) Random Strings - Vertices

(b) Actual Strings - Vertices

Figure 5. Vertices of N-gram Graph Over the Number of Modeled Strings

graph is based on the string representation of the change.

The model graph construction process for each set of changes (e.g., of the uninteresting/interesting/critical classes of changes) comprises the initialization of a corresponding graph with the first string representation of content, and the subsequent update of this initial graph with the graphs of the other content instances in the class.

Specifically, given two graphs, G_1 and G_2, the first representing the training set of changes and the second a new instance, we create a single graph that represents the updated model graph G_1 with the graph of new evidence G_2:

$$update(G_1, G_2) \equiv G^u = (E^u, V^u, L, W^u) \qquad (1)$$

such that $E^u = E_1^G \cup E_2^G$, where E_1^G, E_2^G are the edge sets of G_1, G_2, respectively. In our implementation two edges are equal $e_1 = e_2$ when they have the same label, i.e., $L(e_1) = L(e_2)$.

The weights of the resulting graph's edges are calculated as follows: $W^i(e) = W^1(e) + (W^2(e) - W^1(e)) \times l$. The factor $l \in [0, 1]$ is called the learning factor: the higher the value of learning factor, the higher the impact of the second graph to the first graph. More precisely, a value of $l = 0$ indicates that the new graph will completely ignore the second graph. A value of $l = 1$ indicates that the weights of the edges of the first graph will be assigned to the weights of the resulting graph. As we need the model of a class to hold the average weights of all the individual graphs contributing to this model, the i-th graph that updates the class graph (model) uses a learning factor of $l = (1 - \frac{i-1}{i}), i > 1$. This creates a class model that acts as a representative graph for the class content instances.

As already indicated, we need to create more than one model graphs: one per feedback alternative of the user. Within this work we give the user three alternatives for feedback: unimportant information, important information and critical information (in order of increasing significance). Therefore, we create three corresponding model graphs. These graphs represent their corresponding content instances in the user model.

In our system the set of $\mathbb{I}$, which stands for importance, is the set of real numbers $\mathbb{R}$. We set three qualitatively mapped thresholds of importance: -1.0, which indicates unnecessary information, 1.0, which indicates useful information and 2.0, which indicates critical information. Of course, the set of importance alternatives could have as many elements as desired, keeping in mind that, if $\mathbb{I} \equiv \mathbb{R}$, using higher values for higher importance will probably provide better distinction. Given this kind of mapping, we need to be able to judge the importance $i \in \mathbb{I}$ of a new instance of changes, for a particular user.

To further clarify these definitions, we provide examples of changes that a user of the system could evaluate as:

- Critical: name deletion or garbaging names, deletion of default attributes of a type, e.g., deletion of the *longitude* and *latitude* of a location entity.
- Important: change of the longitude and latitude of a location entity with a new, normal value.
- Unimportant feedback: change in a field with a value that references some other ENS entity.

B. *Ranking Using User Feedback*

Having reported on the representation of changes, we can now describe the methodology for assigning importance values to entity changes, based on a user model.

The parts of the process that need to be described are

- What algorithm is used to *learn* which changes are important for a given user?

- What are the descriptive features that can be used to model a change?

The algorithm we use to learn the user model is actually that of Support Vector Machine Regression (SVR). SVMs have already been successfully used in a variety of applications and settings [24]. Within this work we use the LibSVM library [25] and especially its ϵ-SVR implementation of the algorithm found in [26].

The basic idea behind the ϵ-SVR is that, given a set of training data $\{(x_1, y_1), ..., (x_l, y_l)\} \in X \times R$,where X is the space of the input patterns we need to “find a function $f(x)$ that has at most” ϵ “deviation from the actually obtained targets y_i for all the training data, and at the same time is as flat as possible. In other words, we do not care about errors as long as they are less than” ϵ, ” but will not accept any deviation larger than this” [27]. In our case, of change types T, $X \equiv \mathbb{R}^d$, which is the vector space we defined for the representation of types $T \in \mathbb{T}$, and $y_i \in \mathbb{I}$.

For the content C of changes, on the other hand, we first calculate the *size-normalized value similarity*[14] between the n-gram graph G_C of a judged C, with respect to each of the n-gram graphs of the user model $G_U^{important}, G_U^{unimportant}, G_U^{critical}$. This similarity value, which lies between 0.0 and 1.0, indicates what part of the graph of C can be found in the corresponding graphs of the model of the user. This set of similarities $\mathbb{S} = S^{unimportant}, S^{important}, S^{critical}$ is the second constituent of the representation of a description $D =< T, C >$ of a change with respect to a user model. To use this set of similarities, we integrate them within the vectors of the type as new dimensions-features. Therefore, when n-gram graphs are used, the overall importance of a change is estimated based on the *combined vector* for type *and* content in an extended input vector space X'. For an illustration of the overall process of using the graph models into the feature vector, also see Figure 6.

While in the original version of the system we created a single graph for both attribute names and attribute value modeling, experiments (see section V) indicated that these two models needed to be separated. Therefore in a second version of the adaptivity process, each level of importance uses two graphs to represent attribute names and attribute values.

C. Shift-Aware Adaptive Subscription System (SAAS)

The problem we have described so far does not take into account changes of user interest over time. However, in real world scenarios interest shifts occur. This means that a user does not have a single behavior over time, but that at random times she changes the target (i.e., desired) ranking of information.

To tackle the problem of shifting interest we decided to study the case where the system only remembers the feedback instances of the last k iterations. This means that the system “forgets” everything before that. This introduces the notion of system *memory* to our approach. The system memory for iteration i is therefore defined as the set of feedback instances provided by the user within the time interval $[i - k, i - 1]$. This system memory describes the training set of the learning algorithm. The implementation simplicity of the system memory approach makes it easily integrated in a system like the ENS.

The full system that also takes into account interest shifts has been named *SAAS*.

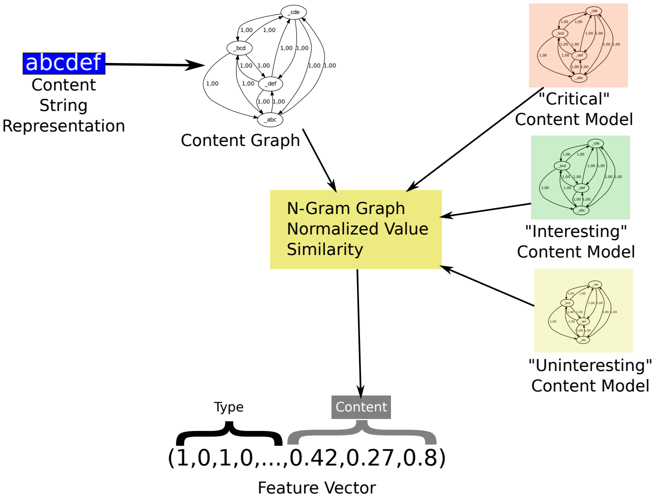


Figure 6. The Process of Content Features’ Extraction

V. Experiments and Evaluation

In order to evaluate our methodology, and given the fact that OKKAM is not yet at a production state, which means that users have been limited in number, we had to generate a synthetic set of entity changes, much like other works on adaptive systems (e.g., [28], [29]). In the following subsections we describe the data, the process and methodology of evaluation, as well as the results of our experiments.

A. Data

The information concerning user behavior is twofold in our case. We try to replicate the behavior of the users of the ENS who change entity data. Then, we create profiles for the behavior of subscribers using the adaptive subscription service. Correspondingly the data we generate are of the following two aspects:

- ENS User Behavior Data.
- Entity Subscribers’ Feedback Data.

1) ENS User Behavior Data: As behavior of the users that change the entities’ data, we generate a number of changes’ descriptions D — 10000 instances split into sets of 1000 instances to provide for 10-fold validation. To generate this kind of dataset, we randomly create changes based on a

User type (Prob.)	*Change type*	*Probability*
Benevolent (0.95)	Attribute change (normal)	0.60
	Attribute insertion	0.30
	Attribute deletion	0.10
Sys.admin.(0.03)	Entity merge	0.45
	Entity split	0.45
	Entity deletion	0.10
Malevolent (0.02)	Attribute change (abnormal)	0.70
	Attribute deletion	0.30

Table I
ENS USER BEHAVIOR: PROBABILITY DISTRIBUTION OF CHANGE EMISSION

selection from the following user behaviors: benevolent user changes, system administration changes and malevolent user changes. The probabilities of emission per user profile and change type are elaborated on in Table I.

2) Entity Subscribers' Feedback Data: The second part of the evaluation dataset consists of subscriber feedback. We consider a few representative cases of subscribers to minimize the evaluation overhead, while providing useful insight on the adaptivity of the system:

- a subscriber who is interested in the deletion of attributes, to make sure no information concerning the entities he has subscribed to are missing. This scenario is expected to mostly use the type of the change description as a discriminating factor of importance.
- a subscriber mostly interested in the changes of names entities. This subscriber scenario is expected to mostly use the content of a change description as a discriminating factor of importance.
- a subscriber who is interested in whether any entity he has subscribed to has its “isDeceased” status changed. This scenario aims to represent the difficulty of users interested in specific attribute *values*.
- a subscriber who has the role of a validator of data in OKKAM, who wants to be aware of abnormal changes in default attributes, such as deletion or change with an abnormal value. This subscriber scenario is expected to use both the type and the content of a change description as a discriminating factor of importance.

A more detailed description of what each user finds interesting and critical can be found in Table II. All changes not noted within a profile are considered uninteresting for the profile. The profiles have been chosen so that they are require different kinds of information to be determined, concerning either the type or the content of the change.

Before the actual evaluation, we use the learning methodology and produce corresponding results, also supplying feedback for every step. This process is reiterated for every subscriber scenario for the whole set of change data. The information supposedly sent to the user is annotated with the iteration number, simulating the time-stamp of the change. For our case, iteration i is considered complete when the system gets the feedback of that i-th step from the user.

Subscriber	*Importance*	*Description*
Type-based	Critical	Attribute deletion.
	Interesting	Entity deletion.
Attribute name-based	Critical	Any change concerning an attribute that contains the string “name”.
	Interesting	(None)
Attribute name-value pair-based	Critical	Attribute change or insertion on “isDeceased” attribute, with a new value of “true”.
	Interesting	Attribute change or insertion on “isDeceased” attribute, with a new value of “false”.
Complex	Critical	Default attribute (some attributes in the ENS are considered default — e.g., the name of a person entity — while all the others non-default) update or insertion with an abnormal value.
	Interesting	Default attribute deletion or normal update.

Table II
PROFILE DESCRIPTIONS. *Note:* ALL CHANGES NOT NOTED WITHIN A PROFILE ARE CONSIDERED UNINTERESTING FOR THE PROFILE.

B. Evaluation Methodology

To evaluate the system learning effectiveness, as well as whether the addition of content-related features is useful, we experiment on different aspects of the system response. We determine how quickly the system learns, by “emitting” changes to the supposed user — in groups of ten — and measuring how well the systems adapts to the feedback. We consider that the user feeds back the system after every new emission, by indicating the importance of all the items in the last group. The performance of the system for every emission is, at this point, based on the mean absolute error of the importance estimation. To judge the learning curve of the system, we study the magnitude of the mean absolute error as related to the current number of emissions. We also define the Rate of Acceptable Errors measure as an alternative measure of performance.

1) Rate of Acceptable Errors (RAE): We define the system performance for a given change emission to be the number of times a ranking error has exceeded 0.5. Given our $\mathbb{I}$ values, errors beyond this 0.5 margin *may* cause an error. Errors below this margin cannot cause an error by themselves. So the performance is *the percentage of the importance estimation in a given set that have their absolute error below* 0.5. Thus, a value of 1.0 in performance indicates a ranking that is ideal, while a value of 0.0 indicates a ranking that will have several errors. We call this measure *Ratio of Acceptable Errors (RAE)*. The formula, for a given set $\mathbb{D}_0$

Subscriber	*Graphs*	*Correlation (p-value)*
Type-based	✓	**-0.3398559** ($< 10^{-2}$)
		-0.2993715 ($< 10^{-2}$)
Attribute name-based	✓	**-0.3734062** ($< 10^{-2}$)
		-0.03564642 (0.2601)
Attribute name-value pair-based	✓	**-0.08581718** ($< 10^{-2}$)
		-0.02968072 (0.3484)
Complex	✓	**-0.5989662** ($< 10^{-2}$)
		-0.03393356 (0.2837)

Table III

CORRELATION BETWEEN EMISSION-ITERATION NUMBER AND MEAN ABSOLUTE ERROR PER SUBSCRIBER PROFILE AND METHOD. *Note:* HIGH STATISTICAL CONFIDENCE IS INDICATED USING **BOLD WRITING**.

of descriptions, their corresponding sequence of importance estimations $\tilde{\mathbb{I}}_0 = \{y_l | y_l = f(D_l, U|\mathbb{F}), D_l \in \mathbb{D}_0\}$ and actual importance values $\mathbb{I}_0$, is:

$$RAE(\tilde{\mathbb{I}}, \mathbb{I}_0) = 1.0 - \frac{\sum_{i\in(1,\ldots,|\mathbb{I}|)} \lfloor min(|\tilde{\mathbb{I}}(i) - \mathbb{I}(i)|, 0.5) + 0.5 \rfloor}{|\mathbb{I}|} \tag{2}$$

where $\tilde{\mathbb{I}}(i)$, $\mathbb{I}(i)$ is the i-th element of the corresponding sequence, $\lfloor x \rfloor$ is the floor operator, $|\mathbb{X}|$, gives the number of elements of a sequence $\mathbb{X}$, $|x|$ is the absolute value of a number x and $min(x, y)$ is the minimum function.

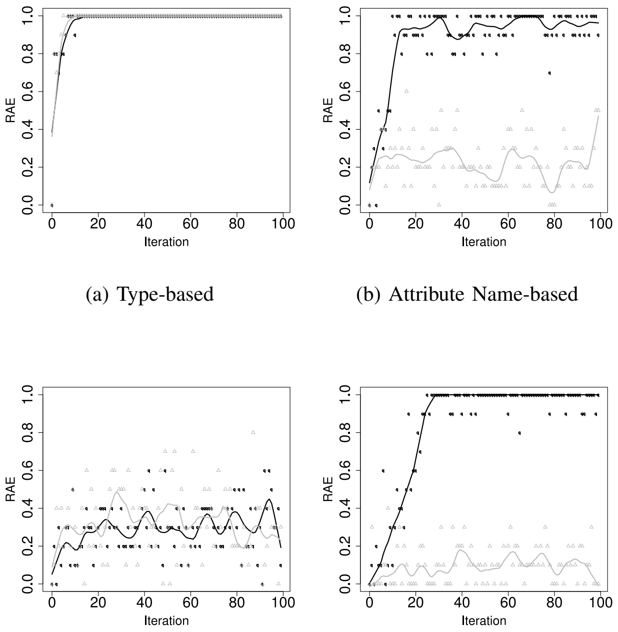
To determine whether the content aspect of the system is a valuable resource, we trying two alternatives. One uses the graph similarities as a feature, while the other does not. When we use content, we expect that the system will perform better for subscriber profiles that indeed use content criteria. For other profiles, we expect no loss in performance.

C. Learning Different User Profiles

Table III indicates the Pearson correlation [30] between the number of emission, i.e., the training iteration, and the mean absolute error. A negative correlation indicates that, after performing more training the error is diminished. We see that in all cases, when using graphs, the system learns, with statistical support. In all the cases where we expected that the content should be used, indeed the system cannot learn when the content-sensitive methodology is not applied.

RAE as a function of time, as illustrated in Figure 7, indicates the learning curve of the system. The points in the graphs indicate the RAE for a given iteration. We note that there can be several RAE for a given iteration, since we have applied 10-fold validation of the results. The lines represent an approximation (LOWESS smoothing [31]) of RAE over iterations. Dark lines refer to the RAE when using graphs, while gray lines represent the performance without them.

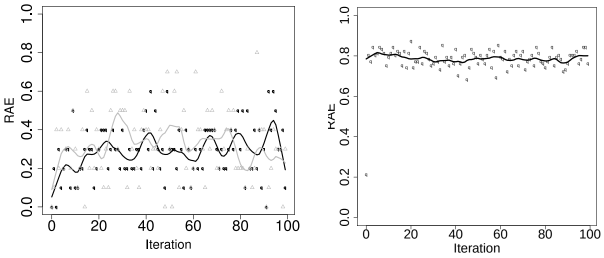
It appears that, when the content methodology is used, all profiles are feasible to learn. At this original version, the most difficult case was the attribute name-value-based one (see Figure 7d), because we used to represent the name-value pair as a single string and, therefore, generate features of similarity for them in common, inducing noise in the n-gram graph pattern matching. In Figure 8 we see the evolution from the common graph to the separation of graphs for the attribute name and the attribute value, leading to new corresponding features in the vector space. It is obvious that the original problem has been adequately faced and the learning of the originally difficult case, now takes minimal iterations to occur.



(a) Type-based (b) Attribute Name-based

(c) Attribute Name-Value-based (d) Complex

Figure 7. The learning curve of the first version of the system for different profiles.



(a) Common Graph (b) Separate Graphs

Figure 8. The learning curve in the original and the enhanced version of the system.

In general, very few iterations (< 10) are enough for the profile to be learned. Even the complex profile is learned in about 20 iterations. This shows that the presented methodology is very promising in learning subscriber profiles on changes, but a new version should take into account the flaws of the first. Furthermore, in the original version there was no support for changing interest over time.

To study also the more realistic case when a user changes her interests, we created a *shifting-interest user* profile in our data generation process. This user changes her interests either in a periodic or a random manner. Upon the iteration of the interest shift the user is replaced by one of the basic profiles (see Table II) in a round robin manner. We will call this user *changing user*. In the second version of the system, we also split the representation of interest for attribute names and attribute values, to check whether all profiles could be effectively learned now.

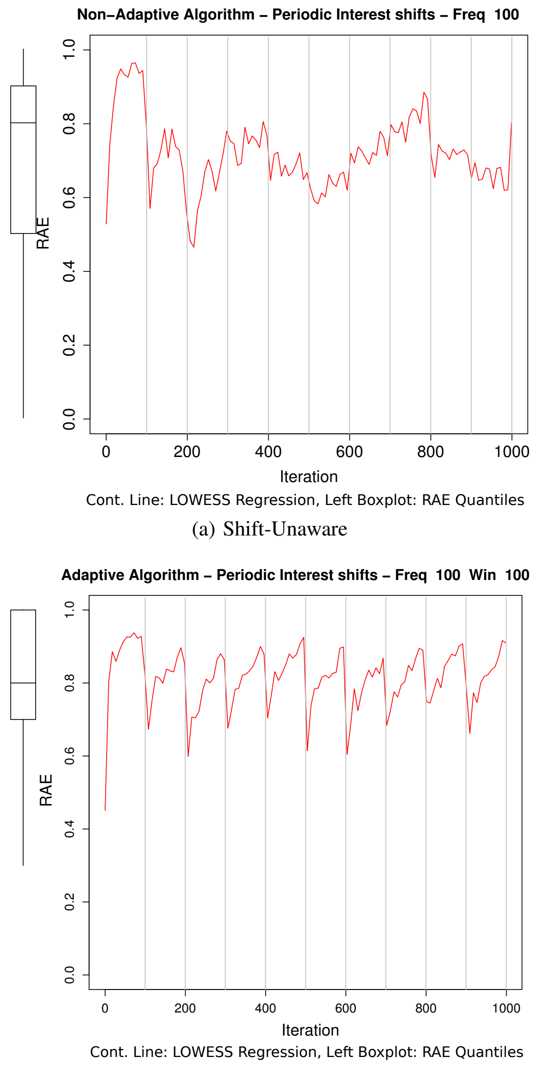
D. Learning Changing Users' Preferences — Periodic Change

In the case of the periodic change we generated the feedback for the changing user, who every 100 iterations shifted her interest profile. Given our previous experiments, we had seen that the system takes about 20 iterations to learn a profile and, thus, the selection of 100 iterations gave enough time for the system to show its behavior.

In order to benchmark our shift-aware version of the system we first used the non-adaptive algorithm, where the system takes into account *all* the feedback from the user so far. This approach, presented in [1], can be effective for non-changing interests, given that the system, once reaching top performance, will stop taking new instances as data. It is important to note here that the feedback of the user can give a direct evaluation of each iteration, therefore providing information about whether the system has reached high performance.

In Figure 9a, we present the behavior of the shift-unaware version of the system. The boxplot on the left indicates the quantiles of the distribution of the values. More precisely, for a population of N ordered sample values, it indicates the values at positions 1 (min. value),$\frac{N}{4}$,$\frac{2N}{4}$ (median), $\frac{3N}{4}$, N (max. value). The vertical gray lines indicate the interest shift. We repeat that, after every interest shift, the user follows another profile. The first thing that is apparent in the figure is that, even though the algorithm starts really well, after the first interest shift it never manages to reach the original levels the performance.

To compare the shift-unaware version to the final SAAS, we performed experiments with windows of various sizes, but first we tested the system with a window equal to the shift frequency[4]. The result is illustrated in Figure 9b. It is obvious that the performance of the SAAS is much more robust than that of the shift-unaware version.



(a) Shift-Unaware

(b) Shift-Aware

Figure 9. RAE over Time (LOWESS regression). The boxplots on the left of each diagram indicate the distribution of values.

In order to determine how sensitive the system is to the selection of the window we conducted a whole set of experiments for a range of different windows. The results, together with the performance of the shift-unaware version, are shown in Table IV. As can be seen, even the extreme cases where the window is really small do better *on average* from the shift-unaware version. This has to do with the fact that too much noise is induced in the training set if the memory holds all the history of feedback. Therefore, the shift-unaware version can be both ineffective and expensive in terms of storage when compared to SAAS.

Another important conclusion however from the table was that, even though the system is generally robust, the performance *does vary* according to the window size. The next step was to study what is the correlation of the window size to the performance. In fact, we studied the correlation between

[4] Within the text we will use the term frequency to actually describe the *period* of the shift. So 1 shift every 100 iterations will be described as having a frequency of 100.

		Quantiles (Measurements in %)				
Shift-	Window	1st	Median	Mean	3rd	Max
Unaware	N/A	50.0	80.0	70.1	90.0	100.0
Aware	5	60.0	80.0	73.0	90.0	100.0
Aware	10	60.0	80.0	74.9	90.0	100.0
Aware	20	70.0	80.0	78.1	90.0	100.0
Aware	40	70.0	80.0	79.4	100.0	100.0
Aware	80	70.0	90.0	80.3	100.0	100.0
Aware	90	70.0	80.0	80.0	100.0	100.0
Aware	100	70.0	80.0	79.4	100.0	100.0
Aware	110	70.0	80.0	78.8	100.0	100.0
Aware	120	70.0	80.0	78.3	100.0	100.0
Aware	150	60.0	80.0	76.3	90.0	100.0

Table IV
RAE PERFORMANCE QUANTILES OF SHIFT-AWARE VS SHIFT-UNAWARE VERSIONS. PERIODIC INTEREST SHIFT — EVERY 100 ITERATIONS.

	Quantiles (Measurements in %)				
Window	1st	Median	Mean	3rd	Max
10	60.0	80.0	76.2	90.0	100.0
20	70.0	80.0	78.3	100.0	100.0
40	70.0	80.0	79.6	100.0	100.0
60	70.0	80.0	78.9	100.0	100.0
80	70.0	80.0	80.3	100.0	100.0
100	70.0	90.0	79.9	100.0	100.0
120	70.0	80.0	77.5	100.0	100.0
140	60.0	80.0	77.2	100.0	100.0
160	60.0	80.0	74.9	90.0	100.0

Table V
RAE PERFORMANCE QUANTILES FOR RANDOM INTEREST SHIFT — EVERY 50-150 ITERATIONS UNIFORMLY

the ratio of the window to the interest shift frequency.

E. Learning Changing Users' Preferences — Random Change

Given the fact that we wanted to study what is the effect of having a small or large window as related to the size of the change frequency, we decided to make the experiment even more demanding: the user would randomly make the interest change. To create this random change we used a random sampler[5] that, given a mean of the change frequency M, would cause an interest shift in the range of $[\frac{M}{2}, \frac{3 \times M}{2}]$.

First, we studied the performance of the SAAS for the *random* change with $M = 100$. The results can be seen in Table V.

To further delve between the relation of window size, M and RAE performance we repeated the experiments with $M = \{50, 75, 100, 125, 150\}$ iterations. The results can be found in Figure 10, where we depict the average level of performance (i.e., average RAE) for different $\frac{Window\ Size}{M}$ ratios, given various window sizes. From the diagram, we noted two things:

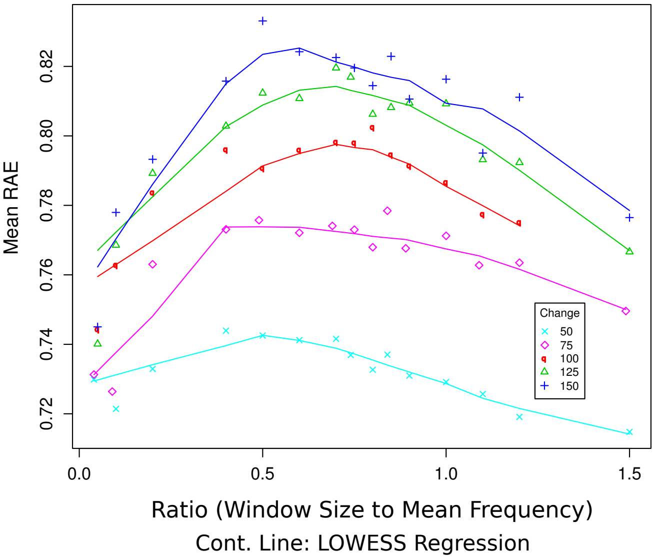


Figure 10. **RAE** Over **Memory Window to Mean Frequency Change Ratio** for SAAS.

- There seems to be a correlation between the average RAE performance of the system and the average frequency of the shift. This correlation was also proven statistically by a Pearson correlation test between the performance of the system and the average frequency of the shift. The test showed that there is a very strong linear correlation (approximately 0.8) with high statistical confidence (p-value was much lower than 10^{-2}. This simply verifies that a users that do not change frequently their interest will be easier to model.
- There is a range for the ratio of window size to expected change frequency, which appears to hold high performance, regardless of the frequency itself. To better illustrate and estimate this range we provide Figure 11, where we present the overall — i.e., over all window sizes — relation between RAE and $\frac{Window\ Size}{M}$ ratio. Given that the estimated user interest shift frequency is F, then it appears that the range between $0.4 \times F$ and $0.8 \times F$ offers near-optimal results. This indicates that it is enough to use the expected value of the frequency to get near-optimal results. Given the fact that less memory means less space, one could select to use $0.4 - 0.5$ of the estimated user interest shift frequency to have low spatial cost with near-optimal performance.

VI. CONCLUSION AND FUTURE WORK

We propose an adaptive subscription service architecture, concerning the update of clients of an ENS with information on entity changes. The subscription service, called *SAAS*, uses information from user feedback to model user needs, taking into account *both the type and the content* of changes. The system appears to learn even complex user preferences

[5] Overall the sampler is adequately unbiased. For a study of the bias of the sampler, please see Appendix A.

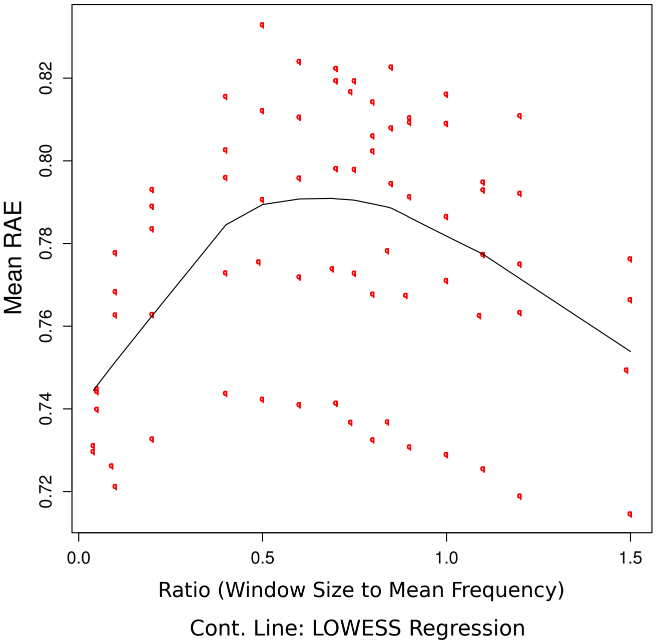


Figure 11. **Overall RAE** Over **Memory Window to Mean Frequency Change Ratio** for SAAS.

in a low number of feedback iterations, also coping with user interest shifts. Our system performs well, even with randomly changing user interest. It should be noted that in our experiments we used a radical shift of interest every time, which made the problem even more demanding than an average real case scenario. Nevertheless, the average performance is high and the system appears robust and able to cope with all different user profiles.

We studied the relation between the memory window size parameter of the user modeling process to the average performance of the system. There we detected a near-optimal range for the parameter, which can be calculated as a simple function of the estimated average change frequency of the user interest. We showed that the mean performance of the system is highly correlated to the interest shift frequency, but the system remains robust enough even for frequent interest shift changes.

In the future, we plan to study on information theoretic grounds whether one can determine a priori the performance of adaptive systems, which use user feedback history, based on a few easily extracted parameters of the system. For unchanging user behavior, we may also need to keep the memory window constant, i.e., not dependent on the current iteration, until a drop in performance indicating user interest shift occurs. This means that need to integrate the ability of detection of user interest shift into the system to improve on our performance. This interest shift detection should probably be based on the grounds of time-series analysis of the feedback data. It will also be important to compare our methods with an alternative that should use active learning to augment the training set with useful additions. Finally, it is important to use the content and type discrimination in other applications of the user modeling domain and determine whether there is a generic gain to using this approach.

Acknowledgment

This work is partially supported by the by the FP7 EU Large-scale Integrating Project OKKAM “Enabling a Web of Entities” (contract no. ICT-215032). See http://www.okkam.org.

References

[1] G. Giannakopoulos and T. Palpanas, “Adaptivity in entity subscription services,” in *Proceedings of ADAPTIVE2009*, Athens, Greece, 2009.

[2] P. Bouquet, H. Stoermer, and B. Bazzanella, “An entity name system (ENS) for the semantic web,” in *ESWC*, 2008, pp. 258–272.

[3] T. Palpanas, J. A. Chaudhry, P. Andritsos, and Y. Velegrakis, “Entity data management in OKKAM,” in *DEXA Workshops*, 2008, pp. 729–733.

[4] B. Bazzanella, J. A. Chaudhry, T. Palpanas, and H. Stoermer, “Towards a general entity representation model,” in *SWAP*, 2008.

[5] E. Rich, “Building and exploiting user models,” in *Proceedings of the 6th international joint conference on Artificial intelligence-Volume 2*, 1979, p. 720722.

[6] P. Langley, “Machine learning for adaptive user interfaces,” *Lecture notes in computer science*, pp. 53–62, 1997.

[7] M. Shepherd, C. Watters, and A. Marath, “Adaptive user modeling for filtering electronic news,” in *System Sciences, 2002. HICSS. Proc. of the 35th Annual Hawaii Int. Conf. on*, 2002, pp. 1180–1188.

[8] X. Zhou and T. Huang, “Relevance feedback in image retrieval: A comprehensive review,” *Multimedia systems*, vol. 8, no. 6, pp. 536–544, 2003.

[9] D. Bonnefoy, M. Bouzid, N. Lhuillier, and K. Mercer, “More like this or Not for me: Delivering personalised recommendations in multi-user environments,” in *User Modeling 2007*, 2007, p. 8796.

[10] J. Rocchio *et al.*, “Relevance feedback in information retrieval,” *The SMART retrieval system: experiments in automatic document processing*, pp. 313–323, 1971.

[11] C. Buckley, G. Salton, and J. Allan, “The effect of adding relevance information in a relevance feedback environment,” in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Dublin, Ireland: Springer-Verlag New York, Inc., 1994, pp. 292–300. [Online]. Available: http://portal.acm.org/citation.cfm?id=188490.188586&type=series

[12] P. Zigoris and Y. Zhang, “Bayesian adaptive user profiling with explicit & implicit feedback,” in *Proc. of the 15th ACM Int. Conf. on Information and knowledge management*. ACM New York, NY, USA, 2006, pp. 397–404.

[13] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[14] G. Giannakopoulos, “Automatic summarization from multiple documents,” Ph.D. dissertation, Department of Information and Communication Systems Engineering, University of the Aegean, Samos, Greece, http://www.iit.demokritos.gr/˜ggianna/thesis.pdf, April 2009.

[15] J. Allan, “Incremental relevance feedback for information filtering,” in *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*. Zurich, Switzerland: ACM, 1996, pp. 270–278. [Online]. Available: http://portal.acm.org/citation.cfm?id=243274&dl=

[16] A. Leuski and J. Allan, “Interactive information retrieval using clustering and spatial proximity,” *User Modeling and User-Adapted Interaction*, vol. 14, no. 2, pp. 259–288, Jun. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:USER.0000028978.09823.47

[17] M. McTear, “User modelling for adaptive computer systems: a survey of recent developments,” *Artificial intelligence review*, vol. 7, no. 3, pp. 157–184, 1993.

[18] W. Cohen, R. Schapire, and Y. Singer, “Learning to order things,” *J Artif Intell Res*, vol. 10, pp. 243–270, 1999.

[19] K. Onuma, H. Tong, and C. Faloutsos, “TANGENT: a novel,’Surprise me’, recommendation algorithm,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM New York, NY, USA, 2009, pp. 657–666.

[20] M. Vojnovic, J. Cruise, D. Gunawardena, and P. Marbach, “Ranking and Suggesting Popular Items,” *Online im Internet: http://research. microsoft. com/˜ milanv/popularity. pdf [10.06. 2008]*, 2007.

[21] N. Nanas and A. de Roeck, “Autopoiesis, the immune system, and adaptive information filtering,” *Natural Computing*, vol. 8, no. 2, pp. 387–427, Jun. 2009. [Online]. Available: http://dx.doi.org/10.1007/s11047-008-9068-x

[22] W. Lam and J. Mostafa, “Modeling user interest shift using a bayesian approach,” *Journal of the American Society for Information Science and Technology*, vol. 52, no. 5, pp. 416–429, 2001.

[23] G. Giannakopoulos, V. Karkaletsis, G. Vouros, and P. Stamatopoulos, “Summarization system evaluation revisited: N-gram graphs,” *ACM Trans. Speech Lang. Process.*, vol. 5, no. 3, pp. 1–39, 2008.

[24] M. Hearst, S. Dumais, E. Osman, J. Platt, and B. Scholkopf, “Support vector machines,” *IEEE Intelligent systems*, vol. 13, no. 4, pp. 18–28, 1998.

[25] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[26] V. Vapnik, “Structure of statistical learning theory,” *Computational Learning and Probabilistic Reasoning*, p. 3, 1998.

[27] A. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[28] J. Allan, “Incremental relevance feedback for information filtering,” in *Proc. of the 19th annual Int. ACM SIGIR Conf. on Research and development in information retrieval*. ACM New York, NY, USA, 1996, pp. 270–278.

[29] U. Cetintemel, M. Franklin, and C. Giles, “Self-adaptive user profiles for large-scale data delivery,” in *Data Engineering, 2000. Proc. of 16th Int. Conf. on*, 2000, pp. 622–633.

[30] M. Hollander and D. Wolfe, “Nonparametric statistical methods,” *New York*, p. 518, 1973.

[31] W. Cleveland, “LOWESS: A program for smoothing scatterplots by robust locally weighted regression,” *American Statistician*, pp. 54–54, 1981.

Appendix A. The Bias of the Random Sampler

To determine whether we have a bias concerning the frequency sampling from the uniform distribution over the values [70,210], we perform bootstrapping over our sample values for all folds and we measure the mean difference between the input value and a randomly generated, equally sized, sequence of uniformly sampled numbers from the same interval (generated by the R statistical package). In every bootstrap, a new sequence of random numbers is used. This is repeated 20000 times and we can then estimate the mean difference between the means, within a 99% interval.

The two-tailed equi-tailed interval returned for the 99% confidence level indicates that it is most probable that the mean difference between the numbers generated within the experiment and a gold-standard generator range from (approximately) -13 to to 2, which is a pretty good indication that our generated values can be considered as uniformly distributed within the 50, 150 interval.

A Proactive Energy-Efficient Technique for Change Management in Computing Clouds

Hady AbdelSalam Kurt Maly
Ravi Mukkamala Mohammad Zubair
Computer Science Department, Old Dominion University
Engineering & Computational Sciences Building,
Norfolk, VA 23529, USA
{asalam, maly, mukka, zubair}@ cs.odu.edu

David Kaminksy
Software Strategy and Technology, IBM
Research Triangle Park, NC 27709, USA
dlk@us.ibm.com

***Abstract*—The intensive use of portable and thin client devices along with the continuously increasing cost of IT management have pushed large IT companies to look for solutions that allow the use of such devices to access the massive computing power of supercomputers available at the company. The cloud computing concept has emerged with promises to simplify and speed up application deployment and maintenance. All these benefits should be provided for a small fraction of current maintenance cost. To be able to provide services to its customers, a cloud requires high level of maintenance and an appropriate strategy for change management. Replacing defective items (hardware/software), applying security patches, or upgrading firmware are just few examples of typical maintenance procedures needed in such environment. While taking resources down for maintenance, applying efficient change management techniques is a key factor to the success of the cloud. Moreover, the increasing cost of energy consumption in such systems has imposed an additional constraint on proposed techniques making the problem more challenging. In this paper, we propose a proactive energy efficient technique for change management in cloud computing environments. We formulate the management problem into an optimization problem to minimize the total energy consumption of the cloud. We validate our analytical model by providing scenarios that illustrate the mathematical relationships for a sample cloud and that provide a range of possible power consumption savings for different environments.**

***Keywords*-Cloud Computing, Autonomic Manager, Policy languages, Change Management, Energy Efficient.**

I. Introduction

A computing cloud [4] can be defined as a pool of computer resources that can host a variety of different workloads, ranging from long-running scientific jobs (e.g., modeling and simulation) to transactional work (e.g., web applications and payroll processing). A cloud computing framework should be able to autonomously and dynamically provision, configure, reconfigure, and deprovision servers as needed in order to satisfy the needs of the cloud users. Servers in the cloud can be physical machines or virtual machines. Cloud-hosting facilities, including many large businesses that run clouds in-house, became more common as businesses tend to outsource their computing needs more and more.

While intermixing workload can lead to higher resource utilization, we believe that the use of subclouds will be more appropriate for future clouds. Not all hardware is created equal: high-end workstations often contain co-processors that speed scientific computations; lower-end workstations can be appropriate for limited I/O requirements; mainframe computers are designed for efficient intensive computing; and so on. For efficiency reasons, we believe that workloads will be partitioned and assigned to subclouds comprised of homogeneous hardware, which is suitable for executing the assigned workloads. A cloud infrastructure can be viewed as a cost-efficient model for delivering information services and reducing IT management complexity. Several commercial realizations of computing clouds are already available today (e.g., Amazon, Google, IBM, Yahoo, etc.) [7].

Managing large IT environments can be expensive and labor intensive [1], [2]. Typically, servers go through several software and hardware upgrades. Maintaining and upgrading the infrastructure, with

minimal disruption and administrative support, can be extremely challenging especially for complex IT environments where a change in some part of the network may have unexpected impacts on other parts due to the complex connectivity and dependency relationships. Currently, most IT organizations handle change management through human group interactions and coordination. Such a manual process is time consuming, informal, and not scalable, especially for a cloud computing environment. The continuously increasing cost of energy consumption in IT systems has imposed an additional constraint on the management problem making it more challenging [5], [6].

The impact of high power consumption is not just limited to the energy cost but extends to the cost of initial investment of the cooling systems needed to get rid of the generated heat and the continuous cost needed to power these systems. To reduce operational cost at these centers while meeting any performance based SLAs (Service Level Agreement), efficient techniques are needed to provision the right number of resources at the right time.

Several software techniques like operating system virtualization, and Advanced Configuration and Power Interface (ACPI) have been proposed to reduce server energy consumption. Other hardware techniques have also been proposed like processor throttling, dynamic voltage and frequency scaling (DVFS), and low-power DRAM states. Modern computing devices have the ability to run at various frequencies each one with a different power consumption level. Hence, the possibility exists to choose frequencies at which different servers run to optimize total power consumption while staying within the constraints of the SLA that govern the running applications.

The process of updating both software and hardware as well as taking them down for repair and/or replacement is commonly referred to as change management. In an earlier work [1], we proposed and implemented an infrastructure-aware autonomic manager for change management. In [2], we enhanced our proposed autonomic manager by integrating it with a scheduler that can simplify change management by proposing open time slots in which changes can be applied without violating any of SLAs reservations represented by availability policies requirements. Motivated by the importance of developing energy efficient techniques, we extend our previous work by proposing a pro-active energy-aware technique for change management in a cloud computing environment.

In this paper, we analyze the mathematical relationship of these SLAs and the number of servers that should be used and at what frequencies they should be running. We discuss a proactive provisioning model that includes hardware failures, devices available for services, and devices available for change management, all as a function of time and within constraints of SLAs. We validate our analytical model by providing scenarios that illustrate the mathematical relationships for a sample cloud and that provides a range of possible power consumption savings for different environments. In other ways, we simply develop a mathematical model that will - under certain assumptions - allow system administrators to calculate the optimal number of servers needed to satisfy the aggregate service needs committed by the cloud owner along with the computation of the frequencies the servers should use.

It is instructive for the reader at this point to note the differences between the analytical model presented in this paper and the model we previously published in [3]. In the previous model, the derivation for the optimal number of servers, k, was based on two main assumptions: (1) The expected cycles per instruction is 1, (2) The approximation that for the same cloud load, the sum of normalized frequencies $(L = \sum_i f_i)$ is independent from the number of running servers k. Unfortunately, this turns out not to be true when servers running frequencies were normalized within $[0, 1]$ range. In this paper, we get rid of this approximation by representing k as a function of the running frequencies. We also generalize the model for processors with $CPI > 1$. Although, the resulting formula we obtained from new model are much more complex than the simplified model, the results turns out to be more accurate. Furthermore, we tried to validate the new model by providing more scenarios.

The remainder of this paper is organized as follows. In Section II, we describe how interactive jobs can be distributed over servers in clouds. In Section III we provide our assumptions about the underlying infrastructure, and the power consumption analysis model we assume in the paper. In Section IV, we provide a robust analysis for energy consumption within the cloud. In particular, we provide two formulas that can be used to obtain optimal number of servers and optimal running frequencies for these servers. In Section V, we apply the equations from our analysis to change management and provide various scenarios to illustrate the power savings one can expect for various cloud environments. Section VI gives our conclusions and future work.

II. Job Distribution Problem

It is important to understand the complexity of the problem of distributing jobs to servers. Actually, this problem can be viewed as a modified instance of the bin packing problem [8] in which n objects of different sizes must be packed into a finite number of bins each with capacity C in a way that minimizes the number of bins used. Similarly, we have n jobs each with different processing requirements; we would like to distribute these jobs into servers with limited processing capacity such that the number of servers used is kept to the minimum. Being an NP-hard problem, there is no fast polynomial time algorithm available to solve the bin packing problem. Next, we attempt to simplify the general problem to a more constrained version for which we can obtain a solution efficiently.

Basically, applications that run in a cloud computing environment can be broadly classified into two different types. The first type includes applications that require intensive processing and usually these are non-interactive applications. The best strategy to run such applications in a cloud environment is to dedicate one or more powerful servers to each of these applications. Obviously, the number of dedicated servers depends on the underlying SLA and the availability of servers in the cloud. These servers should run at their top speed (frequency) so the application can finish as soon as possible. The reason behind this strategy is to allow dedicated servers to be idle for longer periods saving their total energy consumption.

On the other hand, the second type includes applications that heavily depend on user interaction. Web applications and web services are typical examples of this type of applications. Although, in general, interactive applications do not require intensive processing power, they have many clients. If the number of clients for any of these applications is large, then it might be appropriate to run multiple instances of the same application on different servers and balance the load of each server to satisfy the required response time determined by the SLA. Due to the overwhelming number of web based applications available today, it is highly expected to find these applications more common in a cloud computing environment; hence, in this paper we focus on user interactive applications.

As shown below, by focusing on interactive applications, we simplify the problem of distributing jobs into servers. The key idea behind this simplification is to make a job divisible over multiple servers. To clarify this point, we introduce the following example. Assume that, based on its SLA; Job X requires s seconds response time for u users. From the historical data for Job X, we estimate the average processing required for a user query to be I instructions. Assume that job X is to be run on a server that runs on frequency F and on the average requires CPI clock ticks (CPU cycles) to execute an instruction. Within s seconds the server would be able to execute $\frac{s \cdot F}{CPI}$ instructions. Thus, the server can execute $q = \frac{s \cdot F}{I \cdot CPI}$ user queries within s seconds. Basically, if $q < u$, then the remaining $(u - q)$ user requests should be routed to another server. This can be done through the load balancer module at the cloud gateway. When a new job is assigned to the cloud, the job scheduler analyzes the associated SLA and processing requirements of the new job. Based on this information and the availability of servers, the job scheduler module estimates total processing requirements and assigns this job to one or more of the cloud servers.

III. System Model and Assumptions

A cloud consists of a number of server groups; each group has a number of servers that are identical in hardware and software configuration. All the servers in a group are equally capable of running any application within their software configuration. Cloud clients sign a service level agreement SLA with the company running the cloud. In this agreement, each client determines its needs by aggregating the processing needs of its user applications, the expected number of users, and the average response time per user request.

To be able to estimate the computing power (MIPS) needed to achieve the required response time, the client should provide the cloud administrators with any necessary information about the type of the queries expected from its users. One way of doing this is through providing a histogram that shows the frequency of each expected query. Cloud administrators run these queries on testing servers and estimate their computing requirements from their response time. Based on the frequency of each query, cloud administrators can estimate average computing requirement for a user query.

Average response time for a user query depends on many factors, i.e., the nature of the application, the configuration of the server running the application, and the load on the server when running the application. To reduce the number of factors and to simplify our mathematical model, we replace the minimum average response time constraint in SLA by the minimum number of instructions that the application is allowed to execute every second. This kind of conversion can be easily done as follows. If user query has average response time of t_1 seconds when it runs solely on a server configuration with x MIPS (million instructions per second, this can be benchmarked for each server configuration), then to have an average response time of t_2 seconds, it is required to run the query such that it can execute a minimum of $\frac{t_1 \cdot x}{t_2}$ million instructions per second. We assume that each application can be assigned to more than one server to achieve the required response time. When a server runs, it can run on a frequency between F_{min} (the least power consumption) and F_{max} (the

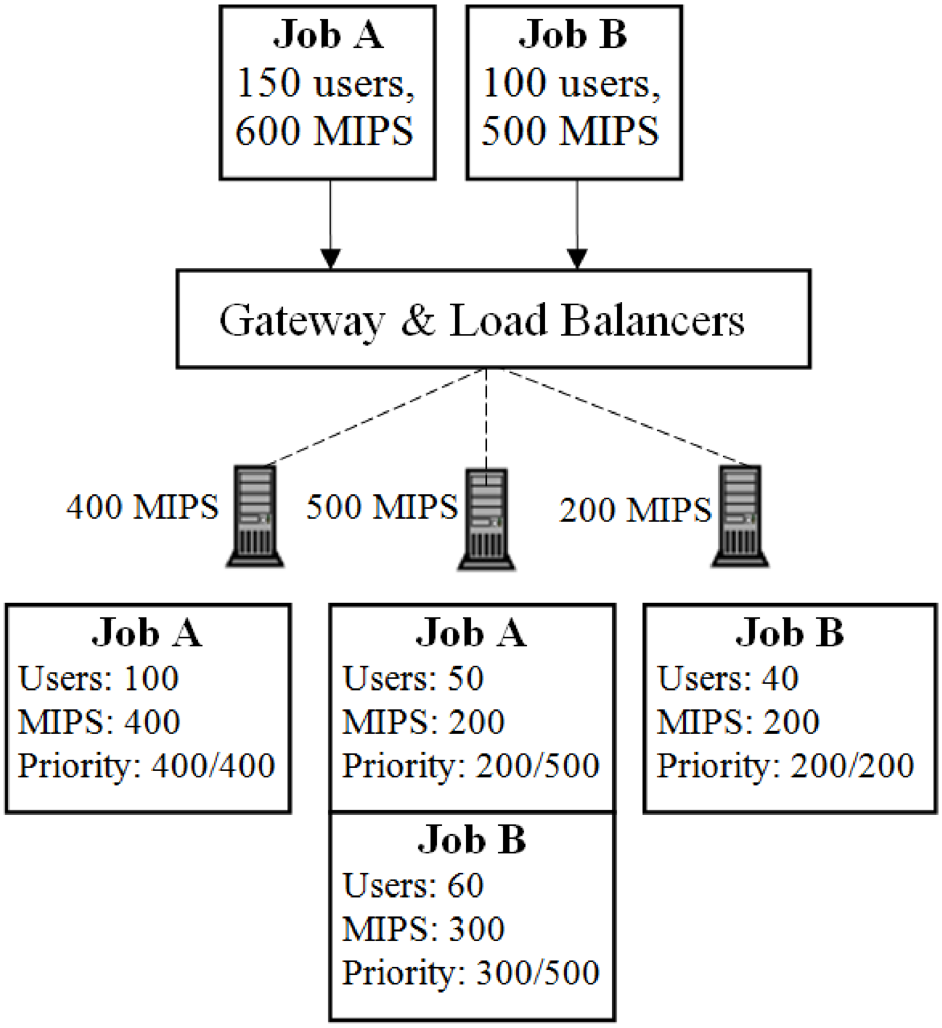


Fig. 1. Distribution of jobs onto servers

highest power consumption), with a range of discrete operating frequency levels in-between.

In general, there are two mechanisms available today for managing the power consumption of these systems: One can temporarily power down the blade, which ensures that no electricity flows to any component of this server. While this can provide the most power savings, the downside is that this blade is not available to serve any requests. Bringing up the machine to serve requests would incur additional costs, in terms of (i) time and energy expended to boot up the machine during which requests cannot be served, and (ii) increased wear-and-tear of components (the disks, in particular) that can reduce the mean-time between failures (MTBF) leading to additional costs for replacements and personnel. Another common option for power management is dynamic voltage/frequency scaling (DVS).

The dynamic power consumed in circuits is proportional to the cubic power of the operating clock frequency. Slowing down the clock allows the scaling down of the supply voltages for the circuits, resulting in power savings. Even though not all server components may be exporting software interfaces to perform

DVS, most CPUs in the server market are starting to allow such dynamic control [5], [6]. The CPU usually consumes the bulk of the power in a server (e.g., an Intel Xeon consumes between 75-100 Watts at full speed, while the other blade components including the disk can add about 15-30 Watts) [5]. The previous example shows that DVS control in cloud computing environment can provide substantial power savings.

When the machine is on, it can operate at a number of discrete frequencies where the relationship between the power consumption and these operating frequencies is of the form [3], [5], [6]

$$P = A + B \cdot F^3$$

so that we capture the cubic relationship with the CPU frequency while still accounting for the power consumption of other components (A) that do not scale with the frequency. In Section V, we shall provide sample values of the constants A and B.

Cloud Environment And Assumptions: for a cloud, requests from cloud clients flow to the system through a cloud gateway. After necessary authentication and based on the current load on the cloud servers, a load balancing module forwards client requests as described in Section II to one of the cloud servers dedicated to support this type of requests. This implies that the load balancing module at the cloud gateway should have up-to-date information about which client applications are running on which servers and the load on these servers. In addition, the system has a 'power-optimizer' module that computes the optimal number of servers and operational frequencies for a particular load requirement. Client applications are assigned to servers based on the requirements of the SLA for each client. This process may involve running the same application on several servers and distributing requests of the same client over different servers based on the load on these servers. To distribute the load on cloud servers correctly, the gateway and the load balancers must have access to the traditional schedule information as well as the information from the power-optimizer.

Homogeneity: in the introduction we described the motivation for using homogeneous sub-clouds that exist within a larger cloud infrastructure. Within each sub-cloud, we assume that resources can be treated homogeneously. That does not mean that all computing devices in a sub-cloud are the same, only that all computing devices in the sub-cloud are capable of executing all work assigned to that sub-Cloud. With the increasing adoption of virtualization technology, including Java JVM and VMware images, we believe that this assumption is valid. For the rest of the paper we shall assume that a cloud is homogeneous

Interactive Applications: Applications that run in a cloud computing environment can be broadly classified into two different types. The first type includes applications that require intensive processing; such applications are typically non-interactive applications. The best strategy to run such applications in a cloud environment is to dedicate one or more powerful servers to each of these applications. Obviously, the number of dedicated servers depends on the underlying SLA and the availability of servers in the cloud. These servers should be run at their top speed (frequency) so the application will finish as soon as possible. The reason behind this strategy is to allow dedicated servers to be idle for longer periods saving their total energy consumption.

The second application type is those that depends heavily on user interaction. Web applications and web services are typical examples. Although, in general, interactive applications do not require intensive processing power, they have many clients, leading to a large aggregate processing demand. If the number of clients for any of these applications is large, to satisfy the required response time determined by the SLA, it might be appropriate to run multiple instances of the same application on different servers, balancing the load among them.. Due to the overwhelming number of web based applications available today, such applications are likely to be prevalent in a cloud computing environment; hence, in this paper we focus on user interactive applications. We leave analysis of the former application type to future work.

Power consumption in our model will be manipulated by changing the frequencies at which instructions are executed at a server. As SLAs are typically expressed in many different ways we need to map these compute requirements into a standard form that relates to the number of instructions executed over a period of time. We chose to represent the load an application will put on the cloud in terms of the familiar MIPS. For example, in Fig. 1 we show how a particular client of the cloud has at that time 150 users who require a total of 500 MIPS for the next period of time.

To estimate the computing power (MIPS) needed to achieve the required response time, the client must provide the cloud administrators with any necessary information about the type of the queries expected from its users. One approach is to provide a histogram that shows the frequency of each expected query. Cloud administrators run these queries on testing servers and estimate their computing requirements from their response time. Based on the frequency of each query, cloud administrators can estimate average computing requirement for a user query.

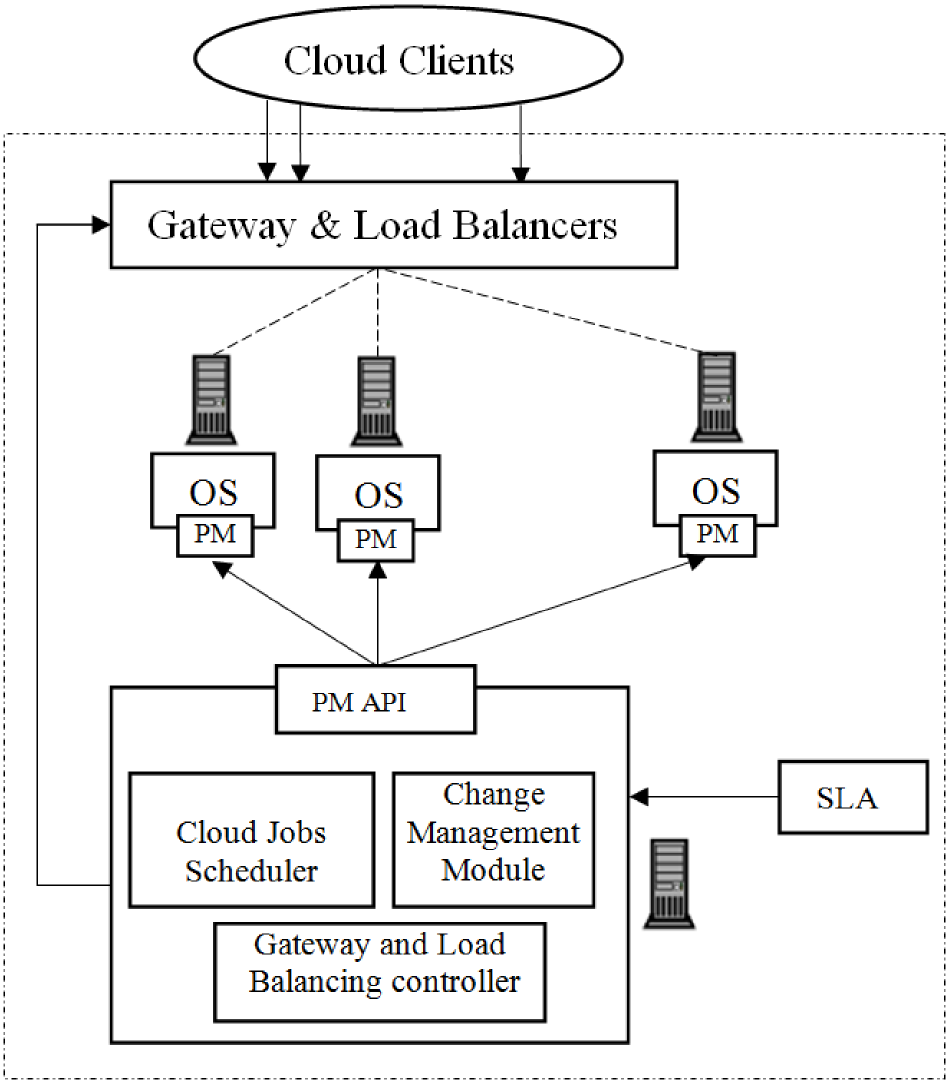


Fig. 2. Cloud Architecture

IV. Power Consumption

To summarize the model assumptions: a cloud consists of a number of server groups; each group has a number of servers that are identical in hardware and software configuration. All the servers in a group are equally capable of running any application within their software configuration. Cloud clients sign a service level agreement SLA with the company running the cloud. In this agreement, each client determines its needs by aggregating the processing needs of its user applications, the expected number of users, and the average response time per user request. When a server runs, it can run on a frequency between (the least power consumption) and (the highest power consumption), with a range of discrete operating frequency levels in-between. In general, there are two mechanisms available today for managing the power consumption of these systems: One can temporarily power down the blade, which ensures that no electricity flows to any component of this server.

While this can provide the most power savings, the downside is that this blade is not available to serve any requests. Bringing up the machine to serve requests would incur additional costs, in terms of (i) time and energy expended to boot up the machine during which requests cannot be served, and (ii) increased wear-and-tear of components (the disks, in particular) that can reduce the mean-time between failures (MTBF) leading to additional costs for replacements and personnel. Another common option for power management is dynamic voltage/frequency scaling (DVS). The dynamic power consumed in circuits is proportional to the cubic power of the operating clock frequency. Slowing down the clock allows the scaling down of the supply voltages for the circuits, resulting in power savings. Even though not all server components may be exporting software interfaces to perform DVS, most CPUs in the server market are starting to allow such dynamic control [5], [6]. The CPU usually consumes the bulk of the power in a server (e.g., an Intel Xeon consumes between 75-100 Watts at full speed, while the other blade components including the disk can add about 15-30 Watts) [5]. The previous example shows that DVS control in cloud computing environment can provide substantial power savings.

When the machine is on, it can operate at a number of discrete frequencies where the relationship between the power consumption and these operating frequencies is of the form [5], [6]

So that we capture the cubic relationship with the CPU frequency while still accounting for the power consumption of other components (A) that do not scale with frequency.

Let $F(t)$ be the total computing load of the cloud at time t. To provide the required computing load, the cloud has k servers that run on frequencies $F_1, F_2, \cdots, F_k$ respectively. We normalize frequencies in the range $[0, 1]$ using,

$$f_i = \frac{F_i - F_{min}}{F_{max} - F_{min}} \qquad (1)$$

Define,

$$L = \sum_{i=1}^{k} f_i \qquad (2)$$

where f_i is the normalized frequency on which server i runs. Assuming that the average clocks per instruction for the cloud servers is CPI, we can relate normalized frequencies to cloud total load as follows.

$$\begin{aligned}
L &= \sum_{i=1}^{k} f_i \\
L &= \sum_{i=1}^{k} \frac{F_i - F_{min}}{F_{max} - F_{min}} \\
\sum_{i=1}^{k} F_i &= (F_{max} - F_{min})L + k \cdot F_{min} \\
\frac{\sum_{i=1}^{k} F_i}{CPI} &= \frac{(F_{max} - F_{min})L + k \cdot F_{min}}{CPI} = F(t) \qquad (3)
\end{aligned}$$

and from equation (3), we can express L as follows,

$$\begin{aligned}
F(t) &= \frac{(F_{max} - F_{min})L + k \cdot F_{min}}{CPI} \\
L &= \frac{F(t) \cdot CPI - k \cdot F_{min}}{F_{max} - F_{min}} \\
&= C_1 + C_2 \cdot k \qquad (4)
\end{aligned}$$

where

$$\begin{aligned}
C_1 &= \frac{F(t) \cdot CPI}{F_{max} - F_{min}} \qquad (5) \\
C_2 &= -\frac{F_{min}}{F_{max} - F_{min}} \qquad (6)
\end{aligned}$$

Total energy consumption is given by,

$$\begin{aligned}
P &= \sum_{i=1}^{k} \left[A + B \cdot f_i^3\right] \\
&= k \cdot A + B \sum_{i=1}^{k-1} f_i^3 + B \left[L - \sum_{i=1}^{k-1} f_i\right]^3
\end{aligned}$$

Now our interest is to evaluate the optimal values for f_i such that the total energy consumption is minimum. Assuming a continuous frequency spectrum, we evaluate the first partial derivative of the total energy consumption with respect to each frequency.

$$\frac{\partial P}{\partial f_i} = 3Bf_i^2 - 3B \left[L - \sum_{i=1}^{k-1} f_i\right]^2 \qquad \forall i \in \{1, \cdots, k-1\}$$

To minimize P, we set $\frac{\partial P}{\partial f_i} = 0$.

$$\begin{aligned}
\frac{\partial P}{\partial f_i} &= 3Bf_i^2 - 3B\left[L - \sum_{i=1}^{k-1} f_i\right]^2 = 0 \\
3Bf_i^2 &= 3B\left[L - \sum_{i=1}^{k-1} f_i\right]^2 \\
f_i &= \left[L - \sum_{i=1}^{k-1} f_i\right] = f_k \qquad \forall i \in \{1, \cdots, k-1\} \\
f_i &= \frac{L}{k} \qquad \forall i \in \{1, \cdots, k\} \qquad (7)
\end{aligned}$$

In other words, to minimize the total energy consumption, cloud servers must run at frequency $\frac{L}{k}$.

Now we turn our interest to evaluate, k, the optimal number of servers to run. Using a similar approach, we first rewrite the equation of total energy consumption after substitution each frequency using equation (7) as follows.

$$\begin{aligned}
P &= \sum_{i=1}^{k} \left[A + B \cdot f_i^3\right] \\
&= k \cdot A + k \cdot B\left[\frac{L}{k}\right]^3 \\
&= k \cdot A + \frac{B(C_1 + C_2k)^3}{k^2} \qquad (8)
\end{aligned}$$

After that we obtain the first derivative of the total energy consumption with respect to k. Mathematically, this can be expressed as,

$$\frac{\partial P}{\partial k} = A + \frac{3BC_2(C_1 + C_2k)^2k^2 - 2B(C_1 + C_2k)^3k}{k^4}$$

Setting $\frac{\partial P}{\partial k} = 0$,

$$\begin{aligned}
Ak^3 &= 2B(C_1 + C_2k)^3 - 3BC_2(C_1 + C_2k)^2k \\
\frac{Ak^3}{B} &= (C_1 + C_2k)^2(2(C_1 + C_2k) - 3C_2k) \\
\frac{Ak^3}{B} &= (C_1 + C_2k)^2(2C_1 - C_2k) \\
\frac{Ak^3}{B} &= (C_1^2 + 2C_1C_2k + C_2^2k^2)(2C_1 - C_2k) \\
\frac{Ak^3}{B} &= 2C_1^3 + 4C_1^2C_2k + 2C_1C_2^2k^2 - C_1^2C_2k \\
&\quad -2C_1C_2^2k^2 - C_2^3k^3
\end{aligned}$$

After rearranging terms,

$$\begin{aligned}
\left[\frac{A}{B} + C_2^3\right]k^3 - 3C_1^2C_2k &= 2C_1^3 \\
k^3 - \frac{3C_1^2C_2}{\frac{A}{B} + C_2^3}k &= \frac{2C_1^3}{\frac{A}{B} + C_2^3} \\
k^3 + pk &= q \qquad (9)
\end{aligned}$$

where

$$p = -\frac{3C_1^2C_2}{\frac{A}{B} + C_2^3} \qquad (10)$$

$$q = \frac{2C_1^3}{\frac{A}{B} + C_2^3} \qquad (11)$$

To solve equation(9), we can use Vieta's substitution [9] by defining x as follows,

$$k = x - \frac{p}{3x} \qquad (12)$$

After substituting k into equation (9),

$$\begin{aligned}
\left[x - \frac{p}{3x}\right]^3 + p(x - \frac{p}{3x}) &= q \\
x^3 - px + \frac{p^2}{3x} - \left[\frac{p}{3x}\right]^3 + px - \frac{p^2}{3x} &= q \\
x^3 - \left[\frac{p}{3x}\right]^3 &= q \qquad (13)
\end{aligned}$$

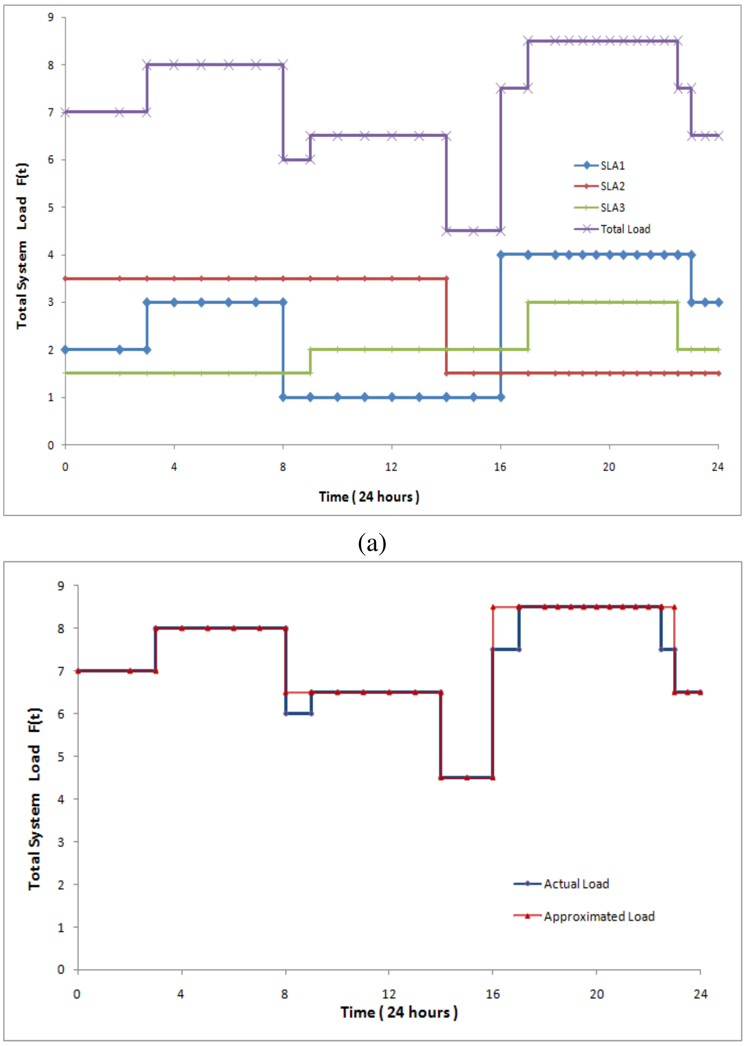
Multiplying both sides of equation (13) by x^3 converts it into the standard quadratic form.

$$\left(x^3\right)^2 - q(x^3) - \frac{p^3}{27} = 0 \qquad (14)$$

Solving equation (14),

$$\begin{aligned}
x^3 &= \frac{q + \sqrt{q^2 + \frac{4p^3}{27}}}{2} \\
&= \frac{q}{2} + \sqrt{\left(\frac{q}{2}\right)^2 + \frac{p^3}{27}} \\
&= \frac{C_1^3}{\frac{A}{B} + C_2^3} + \sqrt{\frac{\left(\frac{A}{B} + C_2^3\right)C_1^6 - C_1^6C_2^3}{\left(\frac{A}{B} + C_2^3\right)^3}} \\
x^3 &= \frac{C_1^3}{\frac{A}{B} + C_2^3} + C_1^3\sqrt{\frac{\frac{A}{B}}{\left(\frac{A}{B} + C_2^3\right)^3}} \\
x &= C_1 \cdot \sqrt[3]{\frac{1}{\frac{A}{B} + C_2^3} + \sqrt{\frac{\frac{A}{B}}{\left(\frac{A}{B} + C_2^3\right)^3}}} \qquad (15)
\end{aligned}$$

From equations (12) and (15), we can evaluate k, the optimal number of servers, that should by running in the cloud to optimize power consumption. Interestingly, the value of k depends on the constants A and B, the frequency range of server processors (i.e., F_{min}, F_{max}), and the total computing load $F(t)$.



(a)

(b)

Fig. 3. (a) Cloud total load as imposed by SLAs (b) Actual vs. Approximated total load due to several SLAs

V. Change Management Scenarios

The computing load of the cloud can be expressed as a function of time, and it usually changes when a new application is started or completed. Typically, the sum of the commitments in service level agreements is periodic with a daily, weekly or even monthly period. To include the dynamic nature of the cloud load into our model, we divide the time line into slots. During one slot, the cloud total load does not change. To eliminate minor changes in the total cloud load curve, we approximate the load on the cloud by an upper bound envelope of this curve such that the length of any slot is larger than a predetermined threshold.

The derivations in Section IV for the optimal number of servers and optimal running frequencies should not change for this simplification. All we need to do is to replace L by $L(t)$. Thus, in each time segment, the number of idle servers in the cloud equals the difference between the total number of cloud servers and $k(t)$. An idle server is a candidate for change management. Figure 4, shows a plot for the number of servers available for change management

based on the cloud load determined in Figure 3-a.

We define change management capacity as the average number of servers available for changes per time unit. Based on this definition, change management capacity can be evaluated as the area under the curve of Figure 4 divided by the periodicity of the SLAs. For example, for the cloud load shown in Figure 3-b, change management capacity as follows,

$$\begin{aligned}\textbf{Capacity} &= \frac{3\cdot 5+5\cdot 4+6\cdot 5+2\cdot 7+7\cdot 3+1\cdot 5}{24}\\ &= \frac{75}{24} = 3.125 \textbf{ servers/hour.}\end{aligned}$$

From historical and statistical change management information about the applications running on the cloud, administrators can estimate change management capacity requirements for these applications. This can be used to determine the optimal number of servers to have in the cloud for a particular set of clients. Particularly, the area under the curve in Figure 4 is proportional to the number of available servers for maintenance in the cloud. Basically, this area can be adjusted by raising the curve up or down, which in turn can be done by changing the total number of servers in the cloud. For example, in Figure 4, we estimated cloud change management capacity to be 3.125 servers/unit time. If statistics showed that applications running on the cloud require an average of 4 servers to be available for changes per unit time, then the total number of servers in the cloud should be increased by one server to satisfy change management requirements.

It is worthwhile to mention that under our proposed model it is straightforward to incorporate hardware failures into the model, increasing the reliability of the cloud. Hardware failure rates can be statistically estimated using information about previous hardware failures, expected recovery rates, hardware replacement strategies, and expected lifetime of the hardware equipments. Given hardware failure rate expressed in terms of failed servers per unit time, cloud applications change requirements can be adjusted to reflect hardware failures. The new change management capacity is estimated based on the sum of application changes requirements and hardware failure requirements. In the previous example of Figure 3-b, if the hardware failures rate is less than 0.875 failed servers/hr, then an average of 4 servers available per unit time is enough to satisfy change and hardware failure requirements. However, if hardware failures rate goes above 0.875 servers/hr, then additional servers are needed.

In this section, we compare the performance of our proposed change management technique against other techniques based on over-provisioning. The main idea behind these techniques is to overly provision computing resources to compensate for any failure or change management requirements. Our calculations for over-provisioning techniques assumes a 5% over-provisioning rate, which means that the available computing power available at any time is 5% higher than what is needed to satisfy the service level agreements.

In our scenario, we assume a computing cloud with 125 servers. Each server has a range of discrete running between $1-3$ GHz. Maximum processing power is achieved when the server runs at 3 GHz. We assume that if the required running frequency is unavailable, the next higher available frequency will be used. To numerically correlate the running frequency with the achieved computing power, we must estimate the average number of cycles needed to execute one instruction on any of these servers. Given running frequency, F, and number of cycles per instruction, CPI the computing power of a server can be estimated in MIPS, million instructions per second, as $\frac{F}{CPI\cdot 10^6}$. In this scenario, we set CPI to 3.00 cycles/instructions. To be able to measure energy consumption, we assume the energy model described in Section III $(i.e., P = A + BF_n^3)$ where A and B are system constants, is the normalized running frequency. In our scenario, we use the same values of the constants A and B as was published in [6]. This also requires to normalize the running frequency to the range [0,1], where stands for the minimum running frequency (1.0 GHZ), and stands for the maximum running frequency. Mathematically this can be obtained through, $F_n = \frac{F-F_{min}}{F_{max}-F_{min}}$.

We assume that the cloud administrators have determined that they will need a change management capacity of 1.2 servers/hr. For a more realistic scenario, we include also server failures in our model. We assume a failure rate of 0.6 servers per hour, which is included as an additional computing requirement. In terms of periodic load, here we assume a period of one day.

During the day, the cloud total load changes as described in Table I. To remind the reader, this information is obtained from client applications historical data and is expressed in SLAs.

Under these assumptions, we compare total energy consumption using our proposed approach against using a 5% over-provisioning. Figure 7 shows how using our approach can reduce cloud energy consumption against over-provisioning. Both approaches can achieve the required level of change management capacity.

Figures 5 and 6 respectively show the total energy consumption and the associated number of servers needed for different frequency ranges. From the figures, it is easy to verify that the optimal running

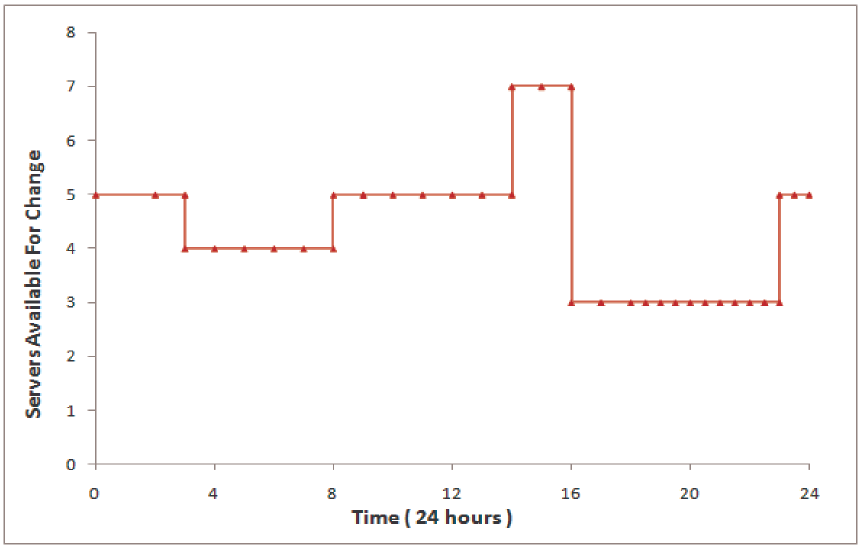


Fig. 4. Servers available for changes as a function of time

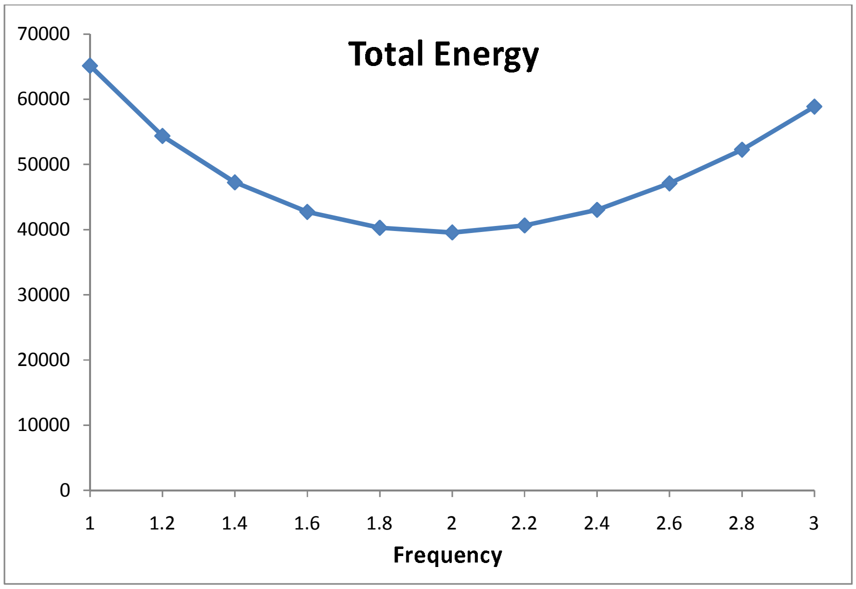


Fig. 5. Total energy consumption when using different frequencies

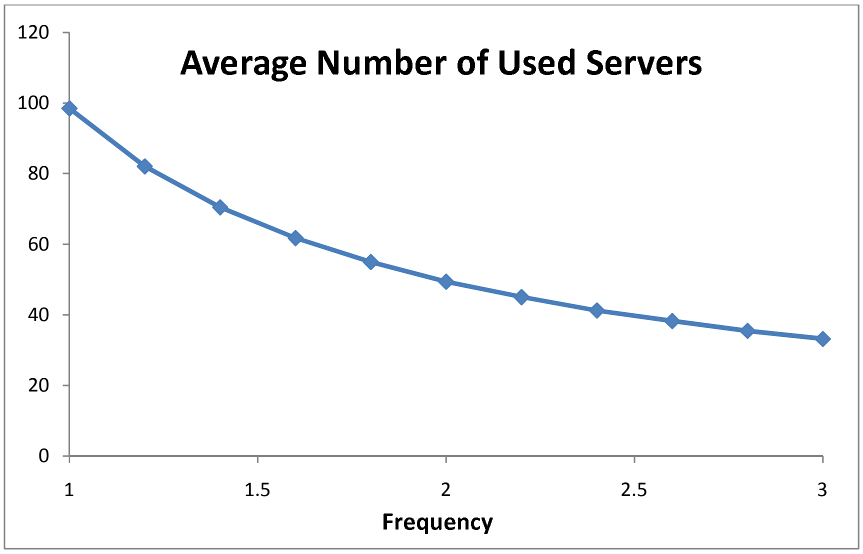


Fig. 6. Average number of servers when using different frequencies

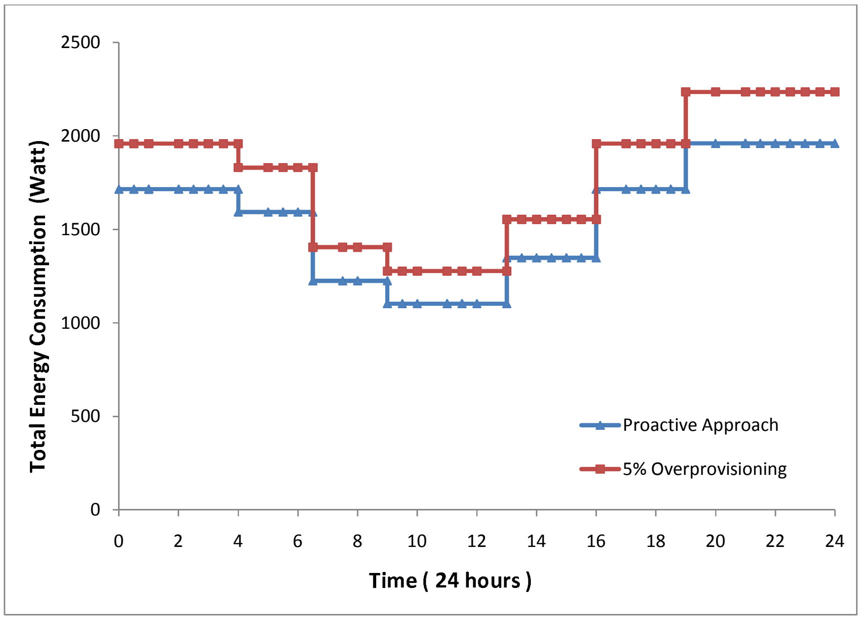


Fig. 7. Proactive approach vs. 5% Over-Provisioning

TABLE I
CLOUD LOAD DURING DIFFERENT TIMES OF THE DAY

From	To	Cloud Load (BIPS)
12:00 AM	04:00 AM	70
04:00 AM	06:30 AM	65
06:30 AM	09:00 AM	50
09:00 AM	01:00 PM	45
01:00 PM	04:00 PM	55
04:00 PM	07:00 PM	70
07:00 PM	12:00 AM	80

TABLE II
TOTAL ENERGY CONSUMPTION IN THE CLOUD DURING ONE DAY ASSUMING 5% PROVISIONING

Frequency	Total(Watt.Hour)	Average(Watt)
1.0 GHZ	64861	2703
2.0 GHZ	39325	1639
2.4 GHZ	42743	1781
3.0 GHZ	58246	2427

frequency is around 2 GHZ as determined by our proposed formula.

We also compare the total energy consumption during one period (one day) using both approaches. Under the proactive approach, the total energy consumption is evaluated to be 37304.76 Watt.Hour, for an average of 1554.36 Watt. Table II, summarizes the total and the average energy consumption when using 5% over-provisioning. Table II shows that using the proactive approach, cloud total energy consumption is smaller than energy consumption using 5% over-provisioning for different running frequencies.

VI. CONCLUSIONS

In this paper we have created a mathematical model for power management for a cloud computing environment that primarily serves clients with interactive applications such as web services. Our mathematical model allows us to compute the optimal (in terms of total power consumption) number of servers and the frequencies at which they should run. We show how the model can be extended to include the needs for change managements and how the other type of typical applications (computing intensive) can be included. Further, we extend the model to account for hardware failure. In Section V, we compare our scheme against various over-provisioning scheme For example, with a cloud of 125 servers, a change management capacity of 1.2 servers/hr, a failure rate of 0.6 servers/hr, the total power consumption for a day with our scheme is $37,305$ Watt.Hour versus $64,861$ Watt.Hour at 5% over-provisioning, and the savings range from $5-74\%$ for various parameters of the cloud environment. In the future we plan to relax some of the model's simplifying assumptions. In particular we can rather straightforwardly adapt the model to have the frequencies assume discrete values rather than be part of a continuous function.

REFERENCES

[1] H. AbdelSalam, K. Maly, R. Mukkamala, and M. Zubair. Infrastructure-aware autonomic manager for change management. In POLICY '07: Proceedings of the Eighth IEEE International Workshop on Policies for Distributed Systems and Networks, pages 66–69, Washington, DC, USA, 2007. IEEE Computer Society.

[2] H. AbdelSalam, K. Maly, R. Mukkamala, M. Zubair, and D. Kaminsky. Scheduling-capable autonomic manager for policy based it change management system. In EDOC: Proceedings of the 12th International IEEE Enterprise Distributed Object Computing Conference, pages 386–392, München, Germany., September 15-19 2008. IEEE Computer Society.

[3] H. S. Abdelsalam, K. Maly, R. Mukkamala, M. Zubair, and D. Kaminsky. Analysis of energy efficiency in clouds. In COMPUTATIONWORLD '09: Proceedings of the 2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, pages 416–421, Washington, DC, USA, 2009. IEEE Computer Society.

[4] G. Boss, P. Malladi, D. Quan, L. Legregni, and H. Hall. Cloud computing. High Performance On Demand Solutions (HiPODS), Web Resource available online at: http://www.ibm.com/developerworks/websphere/zones/hipods/, October 2007. IBM.

[5] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. Managing server energy and operational costs in hosting centers. In SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pages 303–314, New York, NY, USA, 2005. ACM.

[6] M. Elnozahy, M. Kistler, and R. Rajamony. Energy-efficient server clusters. In PACS'02: Proceedings of the 2nd Workshop on Power-Aware Computing Systems, in conjunction with HPCA-8., pages 179–196, Cambridge, Ma, USA, February 2002.

[7] H. Erdogmus. Cloud computing: Does nirvana hide behind the nebula? IEEE Software, 26(2):4–6, March–April 2009.

[8] M. R. Garey and D. S. Johnson. Computers and Intractability; A Guide to the Theory of NP-Completeness, pages 10–20. W. H. Freeman & Co., New York, NY, USA, 1990.

[9] E. W. Weisstein. Vieta's substitution. From MathWorld – A Wolfram Web Resource available online at: http://mathworld.wolfram.com/VietasSubstitution.html, June 2010.

www.iariajournals.org

International Journal On Advances in Intelligent Systems

ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS

issn: 1942-2679

International Journal On Advances in Internet Technology

ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING

issn: 1942-2652

International Journal On Advances in Life Sciences

eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO

issn: 1942-2660

International Journal On Advances in Networks and Services

ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION

issn: 1942-2644

International Journal On Advances in Security

ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS

issn: 1942-2636

International Journal On Advances in Software

ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS

issn: 1942-2628

International Journal On Advances in Systems and Measurements

ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL

issn: 1942-261x

International Journal On Advances in Telecommunications

AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA

issn: 1942-2601