Advances in Intelligent Systems and Computing 461

Maciej Grzenda
Ali Ismail Awad
Janusz Furtak
Jarosław Legierski *Editors*

Advances in Network Systems

Architectures, Security, and Applications

Springer

Advances in Intelligent Systems and Computing

Volume 461

About this Series

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within "Advances in Intelligent Systems and Computing" are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

More information about this series at http://www.springer.com/series/11156

Maciej Grzenda · Ali Ismail Awad
Janusz Furtak · Jarosław Legierski
Editors

Advances in Network Systems

Architectures, Security, and Applications

Springer

Editors
Maciej Grzenda
Faculty of Mathematics and Information Science
Warsaw University of Technology
Warsaw
Poland

and

Research and Development Center
Orange Polska
Warsaw
Poland

Ali Ismail Awad
Department of Computer Science, Electrical and Space Engineering
Luleå University of Technology
Luleå
Sweden

and

Faculty of Engineering
Al Azhar University
Qena
Egypt

Janusz Furtak
Military University of Technology
Warsaw
Poland

Jarosław Legierski
Faculty of Mathematics and Information Science
Warsaw University of Technology
Warsaw
Poland

and

Research and Development Center
Orange Polska
Warsaw
Poland

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-44352-2 ISBN 978-3-319-44354-6 (eBook)
DOI 10.1007/978-3-319-44354-6

Library of Congress Control Number: 2016951683

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Owing to the ever growing communication systems, modern networks currently encompass a wide range of solutions and technologies, including wireless and wired networks, and provide a basis for network systems from multiple partly overlapping domains such as the Internet of Things (IoT), cloud services, and network applications. This appears in numerous active research areas with particular attention paid to the architectures and security of network systems. In parallel, novel applications are developed, in some cases strongly linked to rapidly developing network-based data acquisition and processing frameworks.

In the domain of architectures, growing distribution of components interconnected in variety of ways is observed. This is exemplified by the growth of wireless sensor networks and development of algorithms they require. At the same time, information security works as a backbone for protecting both user data and electronic transactions in network systems. Security has emerged as an important scientific discipline whose many multifaceted complexities deserve the attention and synergy of the computer science, engineering, and information systems communities. Equally importantly, novel applications which both arise from and promote further development of network systems are evolved.

The significance of this book volume comes from the demand for a better understanding of the network systems. The volume provides a comprehensive selection of cutting-edge state-of-the-art algorithms, technologies, and applications, providing new insights into a range of fundamentally important topics in network infrastructures, network security, and network applications.

The volume includes 19 chapters in total that are divided into three parts. Part I is devoted to network architectures and is composed of 6 chapters. Part II includes 6 chapters that cover several network security aspects. The final 6 chapters grouped in Part III are covering some network applications. Additionally, an introduction chapter, Chapter "Network Architectures, Security, and Applications: An Introduction," is placed at the beginning of the volume for offering preliminary information for all the chapters in the three parts of the volume.

The volume has attracted authors from many countries worldwide such as France, Poland, Portugal, Romania, and United Kingdom. Several of the chapters result from the further research made by the authors of selected papers presented during the International Conference on Innovative Network Systems and Applications (iNetSApp) organized under the frame of Federated Conference on Computer Science and Information Systems.

The editors are very grateful to Dr. Janusz Kacprzyk, the editor of the Advances in Intelligent Systems and Computing (AISC) series by Springer for his support, which made the development of this module possible. The editors are indebted to the efforts of Dr. Thomas Ditzinger, the senior editor of the AISC series, and Mr. Holger Schäpe, the editorial assistant of the AISC. Finally, the editors and the authors acknowledge the efforts of Advances in Intelligent Systems and Computing team at Springer for their support and cooperation in publishing the book as a volume in the AISC series by Springer.

Warsaw, Poland Maciej Grzenda
Luleå, Sweden Ali Ismail Awad
Warsaw, Poland Janusz Furtak
Warsaw, Poland Jarosław Legierski
May 2016

Program Committee

Ali Ismail Awad, Luleå University of Technology
Andrzej Białas, Institute of Innovative Technologies
Peter Brida, University of Zilina, Faculty of Electrical Engineering, Department of Telecommunications and Multimedia
Hacene Fouchal, Universite de Reims Champagne-Ardenne
Janusz Furtak, Military University of Technology
Antonio García-Domínguez, Departamento de Ingeniería Informática, Universidad de Cádiz
Gebhard Geiger, Technische Universitaet Muenchen
Maciej Grzenda, Warsaw University of Technology
Zineb Habbas, LCOMS, University of Lorraine
Michal Hodon, University of Zilina
Bernhard Hämmerli, Acris GmbH
Krzysztof Kaczmarski, Warsaw University of Technology
Ondrej Karpis, University of Zilina
Ioannis Katakis, University of Athens
Piotr Korbel, Lodz University of Technology
Emil Kowalczyk, Orange Polska
Jarosław Legierski, Orange Polska
Marcin Luckner, Warsaw University of Technology
Zakaria Maamar, Zayed University
Juraj Micek, Zilinska univerzita
Malgorzata Pankowska, University of Economics
Jan Papaj, Technical University of Košice
Eman Shaaban, Ain Shams University, Cairo
Zbigniew Suski, Military University of Technology
Maciej Szmit, Orange Polska
Devinder Thapa, University of Agder
Konrad Wrona, NATO Communications and Information Agency
Mee Yang, Auckland University of Technology
Zbigniew Zieliński, Military University of Technology

Contents

Part I Network Architectures

Part II Network Security

Part III Network Applications

About the Editors

Dr. Maciej Grzenda received his M.Sc. in computer science in 1997 from the Warsaw University of Technology, Poland. In 2001, he received his Ph.D. degree in technical sciences from the same university. He is currently an assistant professor at the Faculty of Mathematics and Information Science of the Warsaw University of Technology. In parallel, he participates as an expert and project manager in the works of Research and Development Center of Orange Polska. His areas of expertise are industrial applications of intelligent data processing techniques and the architecture of data processing systems with particular emphasis on storage and stream processing with Big Data frameworks. Maciej Grzenda participates also in program committees of international conferences on intelligent data processing.

Dr. Ali Ismail Awad is currently a senior lecturer (assistant professor) at Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden. He also holds a permanent position as an assistant professor at Electrical Engineering Department, Faculty of Engineering, Al Azhar University, Qena, Egypt. Dr. Awad received his B.Sc. from Al Azhar University, Egypt, 2001, the M.Sc. degree from Minia University, Egypt, 2007, and the Ph.D. degree from Kyushu University, Japan, 2012. He has been awarded his second Ph.D. degree from Minia University, Egypt, May 2013. Dr. Awad serves as an external reviewer in several international journals. His research interests include information security, information security laboratories, biometrics, image processing, pattern recognition, and networking.

Dr. Janusz Furtak received his M.Sc. from the Cybernetics Faculty of Military University of Technology, Warsaw, Poland in 1982. For eight years, he was a member of the design team which developed software for command systems. Since 1990, he has been a university teacher at the Cybernetics Faculty of Military University of Technology. In 1999, he received Ph.D. degree in the field of computer science. Currently, he is an assistant professor in the Institute of Teleinformatics and Automation of Cybernetics Faculty, Military University of Technology and Director of this Institute. His main areas of expertise are computer networks, network security, security in Internet of Things, cyber defense, and administering of network operating systems.

Dr. Jarosław Legierski received his M.Sc. in electronics and telecommunication and Ph.D. degree in electronics from Technical University of Lodz. Since 1998, he has worked in the telecommunications industry. He is currently R&D Expert in Research and Development Center, Orange Labs at Orange Polska and assistant professor at Faculty of Mathematics and Information Science of Warsaw University of Technology. Jarosław Legierski is the cocreator of Open Middleware 2.0 Community (www.openmiddleware.pl). His research interest includes open application programming interfaces (APIs), Open (Big) Data, and next-generation service delivery platforms. Author of publications in the area of open API and Open Data exposure.

Network Architectures, Security, and Applications: An Introduction

Maciej Grzenda, Janusz Furtak, Jarosław Legierski and Ali Ismail Awad

Abstract Owing to the ever growing communication systems, modern networks currently encompass a wide range of solutions and technologies, including wireless and wired networks and provide basis for network systems from multiple partly overlapping domains such as the Internet of Things (IoT), cloud services, and network applications. This appears in numerous active research areas with particular attention paid to the architecture and security of network systems. In parallel, novel applications are developed, in some cases strongly linked to rapidly developing network-based data acquisition and processing frameworks. This chapter presents a general introduction to the topics of network architectures, security, and applications in addition to short descriptions of the chapters included in this volume.

Keywords Network architectures · Network security · Network applications

M. Grzenda (✉) · J. Legierski (✉)
Faculty of Mathematics and Information Science,
Warsaw University of Technology, Warszawa, Poland
e-mail: m.grzenda@mini.pw.edu.pl

J. Legierski
e-mail: Jaroslaw.legierski@orange.com

M. Grzenda · J. Legierski
Research and Development Center, Orange Polska Warszawa, Warszawa, Poland

J. Furtak (✉)
Military University of Technology, Warszawa, Poland
e-mail: jfurtak@wat.edu.pl

A.I. Awad (✉)
Department of Computer Science, Electrical and Space Engineering,
Luleå University of Technology, Luleå, Sweden
e-mail: ali.awad@ltu.se; aawad@ieee.org

A.I. Awad
Faculty of Engineering, Al Azhar University, Qena, Egypt

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent
Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_1

1 Introduction

A growing proportion of modern software systems are developed as network systems. The growth of network bandwidth, ever growing coverage of mobile networks and development of disruptive network services all contribute to this phenomenon. This volume reflects the variety of undertaken efforts in the research and development community working within the domain of network systems. The first part of this book covers items pertaining to the network architectures. It includes chapters related to server placement in regular networks, client-server architecture, analysis of TCP connections, wireless sensor networks, marker localisation methods, and photonic data transport networks.

We live in the era of dynamic development of Internet technologies. 5G technology, the Internet of Things (IoT) and its successor, the Internet of Everything, advanced applications of virtualization, and cloud computing are examples that can be mentioned in this context. Data security and user authentication are crucial concerns in all of these examples. Data encryption plays a major role in assuring the information confidentiality [1]. On the applications' level, and in addition to the traditional authentication methods, biometrics technology expands and offers a reliable user identification or verification solution, access control mechanism, that is required for most of the available network applications [2–5]. On the network infrastructure level, traffic passing though the network is a rich source of information [6, 7]. Network traffic collection and analysis can be a good tool for addressing security solutions to the network systems.

The second part of the book is dedicated to numerous issues in the security domain. We can find proposals of solving the following problems: securing the transmission in wireless sensor networks, partitioning of security policies in tactical service-oriented architecture, the usage of Trust Management of Credentials (TMC), estimating time delay and energy consumption when using the AES encryption in Wireless Sensor Networks, computer support for risk management in critical infrastructures, and risk management systems for monitoring flood hazards.

Network systems form the critical infrastructure and the foundation of a wide variety of applications. The network infrastructure includes high-speed wired networks, wireless networks, Wireless Sensor Networks, IoT networks, and mobile networks. Due to the diversity of the network systems, several business and industrial applications have emerged for each network infrastructure. The third part addresses several aspects for network applications such as automotive applications, live TV services, telemetry-oriented applications, drip irrigation systems, and energy harvesting platform using Wireless Sensor Networks.

2 Chapters of the Book

The rest of the volume contains 18 chapters which are divided into three categories. The following are brief summaries for the content of each chapter.

Part I: Network Architectures

Chapter "An Analytical Method of Server Placement in Regular Networks and its Evaluation by Simulation Experiments" illustrates the optimization issues present in the design of network systems. The chapter presents the challenge of determining optimal server placements in the hybercube network structure [8], which follows from hypercube structure of multiprocessor systems [9]. The latter network architecture is of particular importance for critical applications present in the field of military, aerospace or medical domains. The way the server placement can be followed by the generation of appropriate communication structure is proposed in the chapter. At the same time, the chapter illustrates how the needs of network systems promote the development of novel algorithms.

Another aspect of network systems that is of growing importance nowadays is the adaptation of these systems to serve user interface on variety of devices. Device-Independent Architecture (DIA) [10] and its relation to user interface adaptation are discussed in Chap. "Model and Case Studies of the Runtime UI Adaptation Process in Client-Server Systems". The chapter shows possible use of existing user interface adaptation in DIA systems and addresses the need for network systems serving their user interface on multiple devices of varied capabilities [11]. This issue can be considered in the context of ubiquitous computing, where dynamic adaptation to not only user interface changes, but also changes of user preferences, profile, location, and context becomes necessary [12]. At the same time, it shows an interesting example of the importance of network protocols reducing the volume of transmitted data. This remains a major objective even though the network bandwidth is constantly growing both in wired and wireless network environments.

Chapter "Analysis of TCP Connection Performance Using Emulation of TCP State" also directly refers to the performance aspects of network protocols. The chapter proposes methodology aiming at the identification of root causes of throughput degradation in TCP connections [13] i.e. connections fundamental for many, or even most modern network applications. Importantly, passive measurements performed through network probes are used to attain the objectives laid out in this chapter. This revisits the use of network probes [14], which is of particular importance for variety of monitoring and security-related purposes such as the collection of data for intrusion detection purposes [15]. What is interesting, the chapter describes the validation of proposed approach performed with the data collected in 4G mobile network. Therefore, a combination of TCP protocol and mobile environments serves to illustrate the chapter and its findings.

Chapter "Firefly-Based Universal Synchronization Algorithm in Wireless Sensor Network" also contributes to the survey of architecture challenges contained in this part of the book. The chapter concentrates on one of the crucial needs of the majority of Wireless Sensor Networks (WSN), being the synchronization of nodes. The algorithm answering this need and based on the fireflies synchronization process [16] is proposed and tested. The synchronisation algorithm proposed in the chapter belongs to a wider class of algorithms stimulated by the development of WSN [17]. The experimental evaluation of the proposed synchronization algorithm has been performed on physical wireless sensor nodes. This clearly illustrates novel challenges introduced with WSN deployments which involve the need for developing and testing network architectures with dedicated hardware devices. Moreover, the challenges caused by the need to minimize active power modes have to be addressed. These go beyond the needs typically present in wired scenarios with no restrictive power consumption constraints.

Chapter "Comparison of Various Marker Localization Methods" extends the analysis of unique needs of WSN networks to concentrate on the cases where even locating a network node can be a challenge. The chapter analyses two methods of marker localization, which rely on Radio Frequency Identification (RFID) [18]. The work addresses the need for underground localisation, discussed among others in [19]. By a marker, which has to be localised a passive RFID transponder (without or with identification chip) is meant. The term marker reflects the fact that the transponder is used to mark underground assets and trace underground networks such as cables and pipes in turn. Localization of the marker is based on evaluation of signal amplitude received from the excited marker. Notably, the location of a wireless device is determined based on signal amplitude received from the excited marker. This shows a wider tendency of using network-related data for variety of needs, going beyond monitoring the quality of network connection.

Chapter "Decomposition Scheme for Flow Design in Photonic Data transport Networks" proposes a mathematical model of network design problem in the context of modern photonic network with wavelength division multiplexing [20]. This answers the needs of modelling photonic data transport networks which is crucial to make the efficient use of this technology possible. To reduce the complexity of the proposed model, the authors applied the Dantzig-Wolfe based decomposition, which is not unusual also in modelling of other types of network systems such as power grid networks [21]. This resulted in significant reduction in the number of constraints and variables used. In a wider context, this chapter confirms the need for in-depth analysis and modelling of modern network technologies and architectures enabled by them. Equally importantly, this clearly shows the challenges in the mathematical domain, which are created by the complexity of modern network systems.

Part II: Network Security

Wireless sensor networks provide basis for a growing number of applications. Such networks contribute also to the development of Internet of Things. Due to mobility of such networks, relatively low operating costs and the fact that these networks create an independent infrastructure communication, they can be successfully used in military applications [22, 23]. In military applications an extremely important problem is the authentication of the network nodes, which is more important than the problem of energy consumption by network nodes. The proposal to build such a network is presented in Chap. "Secure Transmission in Wireless Sensors' Domain Supported by the TPM". The proposed method of the transmission protection between nodes and the way of protecting the node resource, provide for utilization of the Trusted Platform Module (TPM) [24]. For this purpose, sensors create the domain in which one of them is the security authority of the domain. This node is also the recipient of the data from the sensors belonging to the domain. The experimental results confirm the usefulness of the solution. Experiments allow to pre-assess the delay in the transfer of data that is entered by cryptography procedures and the growth of energy demand during these procedures.

Chapter "Constraint Analysis for Security Policy Partitioning Over Tactical Service Oriented Architectures" is devoted to issues related to the partitioning of security policies in tactical service-oriented architecture [25, 26]. The proposed approach is based on the ontologically defined security infrastructure with the use of the Web Ontology Language (OWL) [27], and identification of the involved elements of tactical networks with critical impact. In the considerations were taken into account three categories of governing parameters, regarding to the attainment of the required security policy distribution. The first category refers to the evaluation of the policy from the point of view of the overall and local complexity. The second category refers to the evaluation and categorization of the deployed tactical nodes, based on their expected functional and operational specialization. The last category refers to the sufficient integration of dynamism, emerging from the characteristics of the tactical environment. The result of the work is the mechanism of adjusting the identified parameters for the optimal partitioning and distribution of security policies within the mission preparation stage.

Usually the decision-making procedure connected with access control uses the policy statements established by many principals. For trust management you can use Role-based Trust Management Language (RTML) [28, 29]. Such language allows to operate with security policies, authentication data and relationship in distributed and large scale access control systems. The extension of the Role-based Trust Management Language in the range of the determination of order, and time validity is described in Chap. "Practical Extensions of Trust Management Credentials" [30]. Presented solution proposes to add some extensions (including time data) to credentials to make a trust management system more useful in practice.

One of the most efficient ways of data encryption is AES encryption. Methods based on this encryption can operate in different modes. These modes include: ECB (Electronic Code Book), CBC (Cipher Block Chaining), CFB (Cipher Feedback), CTR (Counter), and GCM (Galois/Counter Mode) [31, 32]. Chapter "Performance and Energy Consumption Analysis of AES in Wireless Sensor Networks" describes the Performance and Energy Consumption Analysis of AES for the needs of Wireless Sensor Networks [33]. In the studies a system ATmega128 RFA1 was used. During the experiments the software implementation and the hardware implementation of AES encryption in different modes was tested. The achieved results show that in all cases the hardware implementations was at least five times faster than the software implementations. Energy consumption for AES was also examined (only CTR mode tested). The results indicate that the average power consumption is almost constant and does not depend on the size of the encrypted message (in the range of from 16 to 128 bytes), but for hardware implementation the average power consumption is about 5 % higher. Average energy consumption in both cases increases by leaps and bounds with an increase in message length. It is worth noting that in the whole range of message length the average power consumption for the software implementation is approximately 6-times greater than for the hardware implementation.

The assessment and management of risk is an essential part of any utilized system. It is particularly important for Critical Infrastructures (CI), such as energy (electricity, oil, gas) or transportation (road transport, rail transport, air transport, inland waterways transport, ocean and short-sea shipping and ports). Chapter "Computer Support for Risk Management in Critical Infrastructures" shows the CIRAS Tool, which was created under the EU CIRAS project. In this solution the assessment and management of risk issues are computer aided [34]. The CIRAS Tool operations are based on three pillars: Risk Reduction Assessment (RRA), Cost-Benefit-Assessment (CBA), Qualitative Criteria Assessment (QCA). In the chapter, particular attention was paid to the use of the OSCAD software platform to perform the tasks of the RRA component. The experimentation tool, called OSCAD-Ciras, was developed. A case study for the railway CI collaborating with the electricity CI was planned and performed.

In Chap. "An Innovative Web Platform for Flood Risk Management" an innovative real-time information system for enhanced support to flood risk emergency in urban and nearby coastal areas was presented [35–37]. The solution includes a description of the platform, its architecture, all innovative aspects related to the User Interface (UI), product creation and choice of technologies. The tool can be used not only for risk management, but also for the coordination of situational awareness and civil protection interaction.

Part III: Network Applications

Chapter "A Distributed Active Vibration Control System Based on the Wireless Sensor Network for Automotive Applications" presents a new approach of an adaptive system for automotive applications based on the AVC—Active Vibration Control Systems concept [38, 39]. The authors assume that the porting of a centralized system in a distributed system can improve its effectiveness and present a Wireless Sensor Network for damping vibrations in automotive applications. Sensors with a piezoelectric element (Series-SSHI method) [40] were used to measure and damp the vibrations and harvest energy from vibrations. The Finite Element Simulations (FEM) using COMSOL 5.1 software were provided to simulate deformations of the mechanical system and were then compared with the measured results.

Chapter "Improvements of Video Delay in OTT Live TV Service" studies the end-to-end delay observed by users of the Over The Top (OTT) Live TV services using Adaptive Bit Rate (ABR) technology [41]. The analysis and measurements in a test environment demonstrate the extent to which the main architecture elements—encoder, packager, Content Delivery Network (CDN) and player (e.g. GPAC [42] or [43]) contribute to this overall delay. The work presented in this chapter has been carried out as part of the EUREKA/CELTIC research project NOTTS (Next-Generation Over-The-Top Services) [44].

Chapter "Power Aware MOM for Telemetry-Oriented Applications—Levee Monitoring Use Case" addresses the issue of the Message-Oriented Middleware [45] utilization in telemetry systems. The authors provide a survey and practical measurements of common data transmission protocols for telemetry applications and wireless sensing. Based on the survey the authors propose concepts of message aggregation mechanisms to improve power consumption of the data transmission channel. As an entry point, the authors assume the utilization of the MQTT protocol [46]. The results of the research have been successfully implemented in a smart levee monitoring system.

Chapter "Monitoring Drip Irrigation System Using Wireless Sensor Networks" presents a model of architecture for a drip irrigation system using the Wireless Sensor and Actuators Networks (WSANs). The investigated model includes the soil moisture, temperature and pressure sensors to monitor the irrigation operations [47, 48]. The researchers have performed extensive simulations with the use of TOSSIM simulators [49]. The results show that the presented solution allows for better performance in terms of the delay, PDR for the priority traffic.

Chapter "BARBEI: A New Adaptive Battery Aware and Reliable Beacon Enabled Technique for IEEE802.15.4 MAC" focuses on the IEEE 802.15.4 standard [50] which supports both physical and Media Access Control (MAC) layers of low rate Wireless Sensor Networks (WSNs). The authors propose a technique that improves the performance of the IEEE802.15.4 standard by allowing its MAC to exploit the nonlinear processes of the battery [51] to prolong the WSN lifetime. The performance of the new algorithm has been examined and compared against that of the legacy IEEE 802.15.4 MAC algorithm through extensive simulation experiments.

Chapter "A Multisource Energy Harvesting Platform for Wireless Methane Sensor" focuses on harvesting energy [52, 53] from ambient sources in order to extend the operation time of wireless sensor networks. In this article, the authors present a multi-source harvesting circuit consisting of wind and solar energy and its implementation for the Wireless Gas Sensor Node (WGSN) [54]. The researchers demonstrate that a catalytic gas sensor can operate for two days without batteries by using the developed scheme.

3 Concluding Remarks

Network systems play vital roles in all of our daily life. This volume provides comprehensive selection of cutting edge state-of-the-art algorithms, technologies, and applications, providing new insights into a range of fundamentally important topics in network architectures, network security, and network applications. The significance of this book comes from the demand for better understanding of the network models. Due to the rapid growth in network architectures, security, and applications, further contributions and research findings are anticipated in the future.

References

1. King, J., Awad, A.I.: A distributed security mechanism for resource-constrained IoT devices. Informatica (Slovenia) **40**(1), 133–143 (2016)
2. Jain, A.K., Ross, A.A., Nandakumar, K.: Introduction to Biometrics. Springer (2011)
3. Awad, A.I., Baba, K.: Evaluation of a fingerprint identification algorithm with SIFT features. In: Proceedings of the 3rd 2012 IIAI International Conference on Advanced Applied Informatics, pp. 129–132. IEEE, Fukuoka, Japan (2012)
4. Egawa, S., Awad, A.I., Baba, K.: Evaluation of acceleration algorithm for biometric identification. In: Benlamri, R. (ed.) Networked Digital Technologies, Communications in Computer and Information Science, vol. 294, pp. 231–242. Springer, Berlin (2012)
5. Awad, A.I., Hassanien, A.E.: Impact of some biometric modalities on forensic science. In: Muda, A.K., Choo, Y.H., Abraham, A.N., Srihari, S. (eds.) Computational Intelligence in Digital Forensics: Forensic Investigation and Applications, Studies in Computational Intelligence, vol. 555, pp. 47–62. Springer International Publishing (2014)
6. Rubio-Loyola, J., Sala, D., Ali, A.I.: Maximizing packet loss monitoring accuracy for reliable trace collections. In: 16th IEEE Workshop on Local and Metropolitan Area Networks, LANMAN 2008, pp. 61–66. IEEE (2008)
7. Rubio-Loyola, J., Sala, D., Ali, A.I.: Accurate real-time monitoring of bottlenecks and performance of packet trace collection. In: 33rd IEEE Conference on Local Computer Networks, LCN 2008, pp. 884–891. IEEE (2008)
8. Chen, J., Kanj, I.A., Wang, G.: Hypercube network fault tolerance: a probabilistic approach. In: Proceedings of International Conference on Parallel Processing, pp. 65–72 (2002)
9. Ishikawa, T.: Hypercube multiprocessors with bus connections for improving communication performance. IEEE Trans. Comput. **44**(11), 1338–1344 (1995)
10. Chmielewski, J.: Device-independent architecture for ubiquitous applications. Pers. Ubiquit. Comput. **18**(2), 481–488 (2013)
11. Kobayashi, N., Tokunaga, E., Kimura, H., Hirakawa, Y., Ayabe, M., Nakajima, T.: An input widget framework for multi-modal and multi-device environments. In: Third IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS'05), pp. 63–70 (2005)
12. Moreira, R.S., Torres, J., Sobral, P., Morla, R., Rouncefield, M., Blair, G.S.: Dynamic adaptation of personal ubicomp environments. Pers. Ubiquit. Comput. **20**(2), 165–166 (2016)
13. Shah, P.A., Rehan, M., Chughtai, H.M.O., Qayyum, A.: On reducing throughput degradation of TCP connection after vertical handover. In: IEEE 13th International Multitopic Conference, INMIC 2009, pp. 1–4 (2009)
14. Gkatzikis, L., Tryfonopoulos, T., Koutsopoulos, I.: An efficient probing mechanism for next generation mobile broadband systems. In: 2012 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1191–1195 (2012)

15. do Carmo, R., Hollick, M.: Analyzing active probing for practical intrusion detection in wireless multihop networks. In: 11th Annual Conference on Wireless On-demand Network Systems and Services (WONS), pp. 77–80 (2014)
16. Tyrrell, A., Auer, G., Bettstetter, C.: Fireflies as role models for synchronization in Ad Hoc networks. In: Proceedings of the 1st International Conference on Bio Inspired Models of Network, Information and Computing Systems. BIONETICS '06, ACM, New York, NY, USA (2006)
17. Iyengar, S.S., Parameshwaran, N., Phoha, V.V., Balakrishnan, N., Okoye, C.D.: Algorithms for Wireless Sensor Networks, pp. 131–154. Wiley-IEEE Press (2011)
18. Hall, D.A.: Conventional and radio frequency identification (RFID) tags. In: Cadrin, S.X., Kerr, L.A., Mariani, S. (eds.) Stock Identification Methods, pp. 365–395, 2nd edn. Academic Press, San Diego (2014)
19. Hautcoeur, J., Talbi, L., Nedil, M.: High gain RFID tag antenna for the underground localization applications at 915 MHz band. In: 2013 IEEE Antennas and Propagation Society International Symposium (APSURSI), pp. 1488–1489 (2013)
20. Vinolee, R., Bhaskar, V.: Performance analysis of mixed integer linear programming with wavelength division multiplexing. In: 2014 2nd International Conference on Devices, Circuits and Systems (ICDCS), pp. 1–6 (2014)
21. Altay, C., Deli, H.: Distributed energy management of microgrids with Dantzig-Wolfe decomposition. In: IEEE PES Innovative Smart Grid Technologies, Europe, pp. 1–5 (2014)
22. Chudzikiewicz, J., Furtak, J., Zielinski, Z.: Secure protocol for wireless communication within internet of military things. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), pp. 508–513. IEEE (2015)
23. Hennebert, C., Dos Santos, J.: Security protocols and privacy issues into 6LoWPAN stack: a synthesis. IEEE Internet Things J. **1**(5), 384–398 (2014)
24. Furtak, J., Chudzikiewicz, J.: Securing transmissions between nodes of WSN using TPM. In: 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), Lodz, Poland, Sept 13–16, pp. 1059–1068. IEEE (2015)
25. Johnsen, F.T., Bloebaum, T.H., Schenkels, L., Fiske, R., Van Selm, M., de Sortis, V., van der Zanden, A., Sliwa, J., Caban, P.: SOA over disadvantaged grids experiment and demonstrator. In: 2012 Military Communications and Information Systems Conference (MCC), pp. 1–8. IEEE (2012)
26. Maule, R.W., Lewis, W.C.: Security for distributed SOA at the tactical edge. In: 2010 Military Communications Conference, (MILCOM 2010), pp. 13–18. IEEE (2010)
27. Souag, A., Salinesi, C., Comyn-Wattiau, I.: Ontologies for security requirements: a literature survey and classification. In: Advanced Information Systems Engineering Workshops, pp. 61–69. Springer (2012)
28. Gorla, D., Hennessy, M., Sassone, V.: Inferring dynamic credentials for role-based trust management. In: Proceedings of the 8th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming, pp. 213–224. ACM (2006)
29. Li, N., Winsborough, W.H., Mitchell, J.C.: Distributed credential chain discovery in trust management. J. Comput. Secur. **11**(1), 35–86 (2003)
30. Felkner, A., Kozakiewicz, A.: More practical application of trust management credentials. In: 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), Lodz, Poland, Sept 13–16, pp. 1125–1134. IEEE (2015)
31. Zhang, F., Dojen, R., Coffey, T.: Comparative performance and energy consumption analysis of different AES implementations on a wireless sensor network node. Int. J. Sens. Netw. **10**(4), 192–201 (2011)
32. Lee, J., Kapitanova, K., Son, S.H.: The price of security in wireless sensor networks. Comput. Netw. **54**(17), 2967–2978 (2010)
33. Panait, C., Dragomir, D.: Measuring the performance and energy consumption of AES in wireless sensor networks. In: 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), Lodz, Poland, Sept 13–16, pp. 1261–1266. IEEE (2015)
34. Bialas, A.: Experimentation tool for critical infrastructures risk management. In: 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), Lodz, Poland, Sept 13–16, pp. 1099–1106. IEEE (2015)

35. Gomes, J.L., Jesus, G., Rogeiro, J., Oliveira, A., Tavares da Costa, R., Fortunato, A.B.: Molines-towards a responsive web platform for flood forecasting and risk mitigation. In: 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), Lodz, Poland, Sept 13–16, pp. 1171–1176. IEEE (2015)
36. Oliveira, A., Jesus, G., Gomes, J., Rogeiro, J., Azevedo, A., Rodrigues, M., Fortunato, A., Dias, J., Tomas, L., Vaz, L., Oliveira, E., Alves, F., den Boer, S.: An interactive WebGIS observatory platform for enhanced support of integrated coastal management. J. Coastal Res. **70**, 507–512 (2014), Special Issue 70— Proceedings of the 13th International Coastal Symposium
37. Deng, Z., Namwamba, F., Zhang, Z.: Development of decision support system for managing and using recreational beaches. J. Hydroinformatics **16**(2), 447–457 (2014)
38. Elliott, S.: A review of active noise and vibration control in road vehicles. Tech. Rep. 981, University of Southampton. http://eprints.soton.ac.uk/65371/ (2008)
39. Svaricek, F., Fueger, T., Karkosch, H.J., Marienfeld, P., Bohn, C.: Automotive Applications of Active Vibration Control. INTECH Engineering—Control Engineering, INTECH (2010)
40. Lefeuvre, E., Badel, A., Richard, C., Petit, L., Guyomar, D.: A comparison between several vibration-powered piezoelectric generators for standalone systems. Sens. Actuators A: Phys. **126**(2), 405–416 (2006)
41. Stockhammer, T.: Dynamic adaptive streaming over HTTP–standards and design principles. In: Proceedings of the Second Annual ACM Conference on Multimedia Systems, pp. 133–144. MMSys'11, ACM, New York, NY, USA (2011)
42. GPAC: GPAC, multimedia player with MPEG-DASH support. https://gpac.wp.mines-telecom.fr/player/ (2015). Last access 27.4.2016
43. DASH-IF: DASH-IF, a reference mpeg-dash client. http://dashif.org/reference/players/javascript/1.4.0/samples/dash-if-reference-player/ (2015). Last access 27.4.2016
44. NOTTS: Eureka/celtic notts. http://projects.celticplus.eu/notts/ (2015). Last access 27.4.2016
45. Curry, E.: Message-Oriented Middleware, pp. 1–28. Wiley (2005)
46. MQTT: Mq telemetry transport (MQTT) documentation. http://mqtt.org/documentation (2015). Last access 30.11.2015
47. Gutiérrez, J., Villa-Medina, J.F., Nieto-Garibay, A., Porta-Gándara, M.A.: Automated irrigation system using a wireless sensor network and GPRS module. IEEE Trans. Instrum. Meas. **63**(1), 166–176 (2014)
48. Mafuta, M., Zennaro, M., Bagula, A., Ault, G., Gombachika, H., Chadza, T.: Successful deployment of a wireless sensor network for precision agriculture in Malawi. In: IEEE International Conference on Networked Embedded Systems for Enterprise Applications, pp. 1–7. IEEE Computer Society, Los Alamitos, CA, USA (2012)
49. TOSSIM: Tossim simulator. http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM (2015). Last access 27.4.2016
50. Ergen, S.C.: Zigbee/ieee 802.15.4 summary. http://home.iitj.ac.in/~ramana/zigbee.pdf (2004). Last access 27.4.2016
51. Jongerden, M.R., Haverkort, B.R.H.M.: Battery modeling. Technical Report TR-CTIT-08-01, Centre for Telematics and Information Technology University of Twente, Enschede (Jan 2008)
52. Vullers, R., van Schaijk, R., Doms, I., Hoof, C.V., Mertens, R.: Micropower energy harvesting. Solid-State Electron. **53**(7), 684–693 (2009), Papers Selected from the 38th European Solid-State Device Research Conference—ESSDERC'08
53. Akbari, S.: Energy harvesting for wireless sensor networks review. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M. (eds.) Proceedings of the 2014 Federated Conference on Computer Science and Information Systems (FedCSIS), Warsaw, Poland, Sept 7–10, pp. 987–992 (2014)
54. Somov, A., Baranov, A., Spirjakin, D., Passerone, R.: Circuit design and power consumption analysis of wireless gas sensor nodes: one-sensor versus two-sensor approach. IEEE Sens. J. **14**(6), 2056–2063 (2014)

Part I
Network Architectures

An Analytical Method of Server Placement in Regular Networks and Its Evaluation by Simulation Experiments

Jan Chudzikiewicz, Tomasz Malinowski and Zbigniew Zieliński

Abstract In the paper, the problem of determining the optimal server placement in the hypercube network structure and its influence on the values of some performance metrics is investigated. The method for the optimal server placement is proposed. It consists of two phases: in the first one the server placement is determined and in the second phase the appropriate communication structure is generated. The usefulness of the method has been verified through simulation experiments prepared and performed in Riverbed Modeler environment. Some results of these simulation tests for exemplary structures along with degradation of the 4-dimensional hypercube network are presented.

Keywords Server placement • Hypercube network • Reconfiguration • Fault-tolerant system

1 Introduction

This work is an extended version of the paper *The method of server placement in the hypercube networks* [1]. In order to guarantee the quality of service and performance of the specialized system with critical application, it is essential for the system to be able to detect faults, and to perform something akin to healing and recovering from events that might cause faults or misbehavior nodes in the network. Some of critical applications are used in near real-time mode and required both very high reliability and high efficiency of data processing throughout all the network life cycle. Furthermore, assuming harsh or even hostile physical environment for

J. Chudzikiewicz (✉) · T. Malinowski · Z. Zieliński
Military University of Technology, Warsaw, Poland
e-mail: jan.chudzikiewicz@wat.edu.pl

T. Malinowski
e-mail: tomasz.malinowski@wat.edu.pl

Z. Zieliński
e-mail: zbigniew.zielinski@wat.edu.pl

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_2

the system a number of factors including self-diagnosing, fault tolerance and reconfiguration should be seriously considered in the designing phase. One of the possible ways to achieve fault tolerance is reconfiguring the network to a smaller sized system after faults diagnosing, i.e. realization of a soft degradation strategy. New (degraded) network continues work after resources reassigning and under the condition that it meets special requirements. In turn, the efficiency of the system will depend heavily on the availability of resources (data bases, files or web data), which is determined by their placement in the network. So, for this kind of networks there is necessity for applying effective methods of resources placement.

In the paper, we focus upon one of the problems of reconfiguration in the regular networks with soft degradation: how to reconfigure application servers and allocate resources in the network to reduce the overall cost of delivering of services to the number of clients? It is known that this cost varies depending on the physical server placement in the network, type of server and delivered content to the clients. In general, there are static content (such as images, videos, text) or dynamic content. For the static type of content, the best server placement could be the one which minimize (overall or in average) the distance to the client. Intuitively this cost could be measured by the average hops. Network hops could be defined as the number of routers present in the path between client and server. Although typical network services are data base services or web application services some others could also be network-critical applications. One of exemplary services which might be regarded as the critical application is VoIP communication (Voice over IP) through a network. In the paper the influence of the network communication structure and its characteristics for values of different type application performance parameters is also investigated.

The computer networks with a regular structure as torus or hypercube [2–5] could be used in many kinds of specialized critical application (for instance military, aerospace or medical systems). An interconnection network with the hypercube logical structure is a well-known interconnection model for multiprocessor systems [6, 7] and still hypercube networks are the field of interest of many theoretical studies concerning (among others) resource placement problem, which has been intensively studied in [8–13].

In order to achieve high reliability of the system the network could be considered as soft degradable computer network [13–15]. In this kind of networks a processor identified as faulty is not repaired (or replaced) but access to it is blocked. In the work [13] an analysis of the different schemas of resources placement in the 4-dimensional hypercube network with soft degradation was conducted.

Designing and exploitation of special networks in critical application is a comprehensive task that requires addressing a number of theoretical and practical problems. One of the problems is a skillful resources deployment in the network and modification of resources deployment after each phase of the network degradation. One of considered in the literature the resource placement problem is a combination the distance-d and the m adjacency problems, where a non-resource node must be a distance of at most d from m processors nodes [8–11, 13]. In [11] a perfect deployment has been introduced and analyzed which is defined as the minimum number of resources processors which are accessible by working processors. The definition—perfect deployment is a characteristic of the value of the generalized cost of information traffic in the network at a given load of tasks. In [13] the notion of (*m*, *d*)-perfect resources placement in the hypercube type structure *G* has been extended to the such allocation of k resources which minimizes the average distances between the working processors and resource processors in the structure *G*.

We thoroughly investigate the case when a specialized computer system is based on the 4-dimensional hypercube skeleton network with communication nodes which could communicate between themselves via cable connections. The main task of the hypercube network is to provide efficient access to resources managed by the server (or cluster of servers) connected directly to one of the network nodes and semi-stationary clients communicating with the assigned network nodes via wireless links. The execution of applications by a client processor requires an access to server services and resources, also some results returned by the server must be submitted to other clients. We assume that all clients are responsible for performing the same or very similar tasks. Thus all clients will generate similar workload of the network. The problem which arises for the given network structure is to determine the most effective server placement in the network structure.

The main goal of this paper is to give an effective method of solving the server placement problem for the hypercube network along with its soft degradation process.

A generalized cost of a network traffic with a specified resources deployment and workload of a network is usually tested through experimental measurements or examined with the use of simulation methods. In the paper we apply a two phased approach. In the first stage we solve the problem of a server placement in the given network structure on the base of analytically determined attainability measure, which was proposed in [13]. It should be noticed that real cost of information traffic in a network for a given deployment of the server with resources depends on the nature of the tasks performed by clients in the network. In the second stage we have examined this problem with the use of simulation methods for the specified server deployment determined by the simple analytical method and given type of task load of the network.

We see our contributions as follows. Firstly, we have extended the approach proposed in [1, 13] to the determining server placement in the hypercube network with soft degradation on the base of nodes attainability calculation. Secondly, we propose the algorithm of the communication structure assignation with the use of dendrite calculation. Next, we show the feasibility of this approach by applying obtained results to some possible structures of degraded 4-dimensional hypercube network and verifying effectiveness of server placement by simulation experiments.

The rest of the paper is organized as follows. In Sect. 2, a basic definitions and properties were introduced. The calculation of radius and attainability for exemplary structures were presented. In Sect. 3, the proposal of the algorithm determining server placement was presented. An illustration of the main algorithm steps for the exemplary structure was given. In Sect. 4, the results of simulation tests for verification the algorithm (implemented in Riverbed Modeler environment) were described. In Sect. 5, some concluding remarks were presented.

2 The Basic Definitions and Assumptions

Definition 1 The logical structure of processors network we call the structure of n-dimensional cube if is described by coherent ordinary graph $G = E, U$ (E—set of computer, U—set of bidirectional data transmission links), which nodes can be described (without repetitions) by n-dimensional binary vectors (labels) in such a way that

$$\left[\delta\left(\varepsilon\left(e^{'}\right), \varepsilon\left(e^{''}\right)\right) = 1\right] \Leftrightarrow \left[\left(e^{'}, e^{''}\right) \in U\right] \tag{1}$$

where $\delta\left(\varepsilon\left(e^{'}\right), \varepsilon\left(e^{''}\right)\right)$ is Hamming distance between the labels of nodes $e^{'}$ and $e^{''}$. The Hamming distance between two binary vectors $\varepsilon\left(e^{'}\right)$ and $\varepsilon\left(e^{''}\right)$. complies with the dependency:

$$\delta\left(\varepsilon\left(e^{'}\right), \varepsilon\left(e^{''}\right)\right) = \sum_{k \in \{1, \ldots, n\}} \left(\varepsilon\left(e^{'}\right)_k \oplus \varepsilon\left(e^{''}\right)_k\right)$$

where:

- $\varepsilon\left(e^{'}\right)_k$—the k-th element of the binary vector $\varepsilon\left(e^{'}\right)$,
- $\oplus$—modulo 2 sum.

We investigate the case when skeleton of the network has the logical structure of 4-dimensional hypercube (Fig. 1).

A topology of the hypercube may be represented by an ordinary consistent graph whose nodes are described by 4-dimensional binary vectors such that the Hamming distance between vectors (labels) of the adjacent nodes equals one. If $|E| = 2^4$ and $|U| = 2|E|$, then such graph we called (non labeled) 4-dimensional cube and will be denote by H^4. Thus H^4 is a regular graph of degree of 4 i.e. such that the degree of a node $e \in E$ we determine as $\mu(e) = |E(e)|$, where $E(e)$ is a set of nodes adjacent to the node $e \in E$ and $\mu(e) = 4$ for each node e of the graph H^4.

Let $d\left(e, e^{'}|G\right)$ be the distance between nodes e and $e^{'}$ in a coherent graph G, that is the length of the shortest chain (in the graph G) connecting node e with the node $e^{'}$.

Let $r(e|G) = \underset{e^{'} \in E(G)}{max} d\left(\left(e, e^{'}\right)|G\right)$ be the greatest distance from the node $e \in E(G)$ to another node of the set $E(G)$, and $r(G)$, and $D(G)$ (respectively) denote the radius and the diameter of a graph G i.e. $r(G) = min\{r(e|G) : e \in E(G)\}$ and $D(G) = max\left\{d\left(e^{'}, e^{''}|G\right) : \left\{e^{'}, e^{''}\right\} \subset E(G)\right\}$.

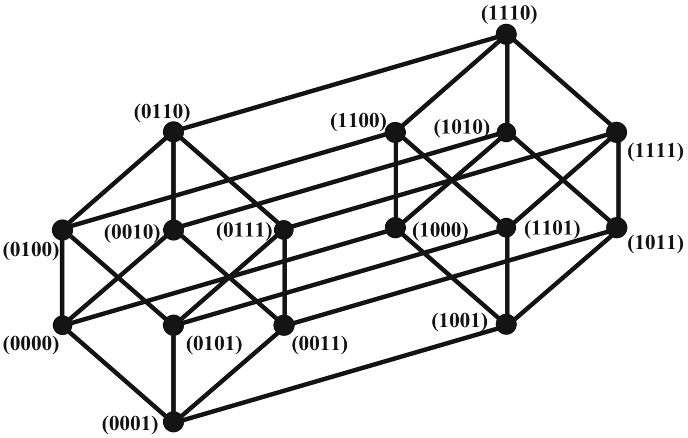


Fig. 1 4-dimensional hypercube with labeled nodes

Property 1 *For the 4-dimensional cube H^4 the equation is complied*

$$D\left(H^4\right) = r\left(H^4\right) = 4.$$

It is known that $D(G) \leq 2r(G)$.

If $r(e|G) = r(G)$ then the node e is called the central node of the network G.

Denote by $E^{(d)}(e|G) = \left\{e^{'} \in E(G) \colon d\left(e, e^{'}|G\right) = d\right\}$ for $d \in \{1, \ldots, D(G)\}$, and by

$$\varsigma(e|G) = \left(\varsigma_1(e|G), \ldots, \varsigma_{r(e|G)}(e|G)\right) \text{ for}$$

$$\varsigma_d(e|G) = \left|E^{(d)}(e|G)\right| \tag{2}$$

Definition 2 Let $\varphi(e|G) = \sum_{e^{'} \in E(G)} d\left(e, e^{'}|G\right)(e \in E(G))$ be attainability of the computer e in the network G and $\Phi(G) = \sum_{e \in E(G)} \varphi(e|G)$ be attainability of the network G.

Using (2) we have

$$\varphi(e|G) = \sum_{d=1}^{r(e|G)} d\varsigma_d(e|G) \tag{3}$$

Property 2 $\Phi(H^4) = 512$ because $\forall_{e \in E(H^4)} \colon \left(r(e|H^4) = 4 \wedge \varsigma_d(e|H^4) = \begin{pmatrix} 4 \\ d \end{pmatrix}\right)$. Using (3) we have $\forall_{e \in E(H^4)} \colon \varphi(e|H^4) = 32$ and $\left|E(H^4)\right| = 2^4$, then $\Phi(H^4) = \left|E(H^4)\right| \varphi(e|H^4)$ [13].

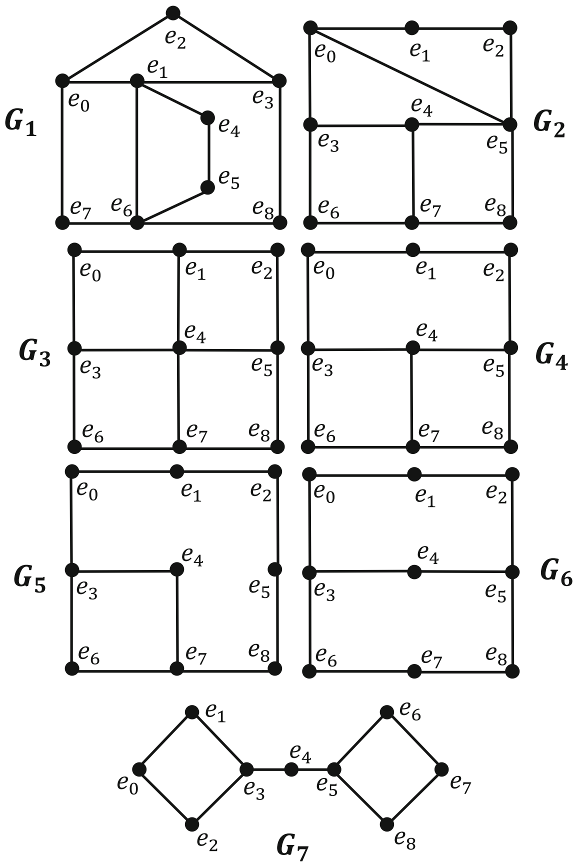


Fig. 2 Example of cyclic subgraphs of H^4 order 9 [14]

Example 1 Figure 2 presents all the seven possible cyclic structures upon the occurrence of $k = 7$ consecutive failures of processors of the network H^4 which are the subgraphs of H^4 [14].

It should be noticed, that for the given network structure G on the base of the obtained measures $r(e|G)$ it would be rational to choose the server placement at the central node of the network or in the node with the minimum value $r(e|G)$. In some cases (let's consider the structures G_2, G_4, G_5, G_6 see Table 1) we are not able to choose the best server placement. Then we can have determined $\varsigma(e|G)$ using (2) and $\varphi(e|G)$ using (3) for these structures. Table 2 shows the values of $\varsigma(e|G)$ and $\varphi(e|G)$ and $\Phi(G)$ for all structures presented in Fig. 2.

Definition 3 Let $T = E, U^*$ be the dendrite i.e. such coherent acyclic partial graph of G that:

$$\exists e', e'' \in U \Rightarrow e', e'' \in U^* \Leftrightarrow \left[\left(d\left(e_i, e'\right) \neq d\left(e_i, e''\right)\right) \wedge d\left(e', e''\right) = 1\right] \text{ for } r(e_i) = \underset{e \in E(G)}{min} r(e).$$

Table 1 The $r(e,G)$, $r(G)$, and $D(G)$ for the structures presented in the Fig. 2

$r(e\|G_i)$ $e \in E(G)$	$r(e\|G_1)$	$r(e\|G_2)$	$r(e\|G_3)$	$r(e\|G_4)$	$r(e\|G_5)$	$r(e\|G_6)$	$r(e\|G_7)$
e_0	3	3	4	4	4	4	6
e_1	2	4	3	4	4	4	5
e_2	4	4	4	4	4	4	5
e_3	3	3	3	3	4	3	4
e_4	3	3	2	3	4	3	3
e_5	4	3	3	3	4	3	4
e_6	3	4	4	4	4	4	5
e_7	3	4	3	4	4	4	6
e_8	3	3	4	4	4	4	5
$r(G)$	2	3	2	3	4	3	3
$D(G)$	4	4	4	4	4	4	6

The dendrite T is a communication structure of G. The algorithm for determined the dendrite T is presented in Sect. 3.

3 The Method of Optimal Server Placement and a Network Communication Structure Determining

The method consists of two phases. In the first phase, the node satisfying the equations $r(e_i) = \underset{e \in E(G)}{min} r(e)$ or $\varphi(e_i|G) = \underset{e \in E(G)}{min} \varphi(e|G)$ is chosen as the *server placement*. In the second phase, for chosen node the dendrite T (communication structure satisfying the condition $d_{max}(e_i|T) = r(e_i)$) is determined. Based on the presented method, the algorithm for determining the server placement and the communication structure was developed.

The algorithm for determining the server placement and communication structure.

Step 1. Determine $r(e|G)$ for $e \in E(G)$.

Step 2. Choose a node $e_i \in E(G)$ such that $r(e_i) = \underset{e \in E(G)}{min} r(e)$.

If $|\{e_i\}| > 1$ go to step 3 else go to step 5.

Step 3. Determine $\varphi(e|G)$ for $e \in E(G)$.

Step 4. Choose a node $e_i \in E(G)$ such that $\left(\varphi(e_i|G) = \underset{e \in E(G)}{min} \varphi(e|G)\right) \wedge \left(\mu(e_i) = \underset{e \in E(G)}{max} \mu(e)\right)$.

Selected node e_i will be a central node of dendrite.

Table 2 The $\varsigma(e|G)$, $\varphi(e|G)$ and $\varPhi(G)$ for the structures presented in the Fig. 2

	G_1					G_2				
$d(e,e'\|G)$ $e \in E(G)$	**1**	**2**	**3**	**4**	$\varphi(e,G_1)$	**1**	**2**	**3**	**4**	$\varphi(e,G_2)$
$\mathbf{e_0}$	3	3	2	0	15	3	4	1	0	14
$\mathbf{e_1}$	4	4	0	0	12	2	2	3	1	19
$\mathbf{e_2}$	2	3	2	1	18	2	3	2	1	18
$\mathbf{e_3}$	3	3	2	0	15	3	3	2	0	15
$\mathbf{e_4}$	2	3	3	0	17	3	4	1	0	14
$\mathbf{e_5}$	2	3	2	1	18	4	3	1	0	13
$\mathbf{e_6}$	4	3	1	0	13	2	3	1	1	18
$\mathbf{e_7}$	2	4	2	0	16	3	2	2	1	17
$\mathbf{e_8}$	2	4	2	0	16	2	4	2	0	16
			$\varPhi(G_1)$		140			$\varPhi(G_2)$		144

	G_3					G_4				
$d(e,e'\|G)$ $e \in E(G)$	**1**	**2**	**3**	**4**	$\varphi(e,G_3)$	**1**	**2**	**3**	**4**	$\varphi(e,G_4)$
$\mathbf{e_0}$	2	3	2	1	18	2	3	2	1	18
$\mathbf{e_1}$	3	3	2	0	15	2	2	3	1	19
$\mathbf{e_2}$	2	3	2	1	18	2	3	2	1	18
$\mathbf{e_3}$	3	3	2	0	15	3	3	2	0	15
$\mathbf{e_4}$	4	4	0	0	12	3	4	1	0	14
$\mathbf{e_5}$	3	3	2	0	15	3	3	2	0	15
$\mathbf{e_6}$	2	3	2	1	18	2	3	2	1	18
$\mathbf{e_7}$	3	3	2	0	15	3	2	2	1	17
$\mathbf{e_8}$	2	3	2	1	18	2	3	2	1	18
			$\varPhi(G_3)$		144			$\varPhi(G_4)$		152

	G_5					G_6				
$d(e,e'\|G)$ $e \in E(G)$	**1**	**2**	**3**	**4**	$\varphi(e,G_5)$	**1**	**2**	**3**	**4**	$\varphi(e,G_6)$
$\mathbf{e_0}$	2	3	2	1	18	2	3	2	1	18
$\mathbf{e_1}$	2	2	3	1	19	2	2	3	1	19
$\mathbf{e_2}$	2	2	2	2	20	2	3	2	1	18
$\mathbf{e_3}$	3	2	2	1	17	3	2	3	0	15
$\mathbf{e_4}$	2	3	2	1	18	2	4	2	0	16
$\mathbf{e_5}$	2	2	3	1	19	3	3	2	0	15
$\mathbf{e_6}$	2	3	2	1	18	2	3	2	1	18
$\mathbf{e_7}$	3	2	2	1	17	2	2	3	1	19
$\mathbf{e_8}$	2	3	2	1	18	2	3	2	1	18
		$\varPhi(G_5)$			164			$\varPhi(G_6)$		156

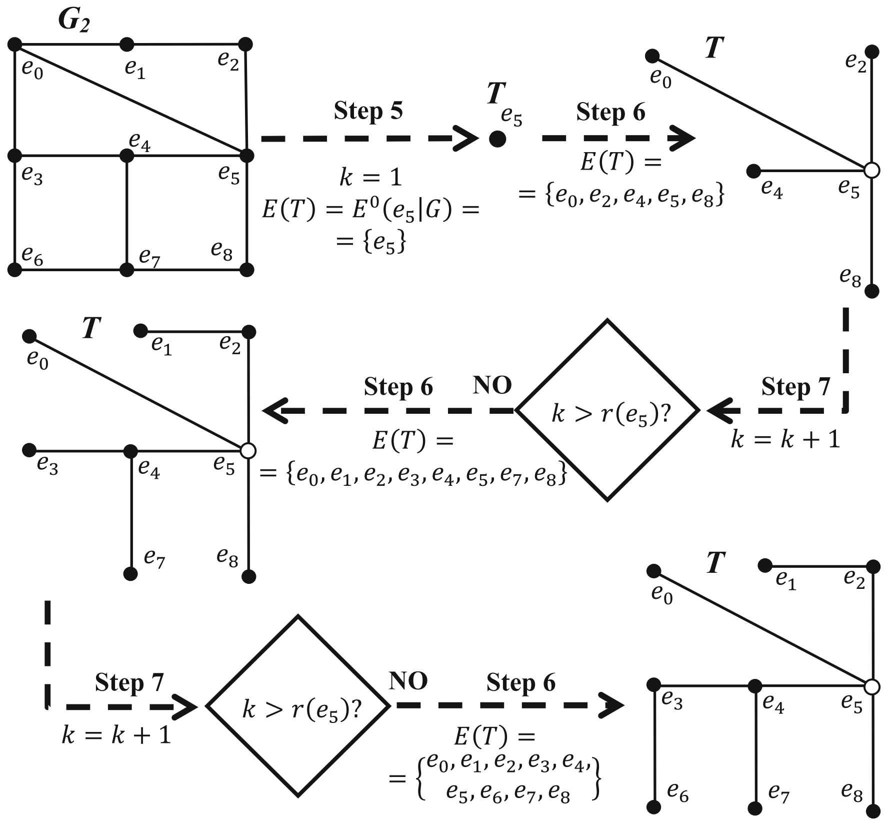
	G_7						
$d(e,e'\|G)$ $e \in E(G)$	**1**	**2**	**3**	**4**	**5**	**6**	$\varphi(e,G_7)$
$\mathbf{e_0}$	2	1	1	1	2	1	27
$\mathbf{e_1}$	2	2	1	2	1	0	22
$\mathbf{e_2}$	2	2	1	2	1	0	22
$\mathbf{e_3}$	3	2	2	1	0	0	17
$\mathbf{e_4}$	2	4	2	0	0	0	16
$\mathbf{e_5}$	3	2	2	1	0	0	17
$\mathbf{e_6}$	2	1	1	1	2	1	22
$\mathbf{e_7}$	2	1	1	1	2	1	27
$\mathbf{e_8}$	2	2	1	2	1	0	22
					$\varPhi(G_7)$		192

Step 5. (second stage)
Set $k = 1$ and $(E(T) = E^0(e_i|G) = \{e_i\})$.

Step 6. $E(T) = E(T) + E^k(e_i|G).U^* = U^* \bigcup\limits_{(e' \in E^{k-1}(e_i|G)) \wedge (e'' \in E^k(e_i|G))} e', e''.$

Step 7. $k = k + 1$.
Check if $k > r(e_i)$.
YES
Go to step 8.
NO
Return to step 6.

Step 8. The end of the algorithm.
Shows the algorithm, for example, determining the communication structure for G_2. In the first phase of the algorithm (steps 1–4), based on the attainability of G_2 nodes (Table 2 greyed row), node e_5 was appointed as the central node. The second phase of the algorithm is presented in the Fig. 3.



G_2
e_0 e_1 e_2 e_3 e_4 e_5 e_6 e_7 e_8
Step 5
$k = 1$
$E(T) = E^0(e_5|G) =$
$= \{e_5\}$
T
e_5
Step 6
$E(T) =$
$= \{e_0, e_2, e_4, e_5, e_8\}$
T
e_0 e_2 e_4 e_5 e_8
Step 7
$k = k + 1$
$k > r(e_5)?$
NO
Step 6
$E(T) =$
$= \{e_0, e_1, e_2, e_3, e_4, e_5, e_7, e_8\}$
T
e_0 e_1 e_2 e_3 e_4 e_5 e_7 e_8
Step 7
$k = k + 1$
$k > r(e_5)?$
NO
Step 6
$E(T) =$
$= \{e_0, e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$
T
e_0 e_1 e_2 e_3 e_4 e_5 e_6 e_7 e_8

Fig. 3 An illustration of the algorithm steps

Dendrite T determined in the second phase of the algorithm for the central node e_5, is one of the possible (but optimal) communication structure.

4 The Results of Simulation Studies

Procedure for determining the best location of resources in the hypercube network specified in the section III has been verified through simulation tests. The aim of the test was to confirm the correctness of the theoretical considerations and arguments.

Simulation studies have been prepared and implemented in Riverbed Modeler environment. Subgraph G_1 (Fig. 2) has been the subject of research.

Nodes have been modeled as routers and LAN segments attached to them. The cases when the server (with different typical and popular network services, chosen arbitrarily by authors) is connected to the selected node within T set of G_1 structure were examined. In the Fig. 4 different T set of substructures (communication structures) correspond to different simulation scenarios is shown. The circle indicates node which the server is connected to.

For example, the network topology of the single scenario with *T's* dendrite 1 was shown in the Fig. 5.

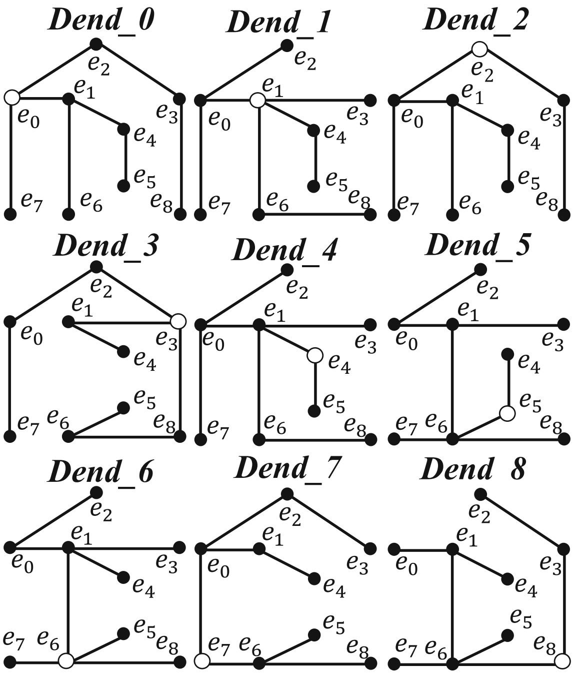


Fig. 4 The *T* of dendrites of G_1

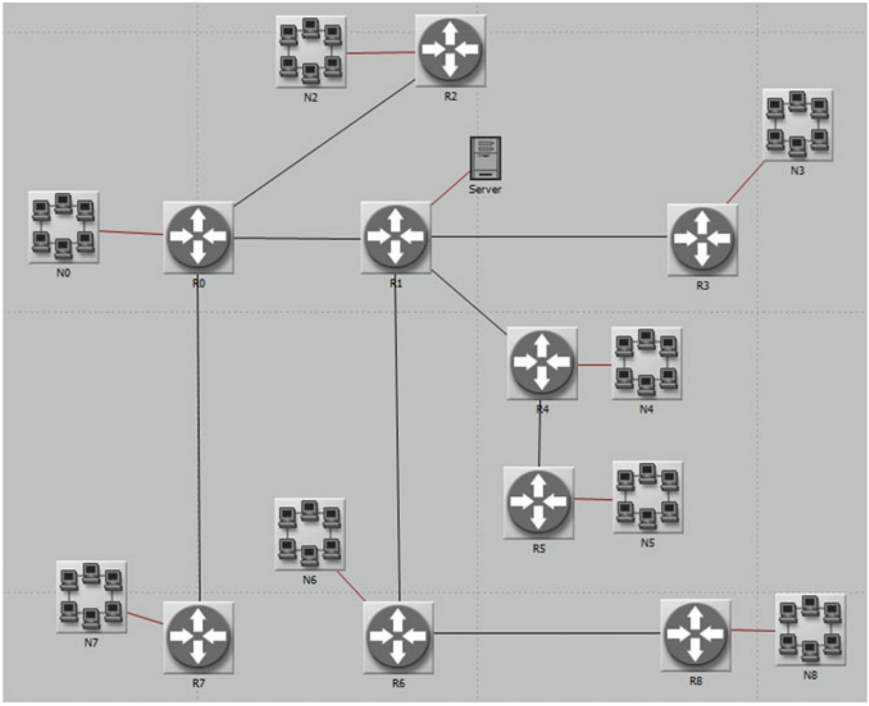


Fig. 5 Network topology of *Dend_1* scenario

The name of the simulation scenario (*Dend_1* in Fig. 5) was associated with the node's number, which the server was connected to (number of the dendrite's central node).

The server was acting as a database server, ftp server, web server, and the node with which it was possible to communicate through VoIP (Voice over IP). Workstations within LAN segments (ten workstations in each segment) were functioning as the server's clients. All network services have used the standard application models, available at Riverbed Modeler ("High Load" ftp and database models, "Heavy Browsing" http model and "PCM Quality Speech" voice model) [16].

The communication structure (skeleton of the network) was modeled as a set of routers connected via 1,5 Mb/s links.

Some interesting results, confirming the correctness of the procedure for determining the server placement and communication structure, are shown in the figures below. The dotted lines (e.g. Fig. 6) corresponds to the results obtained for the structure *Dend_1*, which is, according to the procedure, the most effective (the best) communication structure for subgraph G_1.

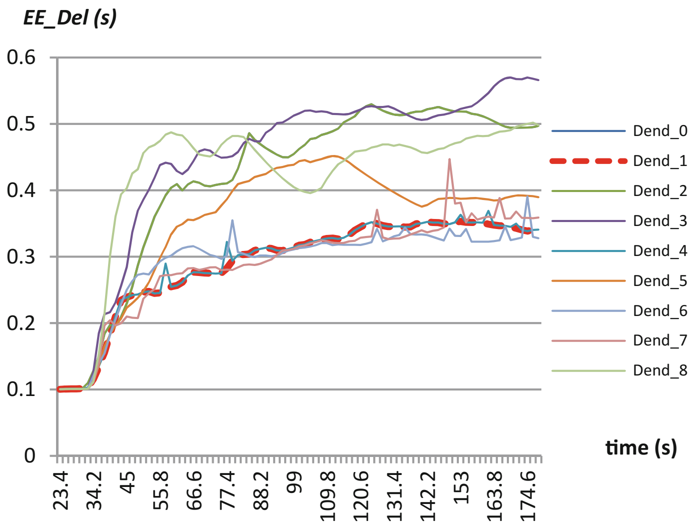


Fig. 6 *End-to-end* delay for VoIP transmission

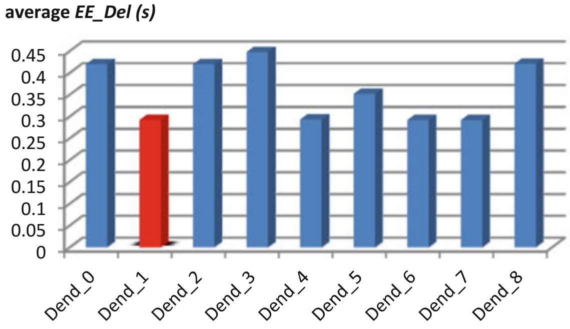


Fig. 7 Average *End-to-end* delay for *Dend_0* to *Dend_8* structures

For each simulation scenarios (*Dend_0* to *Dend_8*) five characteristics were determined.

A. *End-to-end delay (EE_Del)*

End-to-end delay is average delay in seconds for all LAN segments nodes communicating with the server through VoIP. The lowest value of *EE_Del* is desired (it is the best score).

Results obtained during the simulation are presented in the Fig. 6. The Fig. 7 shows the average values of *EE_Del*.

B. *TCP Delay (TCP_Del)*

TCP_del represents delay of TCP packets in seconds. This value is measured from the time an application data packet is sent from the source TCP layer to the time it is completely received by the TCP layer in the destination node. It is average delay in the complete network, for all connections. The lowest values are the best.

The results are presented in Figs. 8 and 9.

C. *Number of Hops (Nr_Hops)*

Nr_Hops represents an average number of IP hops taken by data packets reaching at a destination node.

We expected the lowest value for *Dend_1* structure and results are presented in the Fig. 10.

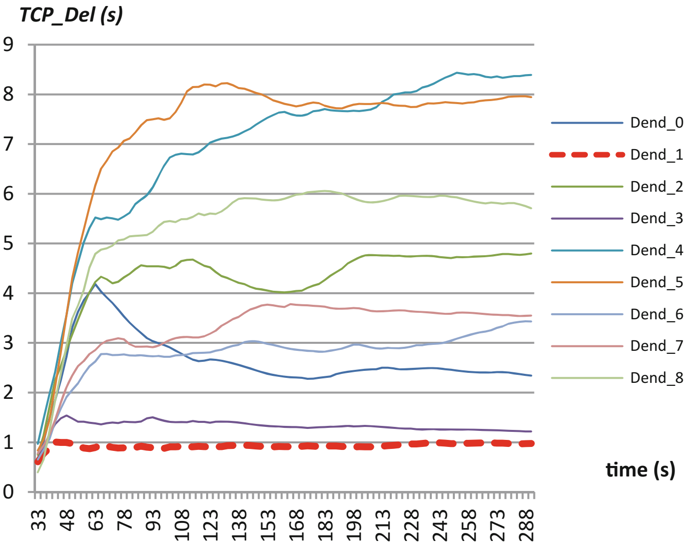
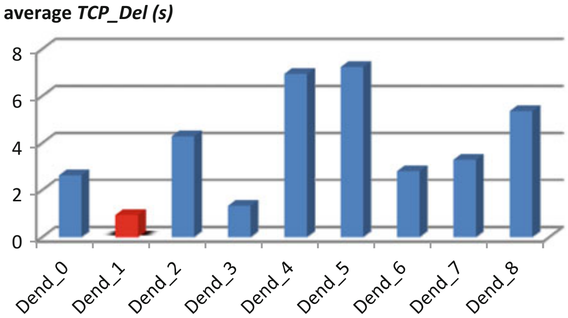


Fig. 8 *TCP delay* for TCP-based services

Fig. 9 Average *TCP delay* for TCP-based services



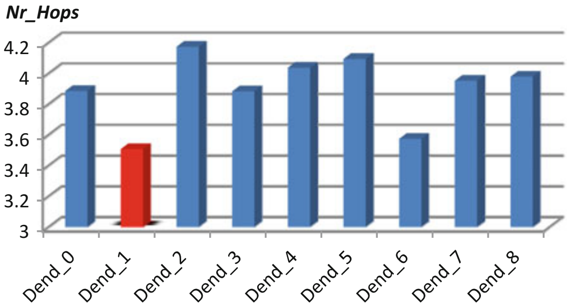


Fig. 10 Average *Number of Hops* for *Dend_0* to *Dend_8* structures

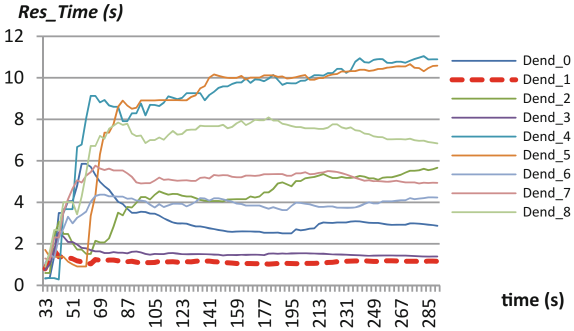


Fig. 11 *Response Time* for database service

D. *Response Time (Res_Time)*

Res_Time is average time elapsed between sending a request and receiving the response packet in seconds. It was measured for all the server's services (database, WWW and FTP).

The selected graph, *Response Time* for the database service was presented in the Fig. 11.

Average values of the database server's response time for each simulation scenario are shown in the Fig. 12.

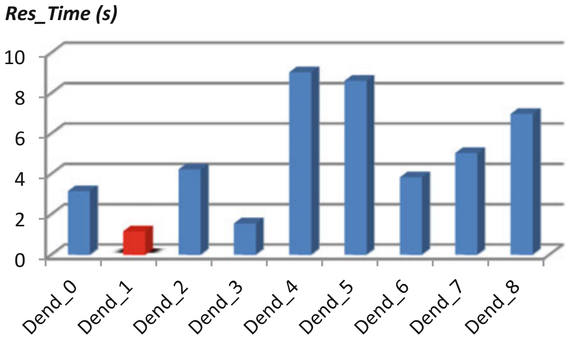


Fig. 12 Average *Response Time* for database service for all G_1's dendrites

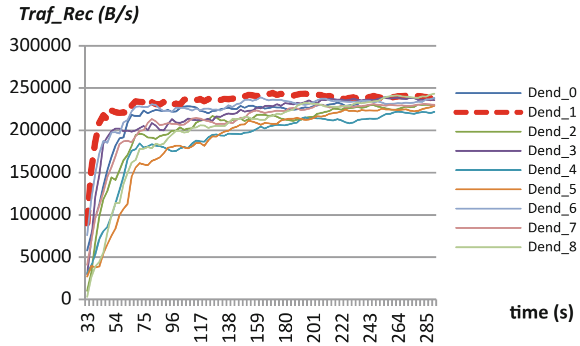


Fig. 13 *Traffic Received* for the FTP server

E. *Traffic Received (Traf_Rec)*

Traf_Rec is an average number of bytes per second forwarded to server's application by the transport layer in the complete network. It was measured for all the server's service and treated as a transmission speed indicator, so we expected highest values for the best communication structure (*Dend_0* in the drawings of the selected service—Figs. 13 and 14).

All the results are presented in Table 3. It should be noted that the best result was reported in most measurements for *Dend_1* structure (winning factor—78 %).

The results summarized in Table 3 indicate that the best structure is a *Dend_1*. Unfortunately, for other structures the results are not so obvious. Consider, for example, the results obtained for dendrites of graph G_2, illustrated in Fig. 15.

Previously used simple Winning factor indicates that the best communication structure is *Dend_7*, while the number of hops (*Nr_Hops*), our main indicator, points to *Dend_5* (best partial results for significant dendrites are highlighted in Table 4). Dendrites which are not winners in any subcategory have been removed from the Table 4.

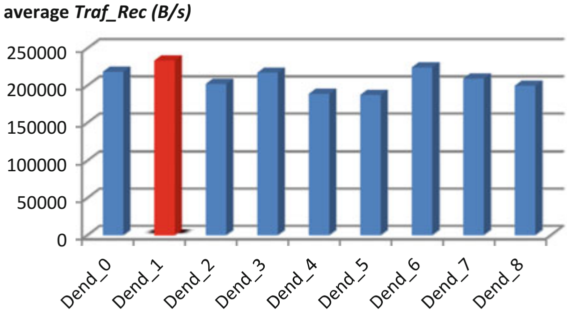


Fig. 14 Average FTP *Traffic Received* for all G_1's dendrites

Table 3 The simulation results for dendrites of G_1 structure

				HTTP		DB		FTP		
	Nr_Hops	***TCP_Del (s)***	***EE_Del (s)***	***Res_Time (s)***	***Traf_Rec (B/s)***	***Res_Time (s)***	***Traf_Rec (B/s)***	***Res_Time (s)***	***Traf_Rec (B/s)***	**Winning factor (%)**
Dend_0	3.74	2.59245	0.419	1.113	70013	3.129	15163	8.22	170518	
Dend_1	3.43	0.87554	0.291	0.641	136128	1.137	18214	5.37	188778	**78**
Dend_2	4.10	4.043734	0.419	1.364	34099	4.210	14121	14.25	153116	
Dend_3	3.75	1.307666	0.447	0.936	119991	1.532	18350	6.86	168044	**11**
Dend_4	3.87	6.015415	0.292	1.959	25875	9.018	12745	19.35	140109	
Dend_5	3.88	6.610183	0.351	1.968	28586	8.599	11137	19.67	134400	
Dend_6	3.46	2.499316	0.291	0.873	66342	3.818	15677	10.45	179894	
Dend_7	3.76	2.953067	0.291	1.397	43371	5.018	16540	11.25	161662	
Dend_8	3.88	5.09135	0.420	1.417	36789	6.954	14086	16.71	143929	

However, upon closer examination of the results, it can be shown that *Dend_5* may compete for the title of the best communication structure.

The authors propose to replace the percentage Winning factor with another factor, which is the sum of the points scored for a “place on the podium” (points for position in subcategories classification). In this case the rule is: “smaller factor wins”.

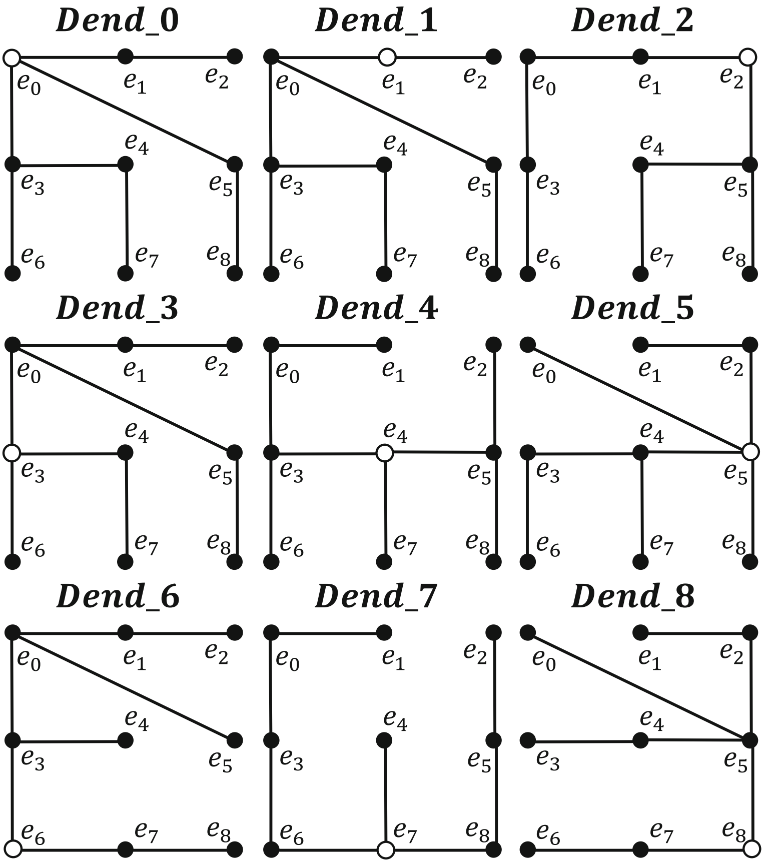


Fig. 15 Dendrites of G_2 structure

The results for this type of classification are shown in Table 5 and the figure corresponding to the results is Fig. 16.

The simulation results provide more complete information about the functioning of the network of a certain structure, unfortunately parameters such as bit rate, latency, response time, determined during simulation, are difficult to estimate and to use in analytical solving the task of choosing an optimal communication structure or server placement. However, the authors understand the need to take into account, in addition to the number of hops used here, other parameters characterizing the communication links and nodes.

Table 4 The simulation results for dendrites of G_2 structure

	Nr_Hops	*TCP_Del (s)*	*EE_Del (s)*	HTTP		DB		FTP		Winning factor (%)
				Res_Time (s)	*Traf_Rec (B/s)*	*Res_Time (s)*	*Traf_Rec (B/s)*	*Res_Time (s)*	*Traf_Rec (B/s)*	
Dend_0	3.95	2.27	0.43	1.35	74931	2.97	16074	8.20	209494	**22**
Dend_1	4.45	7.15	0.17	2.12	1086	9.14	4070	22.25	64659	**11**
Dend_5	3.82	2.97	0.34	1.12	73156	3.72	13664	10.13	221037	**22**
Dend_7	4.14	2.25	0.34	1.35	80051	3.49	12781	8.03	224473	**44**

Table 5 Classification results for dendrites of G_2 structure

	Points for subcategories									
	Nr_Hops	*TCP_Del (s)*	*EE_Del (s)*	HTTP		DB		FTP		Winning factor($\sum$) - Points in general classification
				Res_Time (s)	*Traf_Rec (B/s)*	*Res_Time (s)*	*Traf_Rec (B/s)*	*Res_Time (s)*	*Traf_Rec (B/s)*	
Dend_0	4	2	9	3	2	1	1	2	4	28
Dend_1	8	7	1	9	8	*8*	9	7	9	66
Dend_2	9	6	3	5	9	5	6	5	8	56
Dend_3	3	5	5	4	4	6	4	6	7	44
Dend_4	2	4	6	7	5	4	2	4	3	37
Dend_5	**1**	**3**	**4**	**1**	**3**	**3**	**3**	**3**	**2**	**23**
Dend_6	5	9	2	8	6	9	7	9	6	61
Dend_7	7	1	5	2	1	2	5	1	1	25
Dend_8	6	8	7	6	7	7	8	8	5	62

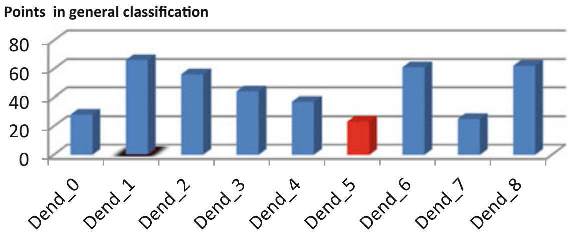


Fig. 16 Classification results for dendrites of G_2 structure, according to results in Table 5

5 Conclusion

How can be easily demonstrated, the algorithm for server placement determining has linear complexity. Correctness of developed algorithm and its usefulness for server placement and the optimal communication structure in the hypercube network with soft degradation on the base of nodes attainability calculation was confirmed by simulation tests. Thus, the goal of the work to find simple method for server placement determining based on the properties of logical network structure was achieved.

Simulation studies have confirmed the average hops can be treated as the basic performance criterion for choosing the best communication structure. It was turned out that in most cases average number of hops also allows for a minimum network latency, or the lower response time and minimum server loads.

In the case of more complex structures, it seems reasonable to take into account other parameters describing the communication links and nodes. The main measure of the network structure's quality should be more complex, which will be taken into account in subsequent works.

Further work will also address the problem of n—application servers deployment. In this case each server will acting as virtual machine with separated application services. An optimal virtual machine placement for minimizing the number of required hosting server (physical machines) is needed. The problem lays in determining such a virtual machine placement in a regular network so that minimum server application survive at any t failures of host server providing the number of hosting server should be minimized. It should be noted that this solution let us avoiding having to move the application server (VM) after each diagnosing (localization) of physical machine failure hosting virtual machines. Similar to the presented approach the results would be evaluated by simulation experiments.

References

1. Chudzikiewicz, J., Malinowski, T., Zieliński, Z.: The method for optimal server placement in the hypercube networks. In: Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, pp. 1175–1181, ACSIS, vol. 5, doi:10.15439/2015F157 (2015)

2. Hongwei, H., Wei, S., Youzhi, X., Hongke, Z.: A virtual hypercube routing algorithm for wireless healthcare networks. Chin. J. Electron. **19**(1), 138–144 (2010)
3. Chuang, P.J., Li, B.Y., Chao, T.H.: Hypercube-based data gathering in wireless sensor networks. J. Inf. Sci. Eng. **23**, 1155–1170 (2007)
4. Zieliński, Z., Chudzikiewicz, J., Arciuch, A., Kulesza, R.: Sieć procesorów o łagodnej degradacji i strukturze logicznej typu sześcianu 4-wymiarowego. In: Metody wytwarzania i zastosowania systemów czasu rzeczywistego, L. Trybus, Ed., pp. 219–232. Wydawnictwo komunikacji i Łączności, Warszawa (2011) (in Polish)
5. Arciuch, A.: Reliability state of connections in a microprocessor network with binary hypercube structure. R.86, No. 9, pp. 154–156. Electrical Review (2010)
6. Ishikawa, T.: Hypercube multiprocessors with bus connections for improving communication performance. IEEE Trans. Comput. **44**(11), 1338–1344 (1995)
7. Izadi, A.B., Özunger, F.: A Real-time fault_tolerant hypercube multiprocessor. IEEE Proceedings—Computers and Digital Techniques, vol. 149, no. 5, pp. 197–202 (2002)
8. AlBdaiwia, B.F., Bose, B.: On resource placements in 3D tori. J. Parallel Distrib. Comput. **63**, 838–845 (2003)
9. AlBdaiwia, B.F., Bose, B.: Quasi-perfect resource placements for two-dimensional toroidal networks. J. Parallel Distrib. Comput. **65**, 815–831 (2005)
10. Bae, M.M., Bose, B.: Resource placement in torus-based networks. IEEE Trans. Comput. **46** (10), 1083–1092 (Oct 1997)
11. Imani, N., Sarbazi-Azad, H., Zomaya, A.Y.: Resource placement in Cartesian product of networks. J. Parallel Distrib. Comput. **70**, 481–495 (2010)
12. Moinzadeh, P., Sarbazi-Azad, H., Yazdani, N.: Resource placement in cube-connected cycles. In: The International Symposium on Parallel Architectures, Algorithms, and Networks, pp. 83–89. IEEE Computer Society (2008)
13. Chudzikiewicz, J., Zieliński, Z.: On some resources placement schemes in the 4-dimensional soft degradable hypercube processors network. In: Advances in Intelligent and Soft Computing. Proceedings of the Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX (W. Zamojski at al. Eds., Series Ed.: Kacprzyk Janusz), pp. 133–143. Springer (2014)
14. Zieliński, Z.: Podstawy diagnostyki systemowej sieci procesorów o łagodnej degradacji i strukturze hipersześcianu. Wojskowa Akademia Techniczna, Warszawa (2012), 182p (in Polish)
15. Malinowski, T., Arciuch, A.: The procedure for monitoring and maintaining a network of distributed resources. In: Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, vol. 2, pp. 947–954. ACSIS (2014). doi:10.15439/2014F159
16. Sethi, A.S., Hnatyshin, V.Y.: The Practical OPNET User Guide for Computer Network Simulation. Chapman and Hall/CRC (2012)

Model and Case Studies of the Runtime UI Adaptation Process in Client-Server Systems

Jacek Chmielewski

Abstract The increasing diversity of end devices used by users to access their applications and systems strengthens the need for device-independent methods for implementing these applications. The Device-Independent Architecture (DIA) is one of the available approaches to this problem, but it does not directly address the issue of user interface (UI) device-independency. This issue can be addressed by runtime UI adaptation, but it is not clear whether the DIA architecture requires new UI adaptation methods or may use existing ones. Through theoretical model-based analysis of UI adaptation in various client-server systems and through case studies of practical UI adaptation solutions we came up with a conclusion that the DIA-based systems may use existing runtime UI adaptation methods. However, they have to be used with a different set of optimization criteria.

Keywords Runtime UI generation • UI adaptation • Device-independency • Mobile applications • Device-Independent Architecture

1 Introduction

The development of software applications that use end devices to communicate and interact with users becomes a complex and time-consuming issue. The increasing diversity of Internet-connected end devices (especially mobile devices) forces application developers to implement multiple variants of each application. Each software platform (Windows, Android, iOS, etc.) and each device type (smartphone, tablet, laptop, watch, glasses, smart TV, etc.) has its own requirements and

J. Chmielewski (✉)
Department of Information Technology, Poznan University of Economics and Business,
Al. Niepodległości 10, 61-875 Poznań, Poland
e-mail: chmielewski@kti.ue.poznan.pl

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent
Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_3

constraints, which makes it difficult to address all of them with a single uniform implementation. Device-independency of the application logic and data is hindered by different programming languages and disparate APIs supported by different software platforms. Device-independency of the application user interface (UI) is even harder to address because of the number and diversity of possible input and output user communication channels—starting with screen sizes and resolutions, and ending with non-standard symbolic interfaces popular in the Internet of Things domain.

To cope with this problem we have proposed the Device-Independent Architecture (DIA) [1] which solves the logic and data device-independency issues. However the DIA does not directly address the UI device-independency, which is supposed to be solved with proper UI design [2, 3] and UI adaptation [4, 5].

To make sure the DIA does not hinder the ability to use UI adaptation to provide UI device-independency in the reported research we have sought to answer the following question: **May DIA-based software use existing runtime UI adaptation methods?**

This work is an extended version of [6] and presents results of our analysis of this issue. To be able to properly analyze the problem we have defined a model of the runtime UI adaptation and generation process. We have used this model to theoretically examine the runtime UI adaptation and generation process in various software architectures similar to the DIA. Additionally we have performed a series of case studies of real implementations of UI adaptation methods to check if these practical solutions confirm our theoretical conclusions and to get some insights on applicability of these methods in the context of device-independency.

Our main findings are the following. Through our research we have shown that DIA-based software may use existing runtime UI adaptation methods designed for client-server systems. We have learned that the main limiting factor for DIA-based implementations of these UI adaptation methods is not the performance of an end device, as it is often the case of typical mobile UI adaptation implementations, but network latency and throughput. Therefore, to provide properly optimized UIs for DIA-based solutions, existing runtime UI-adaptation methods have to be used with a different set of key metrics and guidelines. Moreover, the case studies of real systems highlight the fact that traditional UI adaptation approaches are not well-suited for device-independency and are not capable of fully achieving the UI device-independency.

The chapter is composed of four sections. Section 1 is the introduction. Section 2 provides background information on the topics of UI adaptation and the Device-Independent Architecture. Section 3 contains a definition of the model of the UI generation process, theoretical discussion and overview of case studies. The chapter is concluded in Sect. 4.

2 Background

2.1 UI Adaptation

UI adaptation activities can be split into two phases: design-time UI adaptation and runtime UI adaptation. These two UI adaptation phases focus on different aspects that may influence the UI adaptation process. The whole process, with its various aspects, is best described by the CAMELEON Reference Framework [7], which provides designers and developers with generic principles for structuring and understanding a model-based UI development process. Model-based approaches [8], which rely on high-level specifications, provide the foundations for code generation and code abstraction. The framework fuses together different models that influence the overall UI adaptation.

As shown in Fig. 1, the framework covers the inference process from high-level abstract descriptions to runtime code, using a four-step reification process:

1. **Concepts-and-Tasks Model (CTM)** the highest level on which logical activities, with are required to reach users goals, and domain specific objects, which are manipulated by these activities, are specified at design-time.
2. **Abstract User Interface (AUI)** level, on which a UI is represented as a collection of presentation units, which are abstract in the sense of their final presentation (independent of the modality of interaction), as they can be presented (accessed) in different ways, e.g., visually, vocally, haptically, etc. Presentation units typically focus on semantics and general properties of UI components (e.g., input/output data components as well as structural and logical

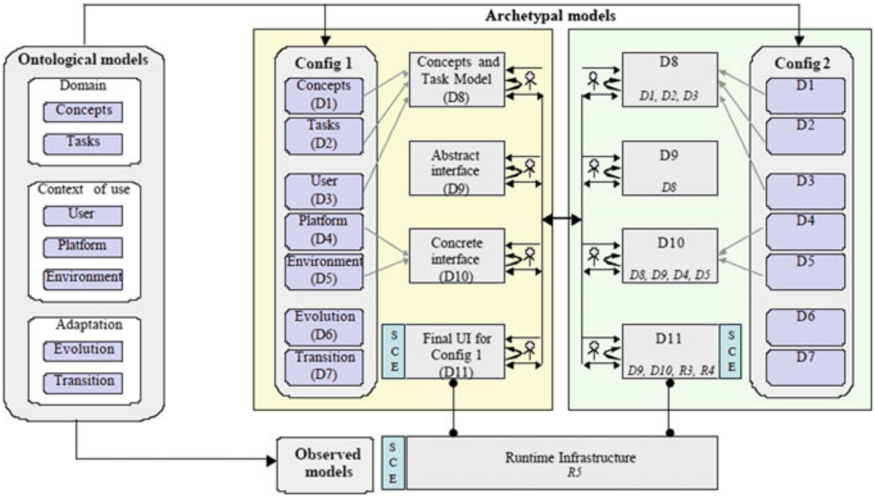


Fig. 1 CAMELEON reference framework

dependencies between components), without covering exact properties related to the final presentation of the UI (e.g., size, position, color of components).

3. **Concrete User Interface (CUI)** level, on which a particular modality is selected and a number of additional descriptive attributes are introduced to an AUI description, which has been created at AUI level, to describe the UI more precisely and to enable the perception of a UI by a user, independently from a particular presentation platform (e.g., layouts, relative size and position of components). At this level the look and feel of a UI is defined, but the description is still device-independent.
4. **Final User Interface (FUI)** level, on which a CUI description, which has been created at CUI level, is encoded in a particular UI description language (a programming language or a markup language, e.g., Java, HTML5, VoiceXML, etc.). A FUI is typically specific to a selected modality of interaction as well as particular hardware and software platform (device, operating system, presentation tools), as it may specify, e.g., properties depending on screen resolution, or type of keyboard. A FUI description may be either compiled or interpreted. It may be presented in various forms, on various platforms depending on, e.g., device capabilities, implementation, or the context of interaction.

At each step the reification is influenced by the "context of use", defined as a set of parameters describing: a user, a platform and the environment. Most of this process belongs to the design-time phase. The runtime phase usually includes only the last reification from the CUI (device-independent) to the FUI (device-specific) and translations between FUI variants.

Both UI adaptation phases are different in nature. In our research on device-independent systems we do not address general UI design issues and we focus on the runtime UI adaptation phase, assuming that the design-time phase produces a device-independent UI description, which is used as a starting point for the runtime UI adaptation.

2.2 Device-Independent Architecture

The Device-Independent Architecture (DIA) has been proposed to facilitate analysis and development of applications that can be made available to users via any capable device from the large, diverse and fast growing pool of Internet-enabled end devices—i.e., devices that are used directly by users to interact with an application, but not sensors that passively record a state of an environment. As presented in [9], the idea of DIA originates from the Service-Oriented Architecture, where systems are decomposed into atomic services, and processes use such services without knowing much about their implementation. A similar approach can be used to decompose an end device. Each end device, be it a laptop or a smartphone, provides: resources, services, and user interaction channels. Resources encompass processing power, memory and storage. Services are providers of context information, such as location, temperature, light intensity, and data from other types of sensors. User interaction channels (both incoming and outgoing) are the means to communicate with a user and include: screen, vibration, keyboard, microphone, camera, etc. The key concept is to use external resources, instead of what is provided by an end device, and to generalize the way services and user interaction channels are accessed. Therefore, in DIA, the separation of application from end devices, which enables the device-independency, is achieved by:

- executing an application outside of end devices,
- accessing sensor data provided by a device via a standardized API,
- using universal UI descriptions, and
- communicating with a user via a set of well-defined user interaction channels.

The execution of the application on external resources ensures that the application logic does not depend on the hardware or software platform of an end device. The interesting consequence is that, in this architecture, end devices could be deprived of their general purpose resources, as these resources are not needed. Services publish data in service-specific formats (e.g., location coordinates for a geolocation service, numerical data for a temperature sensor, and so on) independently of their implementation on a particular end device. Therefore, it is feasible to build a middleware providing a device-independent API, such as the one proposed in Wolfram Language [10], to access such services. The usage of a universal UI description is a key requirement for making the UI of an application independent of parameters of user communication channels available on a given end device (e.g. screen size and pixel density).

However, to enable a UI presentation tailored to parameters of a specific end device, the generic UI description has to be properly adapted before reaching the user. With DIA, the UI adaptation may be introduced either on the server-side or on the client-side (see Fig. 2). In some cases it is also possible to make the process application-agnostic and provide it as a middleware service. This is similar to how existing runtime UI adaptation methods are employed in client-server systems. That is why we have decided to research whether DIA-based software may use existing runtime UI adaptation methods used in client-server systems.

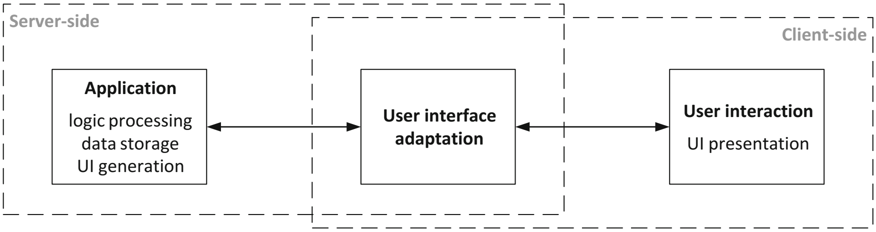


Fig. 2 UI adaptation in the device-independent architecture

3 Model and Analysis

Runtime UI adaptation is a process that transforms a high-level, device-independent UI description (often model-based) prepared at design-time into a final UI presentation. In ideal situations, the high-level UI description may be presented in different ways depending on the UI modality of available user communication channels. For example, presentation of the same UI could be done on screen (Graphical User Interface (GUI)) or via speakers (e.g. Voice User Interface (VUI)). In general, the execution of a runtime UI adaptation process requires three parameters: the UI description, the content and a context of use. The content is used to fill-in the UI. The context of use influences the UI adaptation process and allows tailoring the final UI to the user, her end devices and situation (location, time, etc.).

3.1 Runtime UI Adaptation Model

To be able to analyze the UI adaptation in different application architectures we have defined a simplified model of the UI adaptation and generation process, called the GARP model. The GARP model, presented in Fig. 3, is composed of four main steps:

1. **Input gathering (G)**. At the beginning of the process it is necessary to gather all input required for UI adaptation. The result of this step is a triplet of: UI description, content and context.
2. **Adaptation (A)**. In this step the content is used to fill-in the UI template derived from the UI description and the context is used to guide the transformation of the UI into a final UI tailored for the user, her end devices and situation. The result of this step is a device-specific UI description encoded with a domain specific UI language such as HTML, QML, etc.
3. **Rendering (R)**. The device-specific UI description provided by step 2 is interpreted here, in step 3 and the final UI presentation form is calculated. The final UI presentation form is data (often binary data) prepared for a specific user communication channel, e.g. pixels for screen or audio bits for speakers.
4. **Presentation (P)**. The last step of the process is about presenting the UI to the user using a specific user communication channel of a specific end device, e.g. showing images on screen or playing audio through speakers.

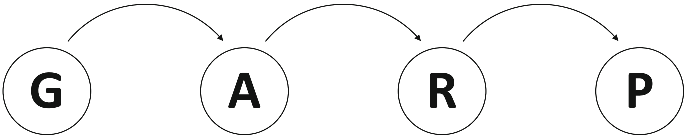


Fig. 3 Model of the runtime UI adaptation and generation process

To make the analysis easier to follow, the model represents only the way towards a user and ignores the process of recording and interpreting user actions. Nevertheless, the path from a user can be modelled in a similar way, so our claims are valid for the whole user interaction loop.

For the analysis we have identified three classes of systems—the class of DIA-based systems and two comparable types of systems also based on the client-server paradigm:

1. **Client-side adaptation systems (CSA)**. Systems of this class include applications that are executed entirely on an end device (a.k.a. local applications) and client-server applications with UI adaptation done on the client side.
2. **Server-side adaptation systems (SSA)**. This class includes client-server applications with UI adaptation performed on the server side.
3. **DIA-based systems (DIA)**. which include applications based on the Device-Independent Architecture.

Our goal was to see how the UI adaptation process differs among these classes of systems and how these differences influence the applicability of known UI adaptation methods. We acknowledge existence of in-between solutions. However, these three classes were selected to clearly show differences in the UI adaptation process.

3.2 UI Adaptation in CSA Systems

The UI adaptation and generation process in CSA systems is done either entirely on an end device (client side) or the G step is supported by the server side, which provides for example the content, UI description or user preferences. However, the fragment of the context gathering related to the end device is local, so the G step can be seen as a task performed jointly by the server side and the client side. The way the G step is performed (locally or split between server and client sides) does not influence the actual UI adaptation, because from the point of view of the A step the results of the G step are always provided in the same way—locally (Fig. 4).

In CSA systems the A step may be implemented using existing UI adaptation methods designed for the use on an end device. These methods are optimized for

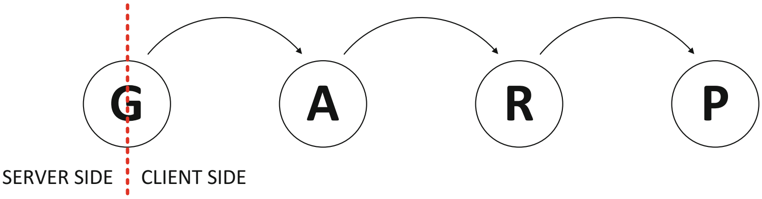


Fig. 4 GARP model in CSA systems

potentially limited processing capabilities of end devices and are closely related to end device characteristics and usage scenarios.

The local UI adaptation methods include solutions built-in into iOS and Android mobile operating systems and used by multiple mobile applications that run on various smartphones and tablets. On these mobile platforms the main issue is the diversity of screen sizes and pixel densities, so it is assumed that each application provides multiple variants of graphical assets (tailored to different screen densities) and some kind of a flexible layout that can be recalculated for any screen size. The drawback of these UI adaptation methods is that they are designed to cope with hardware parameters of a 'standard device' (in most cases a device with a touch-screen). Any UI adaptation that is supposed to take into account for example user preferences or non-standard devices, is not supported by the platform and has to be implemented manually.

The use of the server side for the G step usually does not change the fact that the adaptation implemented on the client side is somehow bound to the characteristic of an end device. In our previous research [11–13] we have analyzed solutions that go beyond this local-only approach and use the server side to provide UI adaptation hints embedded in the high-level UI description provided by the G step, but even such extensions do not change the fact that the UI adaptation itself is device-specific, which makes it hard to reuse on other types of devices (devices with different hardware components, e.g. with two screens).

The Transparent Boundary Service (TBS) presented in [11] is a domain-specific implementation of a universal and adaptive approach to automatic, just-in-time (JIT) generation of user interfaces for mobile and stationary devices based on the application, user preferences, device capabilities and context. The system, tailored for Intelligent Buildings domain, offers two modes of UI generation process: remote and local. Remote UI generation is used for basic end devices with very limited processing capabilities, while the local UI generation is used for more capable devices. In the case of the local mode, the whole TBS runs on an end device and gathers information necessary for UI generation from external modules such as User Preferences Database or Service Interface Cues. What is interesting here, is the aspect of UI generation hints provided by the Service Interface Cues module. By default, a device-independent UI description is derived from a service description and usually results in a quite raw UI. The concept of UI hints enables preparation of more user-friendly user interfaces. The local TBS implementation is device-specific and thus available only for selected end devices.

In the SOA-based system presented in [12, 13] the UI adaptation is supported by the Adaptable SOA Interface System (ASIS). The ASIS framework, presented in Fig. 5, consists of four main elements: SOA Interface Generator Module (SIG), SOIL Interface Templates, Content Access and Adaptation Module (ADAM), and Content Upload and Adaptation Module (UMOD).

The SOA Interface Generator (SIG) consists of an interface generation server and a client application. SIG processes interface templates encoded in the SOIL language, generates the user interface and sends it to the client devices.

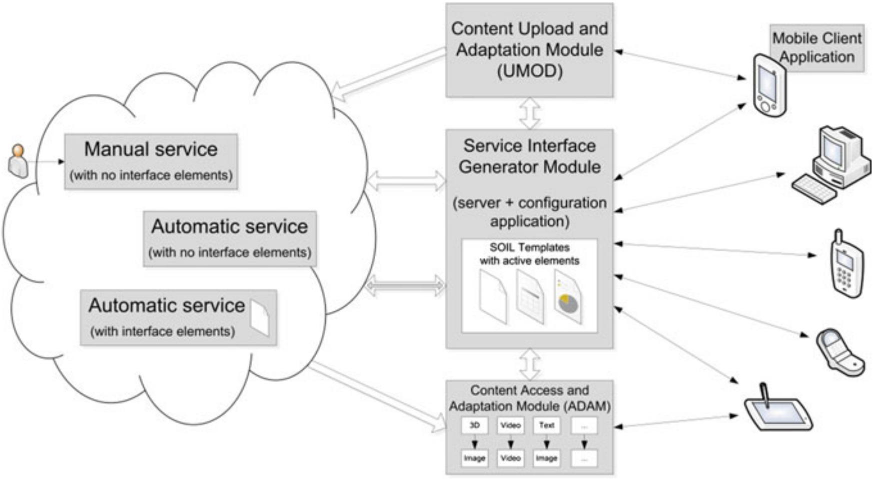

Content Upload and Adaptation Module (UMOD)
Mobile Client Application
Manual service (with no interface elements)
Service Interface Generator Module
(server + configuration application)
SOIL Templates with active elements
Automatic service (with no interface elements)
Automatic service (with interface elements)
Content Access and Adaptation Module (ADAM)
3D
Video
Text
...
Image
Video
Image
...

Fig. 5 The overall architecture of the ASIS framework

The SOIL Interface Templates enable SOA services to be presented to end-users in a user-friendly way. The templates are encoded in a specially designed language, called SOIL—Service-Oriented Interface Language. SOIL is based on XML. It provides commands that can control the process of interface generation, communicate between themselves, and execute calls to SOA services.

The Content Access and Adaptation Module (ADAM), and the Content Upload and Adaptation Module (UMOD) are responsible for providing access and receiving multimedia content, adjusting the formats and properties of these objects to make them suitable for presentation on a particular client device or for storage within the system.

By default, a standard web browser is used on the client device and the ASIS framework creates the interface description on request received from the end-user device, based on interface templates coded in the SOIL language, parameters of the end-user device—either sent by the device or stored in a local database, and the current context of interface generation. The interface description is then sent to the client device and rendered by the web browser. Despite being an SSA-class system, the ASIS framework includes a custom client application, which replaces the standard web browser and supports client-side UI adaptation for specific mobile devices. This application receives a UI description pre-filled with data and uses local knowledge about the device and context to prepare the final user interface. The application is available for a handful of selected mobile devices and the UI adaptation process is device-specific.

Both presented systems use the client-side UI adaptation and generation only as an additional feature to the more universal server-side adaptation and only for specific end devices. Which confirms that this UI adaptation and generation mode is useful only in specific usage scenarios and is usually highly device-specific.

3.3 UI Adaptation in SSA Systems

In SSA systems the two initial steps: G and A, are performed on the server side, and the two other steps: R and P, are performed on the client side. The server gathers all input data, runs the UI adaptation and sends the device-specific UI description to the client. The client then interprets the UI description and presents it to a user (Fig. 6).

The SSA systems approach the issue of portability of the UI adaptation, shown for CSA systems, by implementing the A step on the server side. Such approach means that the UI adaptation is not bound by the performance of an end device and can use external services to support the UI adaptation task (e.g. multimedia converters). Results of our previous research on UI adaptation in SSA systems [13–16] confirm that the A step in SSA systems may accommodate end devices with disparate hardware configurations by using multiple or dynamic UI adaptation scenarios. However, the result of the A step is still interpreted on an end device. Therefore is susceptible to differences in the final rendering and presentation on different end devices. So full control of the resulting UI is not possible.

The already mentioned ASIS framework uses server-side UI generation and adaptation process to build universal user interfaces for any end device supporting a web browser. This approach makes it possible to address a wide range of possible end devices, but suffers from incompatible web browser implementations.

The method of adapting a user interface developed for industrial process monitoring and control applications, presented in [14], which also uses web technologies for the final UI, tries to cope with this problem introducing additional dynamic UI optimization rules. Custom HTTP headers and additional request-response cycle make it possible to precisely identify the target web browser and initiate a set of UI adaptation and transformation rules tailored specifically for this web browser. With this approach it is possible to reach an acceptable level in UI uniformity, as presented in Fig. 7.

The Mobile Interfaces for Web 2.0 LEarning Systems (MILES) system, described in [15, 16], is an extension to an e-learning platform developed in a European LLP project Web 2.0 ERC. The MILES system is based on a new approach to building adaptable user interfaces for e-learning repositories and resources. In this approach, the e-learning platforms are accessed by the use of Web

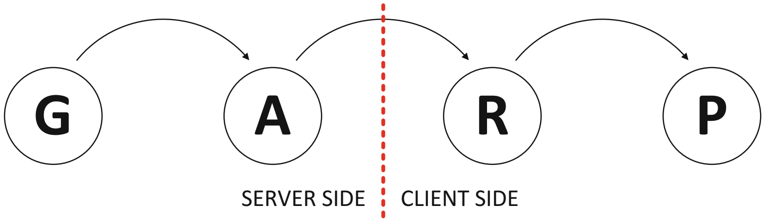


Fig. 6 GARP model in SSA systems

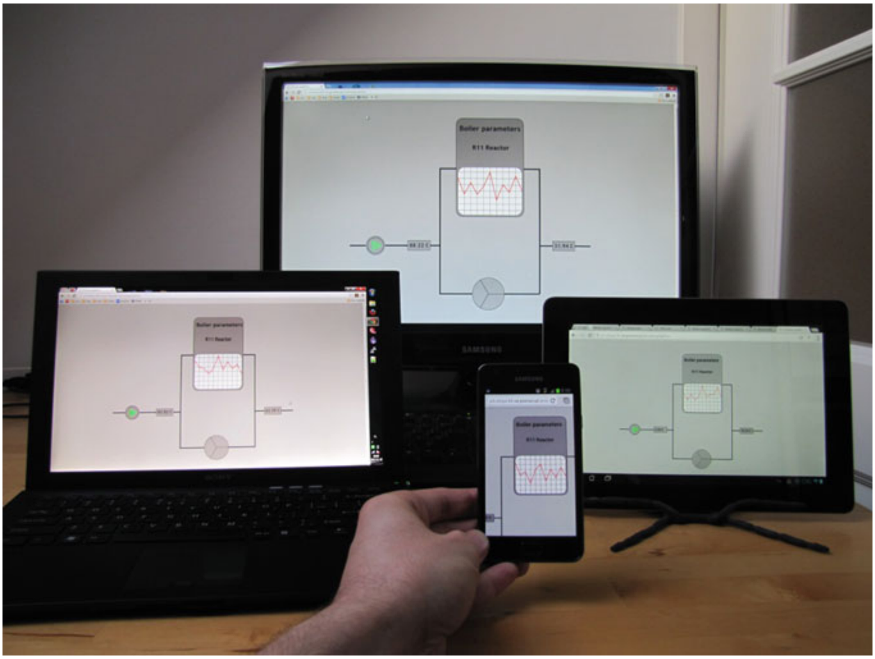


Fig. 7 Example of a uniform cross-device web-based UI

Services instead of typical web-based interfaces. Specific services have been designed to enable high-level operations on Web 2.0 resources without the need to use their complex web interfaces. The interface generation and adaptation combines dynamically generated web user interfaces with results of Web Service calls. Examples of such generated UIs are presented in Fig. 8.

These three systems employ web technologies and a custom UI generation and adaptation process to provide universal user interfaces for a diverse set of end devices. However this diversity and the resulting diversity among web browsers used to interpret, render and present the UI, causes UI inconsistencies that usually out of control.

3.4 UI Adaptation in DIA Systems

The Device-Independent Architecture is based on an assumption that the whole processing is done outside of an end device (the client side) and the end device receives a pre-rendered UI ready for presentation, without the need for any interpretation. So in the case of DIA systems all three initial steps of the GARP model are done on the server side and only the P step is performed on the client side. The data transferred between the server and the client is usually a stream (e.g. a video or

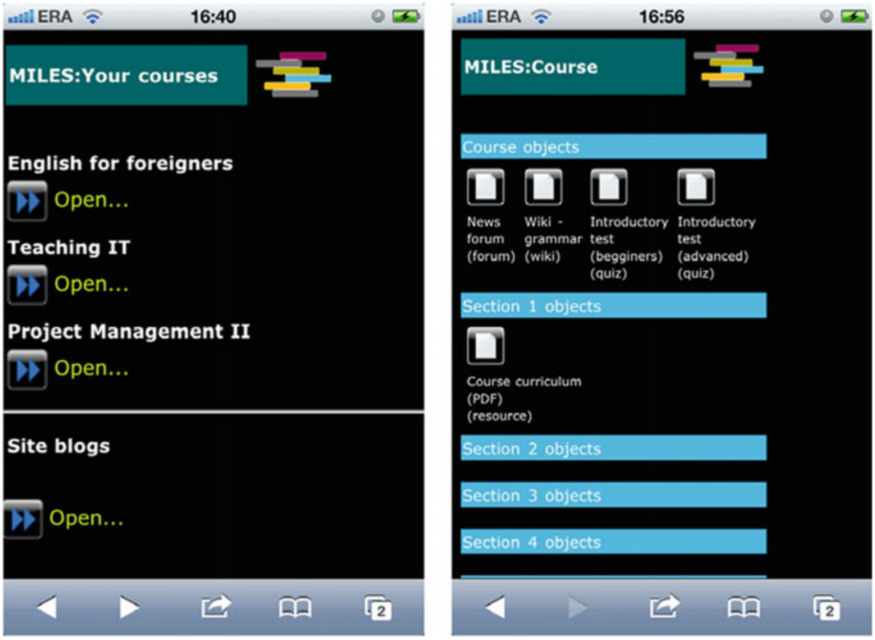


Fig. 8 Examples of web-based UI generated by the MILES system

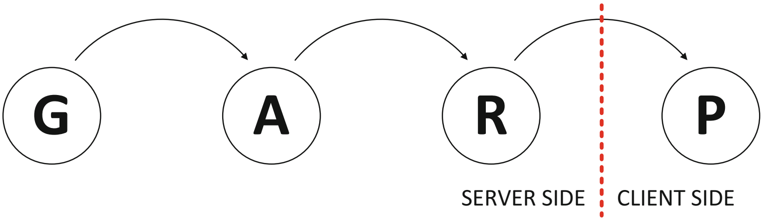
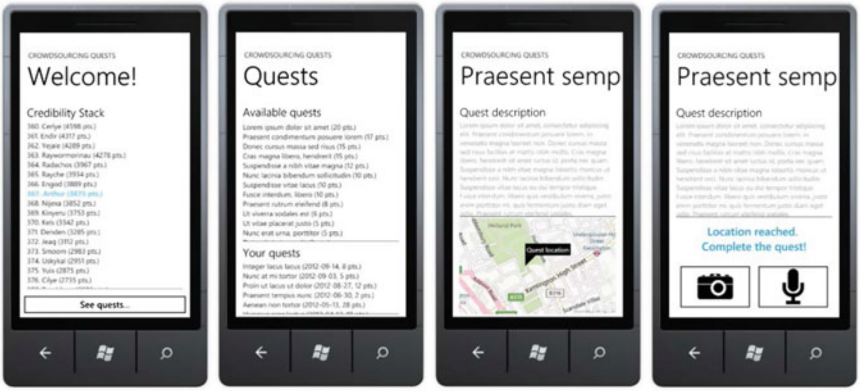


Fig. 9 GARP model in DIA systems

audio stream) or a static UI state (e.g. an image of the UI to be presented on screen or audio file to be played through speakers) ready to be presented on an end device (Fig. 9).

The DIA approach enables full control over the final UI presented to a user by implementing also the R step on the server side. UI adaptation methods used in DIA systems can be still the same as for SSA systems, but the fact that the end device handles only the P step ensures that devices will not show a UI in a way that deviates from intentions of a designer. The consequence of moving the R step to the server side is a different kind of data being transferred between the server side and the client side. In SSA systems, the client side receives a device-specific UI description encoded in a specialized UI language. In DIA systems, the server side



360. Cerlye (4598 pts.)
361. Endir (4317 pts.)
362. Yejaie (4289 pts.)
363. Raywormorinau (4278 pts.)
364. Radachos (3967 pts.)
365. Rayche (3934 pts.)
366. Engod (3889 pts.)
367. Arthur (3875 pts.)
368. Nijexa (3852 pts.)
369. Kinyeru (3753 pts.)
370. Kels (3342 pts.)
371. Deriden (3285 pts.)
372. Jeaq (3112 pts.)
373. Smoom (2983 pts.)
374. Uskykal (2951 pts.)
375. Yuis (2875 pts.)
376. Cilye (2735 pts.)

Lorem ipsum dolor sit amet (20 pts.)
Praesent condimentum posuere lorem (17 pts.)
Donec cursus massa sed risus (15 pts.)
Cras magna libero, hendrerit (15 pts.)
Suspendisse a nibh vitae magna (12 pts.)
Nunc lacinia bibendum sollicitudin (10 pts.)
Suspendisse vitae lacus (10 pts.)
Fusce interdum, libero (10 pts.)
Praesent rutrum eleifend (8 pts.)
Ut viverra sodales est (6 pts.)
Ut vitae placerat justo (5 pts.)
Nunc erat urna, porttitor (5 pts.)

Integer lacus lacus (2012-09-14, 8 pts.)
Nunc at mi tortor (2012-09-03, 5 pts.)
Proin ut lacus ut dolor (2012-08-27, 12 pts.)
Praesent tempus nunc (2012-06-30, 2 pts.)
Aenean non tortor (2012-05-13, 28 pts.)

Fig. 10 UI views of a DIA-based application

has to send either a continuous streams of data tailored for specific user communication channels (e.g. video stream for a screen or an audio stream for speakers) or a static UI state composed of multiple files that are targeted at different user communication channels (e.g. image files to be shown on a screen or audio files to be played through speakers). The main difference here is the increased size of data that has to be transferred. More data to transfer could mean longer response times, but our previous research [17] shows that in the analyzed scenarios DIA-based systems can still maintain proper response times to UI interactions initiated by a user, despite the increased size of transferred data.

The DIA-based application implemented for the research reported in [17] is a prototype of a mobile crowdsourcing application. It features a static, state-based user interface composed of a few screens. The UI is presented in Fig. 10. For comparison, the same functionality and UI features were implemented as a native Windows Phone application. Results of the research confirm that UI response times of a DIA-based application is two to three times slower than for the native application. However, for this application, the standard HTTP protocol and GIF image format were sufficient to maintain the UI response time below 1 s, which is assumed to be reasonable for a static, state-based UI [18]. So, with a proper transmission protocol and image format it is feasible to keep the UI response time of such applications within a range that does not hinder the application usability.

4 Conclusion

The Device-Independent Architecture can be treated as a special case of a client-server architecture, in which the client side is assumed to be an extremely thin client and in which all the processing is done on the server side. The DIA takes it even further and defines the client side as a set of user communication channels, which makes it possible to model multiple end devices as a complex client device, but this distinction does not necessarily change the way the UI adaptation is performed. Therefore, DIA systems may use the same existing UI adaptation methods that were designed for CSA and SSA systems, or for client-server systems in general.

The main difference is related to the fact that in DIA-based systems the data transferred between the server side and the client side tends to be larger than in the case of SSA systems. Therefore, network usage optimization is crucial. Especially that the transmission delay will directly influence the UI responsiveness. Moreover, used communication protocols and formats of presentation data sent to an end device have to be negotiated beforehand, to make sure that the end device is able to receive and properly present the UI.

Summarizing, despite using a different implementation of the GARP model, the DIA systems may use existing runtime UI adaptation methods. The difference in the implementation of GARP model influences only the optimization of the UI adaptation and generation process. In CSA systems the key optimization aspect is end device performance. In SSA systems the key optimization aspect is uniform interpretation of the device-specific UI description. While, in DIA systems the key optimization aspects are network-related. First, it is necessary to use data formats that minimize the amount of bits that have to be transmitted. Second, it is crucial to use the best possible data transfer protocols. The best are the ones with low overhead, low latency and support for QoS. Both points should be taken into account by the runtime UI adaptation task, because the nature of a UI (state-based or continuous) may influence the set of suitable transmission protocols.

We expect that different protocols will be best suited for different user interaction scenarios. Our next research goal in this area is to identify user interaction patterns and UI design patterns, which could be used to define rules for selecting the best transmission protocols and data formats for a given user interaction scenario.

References

1. Chmielewski, J.: Towards an architecture for future internet applications. In: The Future Internet, Lecture Notes in Computer Science, vol. 7858, pp. 214–219. Springer, Berlin (2013). ISBN 978-3-642-38081-5, doi:10.1007/978-3-642-38082-2_18
2. Meixner, G., Calvary, G., Coutaz, J.: Introduction to model-based user interfaces. In: W3C Working Group Note, Dec 2013. http://www.w3.org/2011/mbui/drafts/mbui-intro/
3. Sottet, J.S., Calvary, G., Favre, J.M., Coutaz, J.: Megamodeling and metamodel-driven engineering for plastic user interfaces: MEGA-UI. In: Human-Centered Software Engineering, pp. 173–200. Springer, London (2009). doi:10.1007/978-1-84800-907-3_8
4. Jaouadi, I., Ben Djemaa, R., Ben Abdallah, H.: Interactive systems adaptation approaches: a survey. In: ACHI 2014, The Seventh International Conference on Advances in Computer-Human Interactions, pp. 127–131, Mar 2014. ISSN: 2308-4138, ISBN 978-1-61208-325-4

5. Ye, J.H., Herbert, J.: Framework for user interface adaptation. In: User-Centered Interaction Paradigms for Universal Access in the Information Society, pp. 167–174. Springer, Berlin (2004). doi:10.1007/978-3-540-30111-0_14
6. Chmielewski, J.: Run-time UI adaptation in the context of the device-independent architecture. In: Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, pp. 1157–1162. IEEE (2015). doi:10.15439/2015F259
7. Calvary, G., Coutaz, J., Bouillon, L., Florins, M., Limbourg, Q., Marucci, L., Paternò, F., Santoro, C., Souchon, N., Thevenin, D., Vanderdonckt, J.: The CAMELEON Reference Framework, Deliverable 1.1, CAMELEON Project (2000)
8. Szekely, P.: Retrospective and challenges for model-based interface development. In: Design, Specification and Verification of Interactive Systems ’96, Eurographics 1996, pp. 1–27. Springer, Vienna (1996). doi:10.1007/978-3-7091-7491-3_1
9. Chmielewski, J., Walczak, K.: Application architectures for smart multi-device applications. In: Proceedings of the Workshop on Multi-device App Middleware 2012, Workshop on Multi-device App Middleware 2012, Montreal (Canada), 3–7 Dec 2012, pp. 5:1–5:5. ACM, New York (2012). ISBN 978-1-4503-1617-0, doi:10.1145/2405172.2405177
10. Wolfram Language for Knowledge-Based Programming (2015). https://www.wolfram.com/language/
11. Rykowski, J., Chmielewski, J.: Automatyczna generacja zintegrowanego interfejsu człowiek-maszyna na potrzeby inteligentnego budynku. In: Inteligentne budynki - teoria i praktyka, ed. Mikulik, J., Oficyna Wydawnicza Text, Kraków (2010), pp. 166–188. ISBN 978-83-60560-54-9
12. Chmielewski, J., Walczak, K., Wiza, W.: Mobile interfaces for building control surveyors. In: Cellary, W., Estevez, E. (eds.) Software Services for e-World, IFIP Advances in Information and Communication Technology, vol. 341. The 10th IFIP WG.6.11 Conference on e-Business, e-Services and e-Society I3E 2010, Buenos Aires, Argentina, 3–5 Nov 2010, pp. 29–39. Springer (2010). ISBN 978-3-642-16282-4, doi:10.1007/978-3-642-16283-1_7
13. Chmielewski, J., Walczak, K., Wiza, W., Wójtowicz, A.: Adaptable user interfaces for SOA applications in the construction sector. In: Ambroszkiewicz, S., Brzeziński, J., Cellary, W., Grzech, A., Zieliński, K. (eds.) SOA Infrastructure Tools—Concepts and Methods. Wydawnictwa Uniwersytetu Ekonomicznego w Poznaniu, Poznań (2010), pp. 493–469. ISBN 978-83-7417-544-9
14. Jansen, A., Bronmark, J., Chmielewski, J.: Method of adapting a user interface in industrial process monitoring and control applications. The Swedish Patent and Registration Office. SE 1300702-6 (2013)
15. Walczak, K., Wiza, W., Rumiński, D., Chmielewski, J., Wójtowicz, A.: Adaptable user interfaces for Web 2.0 educational resources. In: Kiełtyka, L. (ed.) IT Tools in Management and Education—Selected Problems, pp. 104–124. Wydawnictwo Politechniki Częstochowskiej, Częstochowa (2011). ISBN 978-83-7193-508-4
16. Walczak, K., Chmielewski, J., Wiza, W., Rumiński, D., Skibiński, G.: Adaptable mobile user interfaces for e-learning repositories. In: IADIS International Conference on Mobile Learning, Avila (Spain), 10–12 Mar 2011, pp. 52–60. IADIS (2011). ISBN 978-972-8939-45-8
17. Chmielewski, J.: Device-independent architecture for ubiquitous applications. In: Personal and Ubiquitous Computing, vol. 18(2), pp. 481–488. Springer, London (2014). doi:10.1007/s00779-013-0666-y
18. Nielsen, J.: Usability Engineering. Morgan Kaufmann, Sept 1993. ISBN 978-01-2518-406-9

Analysis of TCP Connection Performance Using Emulation of TCP State

Andrzej Bąk, Piotr Gajowniczek and Michał Zagożdżon

Abstract Transmission Control Protocol (TCP) is still used by vast majority of Internet applications. However, the huge increase in bandwidth availability during the last decade has stimulated the evolution of TCP and introduction of new versions, better suited for high speed networks. Many factors can influence the performance of TCP protocol, starting from scarcity of network resources, through client or server misconfiguration, to internal limitations of applications. Proper identification of the TCP performance bottlenecks is therefore an important challenge for network operators. In the paper we proposed the methodology for finding root causes of throughput degradation in TCP connections using passive measurements. This methodology was verified by experiments conducted in a live network with 4G wireless Internet access. The paper also presents selected details of its practical implementation.

Keywords TCP · Data flow measurements · Performance bottlenecks

1 Introduction

This work is an extended version of [1]. It presents the methodology of finding the root causes of throughput degradation in TCP connections on the base of passive measurements obtained from probes capturing traffic on the network links.

Since the foundation of the Internet the vast majority of network data is transmitted using Transmission Control Protocol (TCP). TCP underlies many 'traditional' Internet applications such as web browsing, email, bulk data transfer etc., but also

A. Bąk (✉) · P. Gajowniczek
Institute of Telecommunications, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: bak@tele.pw.edu.pl

P. Gajowniczek
e-mail: gajow@tele.pw.edu.pl

M. Zagożdżon
Orange Labs, Orange Polska S.A., Obrzeżna 7, 02-679 Warsaw, Poland
e-mail: michal.zagozdzon@orange.com

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent
Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_4

the relatively new ones, such as HTTP adaptive streaming that is quickly becoming the preferred method for Over-The-Top video delivery. All this makes the TCP performance analysis one of the most important research areas of the Internet networking. Since the beginning of TCP's public use in 1989 a lot of research was devoted to improve its performance, and the protocol itself has evolved significantly.

The early TCP version used RTO (Retransmission Time-Out) timer to recover from packet loss, which was inefficient even on low speed links. The TCP Reno/New Reno version [2] introduced fast retransmit & recovery mechanism that improved the TCP performance in presence of a packet loss. For a long time the TCP Reno was a de-facto standard widely deployed in the Internet. However, the significant increase in network capacity observed during the last decade has stimulated introduction of new TCP congestion control algorithms that are more suited for high speed links, such as Fast TCP [3], BIC [4], STCP [5], CUBIC [6–8], HTCP [9–11], HSTCP [12, 13], Compound TCP [14], TCP Westwood [15] etc. The new versions of Linux operating system do even allow switching between different congestion control algorithms without the need to recompile the kernel.

The methodology proposed in the paper results from the need to quickly test the effectiveness of end-users' TCP connections in the actual 4G mobile network. The emphasis was put on feasibility and a potential for automation. The latter is understood as a capability to implement the method in an application that would run unsupervised, getting input from current network measurements, and being able to provide warnings when issues are detected on tested TCP connections. The resulting approach combines the detection of TCP source application type (greedy vs non-greedy), based on [16] with estimation and analysis of certain metrics related to TCP transmission effectiveness, inspired by RFC 6349 [17].

The methodology is validated using the results obtained from measurements conducted in the live 4G mobile network of Orange Poland. Most of the data required to implement the proposed approach can be taken directly from TCP traces, but not all of them. Some parameters, such as state variables maintained by the TCP sender process can be only estimated by emulating the relevant TCP mechanisms using the recorded packet traces. Therefore, the selected implementation aspects, especially these related to the estimation of the value of congestion window that significantly impacts TCP throughput, are also discussed in this work.

2 Sources of TCP Performance Bottlenecks

TCP uses congestion and flow control mechanisms to control the transmission rate of the sender process by limiting the amount of data that can be transmitted without waiting for acknowledgment (called the *window size*). Changing transmission rate in response to receiver's limitation in processing incoming data (*flow control*) is based on the current size of the receiver's window (*awnd*). This value is advertised to TCP sender process in segments that are sent as acknowledgments to the received data. Too small values of the receiver's window can however negatively affect the performance of the TCP protocol. Therefore, in newer implementations it can be adapted algorithmically depending on the characteristics of the transmission path (such as throughput and delay).

Congestion control is done by algorithms that aim to ‘sense’ the bottleneck throughput on the transmission path and adapt the transmission rate to this limit. The sender process keeps the state variable called the congestion window (*cwnd*) that works in a similar way as advertised window, except that its value is set by an algorithm running on the sender side. Usually, the sender starts with a small value of *cwnd* and tries to increase it each time when an acknowledgment for the previously sent data segment is received. The initial phase of aggressive *cwnd* increase is called a *slow start*—the *cwnd* is increased by 1 segment after each acknowledgment which leads to exponential growth in the amount of transmitted data. After encountering data loss the *cwnd* shrinks; the following increase is usually slower and TCP sender enters the phase called *congestion avoidance*. There are many different congestion control algorithms and their variants—for an excellent review see [18]. However, they all share the same purpose—to maximize the usage of capacity available on the transmission path while also minimizing the probability of data loss.

For the TCP sender process, the actual window size is the minimum of the advertised receiver’s window (*awnd*) and its own congestion window (*cwnd*). After sending full window of data, TCP must stop transmission and wait for acknowledgment. The acknowledgment related to the earliest outstanding segment that was transmitted will start to arrive after the RTT (Round Trip Time) between the sender and the receiver. Each arriving acknowledgment will trigger transmission of the next segment of data awaiting in the output buffer. Hence, the TCP process can send at most $min(cwnd, awnd)$ of data per round trip time cycle and the instantaneous TCP throughput can be roughly estimated as:

$$TCP_{th} = \frac{min(cwnd, awnd)}{RTT} \; . \qquad (1)$$

TCP window that is too small may severely limit the performance of TCP connection. In order to obtain high throughput, TCP must be able to fill the network pipeline with data that will keep the network busy. Therefore, the TCP sender’s window size must be greater than the bandwidth delay product:

$$min(cwnd, awnd) \geq C * RTT \; , \qquad (2)$$

where C denotes the capacity available for TCP connection on the transmission path.

There are various ways to set the values of *awnd* and *cwnd*. As it was noted earlier, especially for *cwnd* there are numerous algorithms that react differently to the potential congestion detected either by the Retransmission Time-Out (a timer on sender’s side) or by receiving a duplicate acknowledgment (Dup-ACK) from the receiver.

Another factor limiting the TCP performance is related to sizing TCP socket buffers on both sides of the connection. The problem of buffers being too small is especially visible in the networks with high bandwidth delay product (the maximum buffer space for TCP sockets depends on the operating system in case of typical Internet hosts). The receiver's socket buffer size can significantly influence the performance of TCP connection (receiver's buffer can limit the sending rate of the TCP source). Therefore, proper configuration of the sockets' buffers is very important to assure high TCP throughput [19]. Modern operating systems introduce automated algorithms for tuning the TCP buffers [20–24].

Similar case is related to sizing the buffers of network devices [25]. TCP sender can emit data in bursts (up to the current *cwnd* window size). If network buffers are too small, the inevitable data loss will prevent the congestion window from growing and TCP connection will not be able to ramp up the transmission rate to available capacity. It is generally advised that network buffers should be at least twice the size of the network bandwidth delay product to assure high TCP throughput.

The achievable TCP throughput can be also impacted by packet reordering [26, 27] that can be introduced for example by parallel packet processing in network devices. Receiving out-of-order segments can result in duplicate acknowledgments being sent and interpreted as data loss. This may in turn lead to unnecessary retransmissions, *cwnd* reduction and throughput degradation. On the receiver's side frequent segment reordering may lead to extensive buffering and potential reduction in receiver's window size.

Finally, the throughput limitation can lie within the application itself. For example, in adaptive HTTP streaming the client requests chunks of video file from the server with frequency related to the encoding rate, even if the available capacity would allow transmitting data faster. In this context, TCP sources can be divided into greedy (always trying to utilize the most of available transmission capacity) and non-greedy (where rate is limited by internal behavior of the source). This classification is utilized in the methodology discussed in Sect. 3.2.

3 Detection of the root causes of TCP performance degradation

In this section we describe the algorithm for detecting the root causes of the TCP performance bottlenecks using passive TCP measurements.

3.1 Network Measurements

We assume that the TCP traffic is monitored near the sender (at the client or at the server, depending on the direction of the transmission). Following the recommendations from RFC 6349, it is advised to perform the MTU (Maximum Transmission Unit) discovery procedure (see [28] for reference) before starting measurements, to avoid unwanted packet fragmentation. The monitored network traffic is saved by the probes in *.pcap* format for further processing.

The throughput of the TCP connection TCP_{th} can be estimated directly from data captured by passive probes as a ratio of data sent and acknowledged during a measurement period to the length t of this period:

$$TCP_{th} = \frac{ACK(t)}{t} \; . \tag{3}$$

$ACK(t)$ denotes the highest acknowledgment sequence number observed up to time t. It can be obtained directly from the headers of the captured TCP segments.

Due to the nature of congestion and flow control mechanisms, the TCP sender needs some time before it can reach the desired transmission speed. This time may vary from few seconds to even couple of minutes depending on the network RTT, bandwidth, TCP congestion and flow control algorithms etc. For example, during congestion avoidance phase the TCP source needs approximately 30 s to increase the transmission rate by 10 Mbps if the network RTT is 200 ms. Therefore, for proper estimation of the TCP throughput the measurement time should be long enough. The following approach is suggested to assure that. Assuming some interval Δt and threshold c, seek for time instant t that satisfies the following condition:

$$\left| \frac{TCP_{th}(t + \Delta t) - TCP_{th}(t)}{\Delta t} \right| < c \; . \tag{4}$$

The above formula approximates the derivative of TCP rate estimator. The measurement time should be long enough to assure that the TCP rate estimator does not significantly change over time. In the experiments presented further in this paper we assumed $\Delta t = 1s$ and $c = 100$ KB/s^2.

In addition to the typical traffic traces captured at measurement points, the proposed methodology requires running some additional measurements to calculate certain TCP performance indicators (described in Sect. 3.6). The first measurement is related to estimation of reference (bottleneck) bandwidth C_{REF}. This can be achieved by probing the network bottleneck with UDP traffic. There are many variants of this approach—for examples see [29] or [30]. In our measurements we have used the latter: a train of 50 UDP packets was sent to the receiver and the available capacity was measured simply by dividing the total length of the received UDP packets by the total reception time (under the condition of no packet loss).

To verify if the buffers are properly dimensioned, the *back to back frames* test should be also performed. This test consists of sending the specified number of UDP packets with the maximum possible rate and repeating it while increasing the number of transmitted packets in each trial. The maximum batch size that can be sent without observing packet loss is an indirect measure of buffer size on the transmission path of the stream.

3.2 Categorization of TCP Sources

TCP throughput depends on the amount of data the TCP source emits during a single RTT period, as this value is controlled by the congestion and flow control mechanism. At any time instant the amount of outstanding data (sent but not acknowledged) is limited to $min(cwnd, awnd)$. As was explained in Sect. 2 the TCP source can send at most $min(cwnd, awnd)$ bytes per RTT period. Therefore, the amount of outstanding data in relation to the RTT is an indicator of instantaneous TCP performance.

If the amount of outstanding data is less than what *cwnd* and *awnd* parameters allow, it means that the sender is not fully exploiting the available transmission capacity. The cause may be related to internal sender faults (such as application software or hardware issues, CPU overload etc.), but more often is a result of the consent behavior of TCP source (that may not require more throughput, as it is in case of typical streaming applications where transmission speed is related to the bitrate of the video stream). In the opposite case (if the outstanding data is close to the *cwnd* or *awnd*), the bottlenecks are introduced either by the network or by the receiver.

Summarizing the above discussion, the TCP source may fall into one of the following categories:

- *Internally limited*
  Non-greedy source i.e. a TCP connection that is not fully exploiting the capacity available in the network; the amount of outstanding data is significantly lower than the *cwnd* and *awnd* windows would allow.
- *Receiver limited*
  TCP source whose transmission rate is limited by the receiver; the amount of outstanding data is close to the *awnd* and also lower than the *cwnd*.
- *Network limited*
  TCP source whose transmission rate is limited by the network, i.e. by the available capacity, packet loss rate or network RTT; the amount of outstanding data is close to the *cwnd*.

In order to classify the TCP source into one of the above categories we need the following parameters: outstanding data, RTT, receiver's window size and congestion window size. The first three parameters can be easily obtained from the packet traces captured by the passive probes. However, the congestion window is not directly measurable as it is an internal parameter of the TCP stack at the sender and cannot be directly inferred from the TCP traces. In order to cope with this problem, we follow the approach of [16]. The TCP connection state is emulated using the recorded TCP traces to recover the *cwnd* parameter. We also recover the value of RTO to distinguish between retransmissions induced by the fast retransmit phase and those due to the timer expiration. This is required to precisely track the changes in the *cwnd* parameter.

3.3 Emulation of TCP Connection State

The internal state of TCP congestion control mechanism is defined by three main parameters: size of congestion window (*cwnd*), threshold for switching between slow start and congestion avoidance phase (*ssthr*), and the retransmission timer (RTO). These parameters are essential for emulation of the TCP connection state.

After a 3-way handshake procedure, the TCP connection is established and the TCP sender starts to transmit data. The sender sets its *cwnd* parameter to some initial value (in Linux, for example, it is equal to 10 segments) and begins transmitting in the *slow start* mode. While in slow start, TCP adds one segment to the *cwnd* for each acknowledged segment, doubling its *cwnd* every RTT period. Therefore, during slow start TCP throughput grows exponentially. The aim of this phase is to quickly probe the network capacity and to estimate the optimum window size without overloading the network.

Slow start phase ends when either the *ssthr* is reached or the segment loss is detected. TCP detects segment loss by two mechanisms: expiration of RTO timer or reception of duplicate acknowledgments (Dup-ACKs). In the first case the TCP sender retransmits all outstanding data, sets the *ssthr* to the half of *cwnd* observed at timer expiration and enters the slow start mode again. TCP switches to slow start phase also after long inactivity, to protect the network against overload caused by sudden activation of the sender.

In the second case, after receiving 3 consecutive Dup-ACKs the TCP sender enters the recovery phase and employs *fast retransmit & recovery* mechanism to recover the lost segment. Contrary to the RTO mechanism, only one segment is retransmitted here. The assumption behind this approach is that in this case only one segment is most likely lost and there is no need to follow the go-back-N protocol and retransmit all outstanding data. The sender sets the *ssthr* to the half of the *cwnd* window before the segment loss, sets the *cwnd* to $ssthr + 3$ segments and retransmits the segment pointed by Dup-ACKs. Each time another Dup-ACK arrives, the sender adds one segment to the *cwnd* (inflating the congestion window). The aim of this is to sustain the TCP throughput (as Dup-ACK indicates that the network is still able to deliver packets).

In the recovery phase, the TCP sender is allowed to transmit new data as indicated by *cwnd*. The recovery phase ends when all outstanding data from the beginning of this phase is acknowledged. When leaving the recovery phase the TCP sets the *cwnd* back to the *ssthr* and enters the *congestion avoidance* mode. In the TCP Reno/NewReno versions, during the congestion avoidance phase one segment is added to the *cwnd* in each RTT period. This means that the *cwnd* grows linearly over time, increasing TCP throughput more conservatively then in slow start phase. However, in the network with high bandwidth delay product it may take a long time to recover TCP throughput. Therefore, new congestion control mechanisms introduce more aggressive approaches for increasing the *cwnd* during congestion avoidance phase and reducing the *cwnd*/*ssthr* after segment loss detection.

In order to track the sender's *cwnd* we emulate the behavior of the TCP protocol. To obtain high accuracy of the emulation we used the original source code of the Linux kernel version 3.18 [31]. The H-TCP congestion control algorithm code was used as this protocol was employed in our test setup. It is one of the newer TCP adaptations, created to increase its effectiveness in high speed networks. The main change in relation to TCP Reno/New Reno is that in congestion avoidance phase the increase in the quantity of sent data is a polynomial function of time since the last segment loss—the longer is this period, the more aggressively the algorithm tries to increase the speed of transmission. In the actual implementation the code from the following Linux core modules were used:

- *tcp_input.c*: estimation of the RTO algorithm,
- *tcp_htcp.c*: H-TCP congestion control algorithm,
- *tcp_cong.c*: NewReno congestion control algorithm.

The snippets of code emulating TCP sender process are shown in the following listings. They are commented and provide an insight into how the most important procedures related to estimation of the *cwnd* are implemented.

The *DATA* and *ACK* procedures are called respectively for data segments sent by the TCP server and acknowledgments sent by the client (for simplicity of presentation we assumed unidirectional data transfer from server to client).

Listing 1: Global variables and .pcap file handle

```
u_int HZ = 100; // TCP clock resolution (tick/sec)
u_int jiffies;  // time (number of TCP clock ticks)
pcap_t *pcap_handle;
char errbuf[PCAP_ERRBUF_SIZE];
```

Listing 2: Reading data from the .pcap file (fragment of the main() function)

```
pcap_handle = pcap_open_offline("my.pcap",errbuf);
pcap_loop(pcap_handle, 0, packet_handler, NULL);
pcap_close(pcap_handle);
```

TCP traffic data is read from .pcap traces registered by monitoring devices in the network. To read the trace files the *winpcap* library was used. This library provides the *pcap_loop*() function allowing iterative handling of captured frames (see Listings: 1 and 2). One of its arguments is the pointer to a function called *packet_handler* that is called consecutively for each frame being read.

Listing 3: Segment type detection

```
// Handling of captured frames
void packet_handler(u_char *, const struct pcap_pkthdr *header,
  const u_char *frame) {
    timestamp = header->ts.tv_sec + header->ts.tv_usec / 1000000.0;
    // Conversion of time to clock ticks
    jiffies = round(timestamp * HZ);
    ih = (ip_header *)(pkt_data + 14); // IP header
    ip_len = (ih->ver_ihl & 0xf) * 4;
```

```
        if (ih->proto == 6)  { // TCP segment
            th = (tcp_header *)((u_char*)ih + ip_len);
            u_int tcp_len = ((th->off_res \gg 4) & 0xf) * 4;
            // Check segment type
            if (CHECK_FLAG(th->flags, SYN) && !CHECK_FLAG(th->flags,ACK)) {
                // SYN segment
            }
            else if (CHECK_FLAG(th->flags,SYN) &&
                     CHECK_FLAG(th->flags,ACK)) {
                // SYN ACK segment
            }
            else if (CHECK_FLAG(th->flags,FIN)) {
                // FIN segment
            }
            else if (COMP_ADDR(ih->saddr, server_addr)) {
                u_int tcp_data_len=(header->len-14- ip_len-tcp_len);
                DATA(th, tcp_data_len);
            }
            else if (COMP_ADDR(ih->saddr, client_addr) &&
                     CHECK_FLAG(th->flags, ACK)) {
                ACK(th);
            }
        }
    }
```

This function in turn takes the following arguments: a pointer to the .pcap file header related to the currently handled frame, and a pointer to the data block representing this frame. The code snippet related to the *packet_handler* function is shown in Listing 3. Detailed definitions of data structures can be found in *winpcap* examples provided e.g. in the *wpdpack* package [32].

As mentioned earlier, to emulate TCP congestion control the original Linux H-TCP code was used. The definitions of functions, parameters and data structures used here can be found in *tcp_htcp.c* Linux source code file. TCP connection state is stored in *tcp_sock* structure (only a subset of its fields is used in the emulation application). Two additional fields were added to this structure: RTO, storing the current value of the retransmission timer, and *t_rtt*, storing the last measured value of RTT.

Each time the data or acknowledgment segment is observed, an appropriate piece of the Linux kernel code is executed. When an acknowledgment segment with higher sequence number is observed, the RTT sample is calculated (the time difference between reception of acknowledgment segment and observation of data segment for the given sequence number at the monitoring point) and the *tcp_rtt_estimator*() function is executed to update the value of the RTO timer.

Listing 4: DATA segment handling

```
    void DATA(tcp_header * th, u_int tcp_data_length) {
        u_int seq = ntohl(th->seq);
        if (seq == tp.snd_nxt) { // New data
            tp.snd_nxt = seq + tcp_data_length;
            // Store transmission instant of the segment
            store_packet(tp.snd_nxt);
            tp.max_packets_out++;
            lastseq = tp.snd_nxt;
        }
        else { //  Data out of sequence
            if(FastRetransmit) {
                // Mark segment as retransmitted
                ...
            }
            else  if(packet_delay(seq+tcp_data_length)>tp.RTO) {
                // Mark segment as retransmitted
                ...
                slowstart();
            }
            else {
                // Mark segment as out of sequence
                ...
            }
        }
    }
```

The *DATA* function (Listing 4) emulates the sending of data by appropriate modification of the TCP process state. If the sequence number of the segment being sent is in sequence, the segment is marked as sent and not acknowledged (it is added to the list of unacknowledged segments), its sending instant is stored, and the value of *snd_nxt* variable is updated to the next proper sequence number.

If the segment sequence number is different than current *snd_nxt* value, the sequence is broken. This may be caused by various reasons: the segment may be a retransmitted one (and the original segment could have been lost before reaching the probe), the preceding segment was lost (so the *snd_nxt* variable was not updated), the sequence of segments was disturbed or the segment was duplicated. Unambiguous distinction between the above cases only on the basis of captured packet traces is difficult. In the paper the following simplification was used. If the TCP connection is in the fast retransmit phase, all segments with wrong sequence numbers are treated as retransmitted. Otherwise, if the time since sending the preceding segment (or the original copy of this segment, as long as the segment with this sequence number was already observed) is greater than the RTO, it it assumed that the segment was retransmitted because of timeout, and the TCP process enters the slow start mode. If none of the above conditions is met, it is assumed that the segment was transmitted out of sequence (duplicated segments are not detected).

Listing 5: ACK segment handling

```
    void ACK(tcp_header * th) {
        u_int ack = ntohl(th->ackseq); // Reading ACK sequence number
        if (ack == tp.snd_una) {
            // Duplicate ACK
            // Reading the advertised window value
            tp.rcv_wnd = ntohs(th->window_size) \ll tp.crws;
            dup_ack++;
            process_dup_ack(dup_ack); // Handling dup-ACKs
        }
        else if ((ack>tp.snd_una) || seq_number_wrap()) {
```

```
        // New ACK
        // Reading the advertised window value
        tp.rcv_wnd = ntohs(th->window_size) \ll tp.crws;
        // Window limit update
        tp.snd_una = ack;
        // RTT for acknowledged segment
        // 0 if segment was retransmitted
        long rtt_sample=retrieve_packet_rtt(tp.snd_una);
        if( rtt_sample!=0) {
            // Calculation of the RTO from Linux code
            tcp_rtt_estimator( &tp, rtt_sample);
            tp.RTO = (tp.srtt \gg 3) + tp.rttvar;
        }
        process_new_ack(); // New ACK handling
    }
    else { // Ignore
    }
}
```

The process of receiving ACK segments by the TCP server is emulated by the *ACK* function (see Listing 5). The procedure *process_dup_ack*, outlined in Listing 6, is called upon receiving of the segment with duplicated sequence number (*ack* == *rp.snd_una*). After reception of three duplicated acknowledgments, the TCP process enters the fast retransmit phase and the *htcp_recalc_ssthresh* function is executed to calculate the new *ssthr* value according to H-TCP algorithm. The *cwnd* takes value of *ssthr* + 3 and is increased by one after each consecutive reception of a duplicated acknowledgment.

Listing 6: Handling of duplicate ACKs

```
void  process_dup_ack(u_int dup_ack) {
    if (dup_ack == 3 && ! FastRetransmit) {
        // Set Fast Retransmit phase flag
        FastRetransmit = 1;
        // Calculate new ssthr using H-TCP algorithm from Linux code
        tp.snd_ssthresh = htcp_recalc_ssthresh(&tp, &ca);
        // Update cwnd value (TCP Reno)
        tp.snd_cwnd = tp.snd_ssthresh + 3;
    }
    else if (FastRetransmit) {
        // Window inflation in Fast Retransmit phase (TCP Reno)
        tp.snd_cwnd++;
    }
}
```

If the sequence number is greater than the one of the last received acknowledgment, the *process_new_ack* function is executed (Listing 7). If the TCP connection was in fast retransmit phase, the exit condition for this phase is checked. Next, the already acknowledged segments are removed from the list of segments awaiting acknowledgment, and the following H-TCP related functions are called: *measure_achieved_throughput* and *htcp_cong_avoid*. Other H-TCP module procedures (*htcp_param_update*, *htcp_alpha_update*, *htcp_beta_update* etc.) are called internally by the two functions mentioned above.

Listing 7: Handling of new ACKs

```
void process_new_ack() {
    dup_ack = 0;
    if (FastRetransmit) {
        // End fast retransmit phase
        tp.snd_cwnd = tp.snd_ssthresh;
        FastRetransmit = 0;
    }
    // Remove acknowledged segments from the list
    u_int count = remove_acked_packets(tp.snd_una);
    // Execute relevant H-TCP code
    measure_achieved_throughput(&tp, &ca, count, tp.t_rtt);
    tp.max_packets_out -= count;
    htcp_cong_avoid(tp.snd_una, count);
}
```

3.4 TCP Classification Algorithm

For automated detection whether the TCP connection is network, receiver or internally limited, we have implemented the algorithm outlined on Listing 8, that was originally proposed and described in detail in [16].

Listing 8: TCP limitation

```
total_sample_cnt++;
if (oustanding_data >= tp.rcv_wnd*gamma) {
    rwnd_cnt++;
}
else {
    diff_1 = abs(oustanding_data - tp.snd_cwnd);
    diff_2 = ...; // second estimator (e.g. AIMD)
    if (diff_1 <= diff_2) {
        if (oustanding_data >= tp.snd_cwnd*mi) {
            cwnd_cnt++;
        }
    }
    else {
        if (oustanding_data >= ...*mi) {
            ... // second estimator
        }
    }
}
```

The algorithm is executed each time a new value of the outstanding data is obtained (the data packet is observed). The new sample of outstanding data is compared with the *awnd* and *cwnd* estimators (note that we can emulate more then one congestion control algorithm at the same time) and the counters related to the closest limit are updated. The sensitivity of the above algorithm is controlled by two parameters *gamma* < 1 and *mi* > 0 (see [16] for more details).

In the original algorithm the TCP connection data was also divided into relatively small chunks, and each chunk was classified as greedy, non-greedy or receiver limited. The per-chunk classification allows obtaining a snapshot of the performance of the set of TCP connections at a given time scale (which was one of the goals of the original work presented in [16]). This paper is however mainly focused on the performance of a single TCP flow, and our aim was to detect the throughput limitation of the TCP connection treated as a whole. Therefore, we did not divide the TCP data into chunks, but rather used all measured data to categorize the TCP connection (in other words we treated the TCP flow as a single chunk and performed the TCP categorization over the whole data set). The division of the TCP data into chunks and its application to the root cause analysis of the TCP performance degradation was left for further study.

Finally, the classification algorithm, shown on Listing 9, is executed at the end of measurement period. Unlike in the original algorithm, we did not implement the AIMD (Additive-Increase/Multiplicative-Decrease) counters, because the TCP AIMD operations were not emulated in our approach. We have however modified the original classification algorithm to detect ambiguity in the detection of TCP connection type. The aim of the modification was to avoid false classification of the TCP connection as non-greedy in case when the TCP connection changes its type during the measurement interval (for example switches from being limited by *awnd* to being bounded by *cwnd*). The modification may be summarized as follows: if the sum of the $rwnd_{c}nt$ and $cwnd_{c}nt$ counters exceeds the assumed threshold (controlled by the *lambda* parameter; see Listing 9), the connection is marked as as *rwnd/cwnd limited* instead of just assuming that the TCP connection is non-greedy (which is not true in this case).

Listing 9: TCP classification algorithm

```
if (rwnd_cnt >= total_sample_cnt*lambda) {
    f_decision = "awnd_limited";
}
else {
    if (cwnd_cnt >= total_sample_cnt*lambda) {
        f_decision = "cwnd_limited";  // Greedy source
    }
    else {
        if (cwnd_cnt + rwnd_cnt >= total_sample_cnt*lambda) {
            f_decision = "rwnd/cwnd limited";
        }  else {
            f_decision = "non_greedy";
        }
    }
}
```

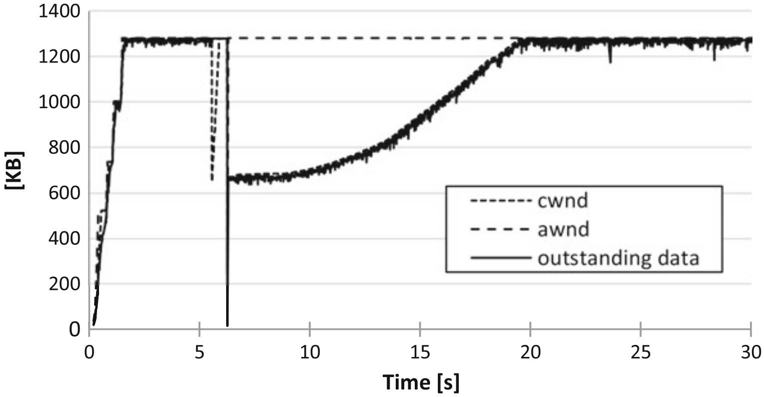


Fig. 1 H-TCP *cwnd* emulation

3.5 Validation of the TCP Emulation Algorithm

To validate the implemented TCP state tracking one can compare the outstanding data calculated from measurements with estimated values of the *cwnd* (as it can approximate *cwnd*, especially for greedy TCP sources). In the following experiment we downloaded a test file from the server to the mobile device using the measurement setup described in more detail in Sect. 3.8.

An example of TCP connection state emulation is shown in Figs. 1 and 2 for H-TCP-based source. Figure 1 shows the comparison of outstanding data with the value of *cwnd* estimated by emulation of the H-TCP congestion control algorithm using Linux kernel source code. The estimated *cwnd* almost exactly matches the amount of measured outstanding data. After slow start, the observed TCP connection enters the congestion avoidance phase where *cwnd* grows according to the H-TCP algorithm (finally reaching its maximum value). Upon segment loss the transition to fast retransmit phase is observed (the *cwnd* is reduced by half) and TCP reenters the congestion avoidance phase.

The estimation of RTO is shown in Fig. 2. This parameter is required to correctly detect timeout events that cause the TCP to enter the slow start phase. The proper validation of the RTO estimation accuracy would require direct measurement of this parameter in actual TCP stack. Unfortunately, such an option was not available in the network at the time when the measurements were conducted.

Analogous results are presented for the TCP Reno/NewReno based source. The accuracy of *cwnd* emulation is shown in Fig. 3, the RTO estimation in Fig. 4.

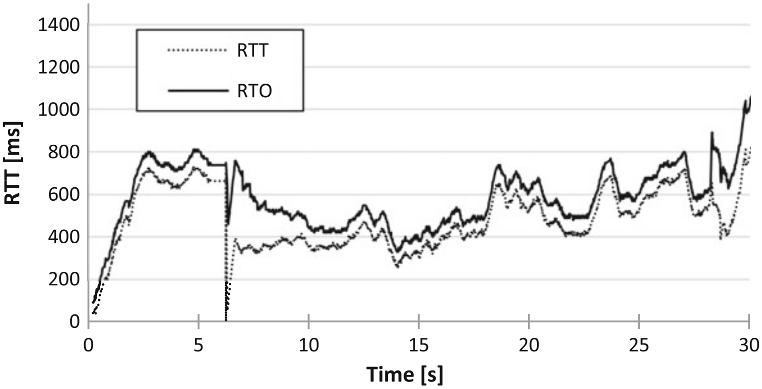


Fig. 2 H-TCP RTO estimation

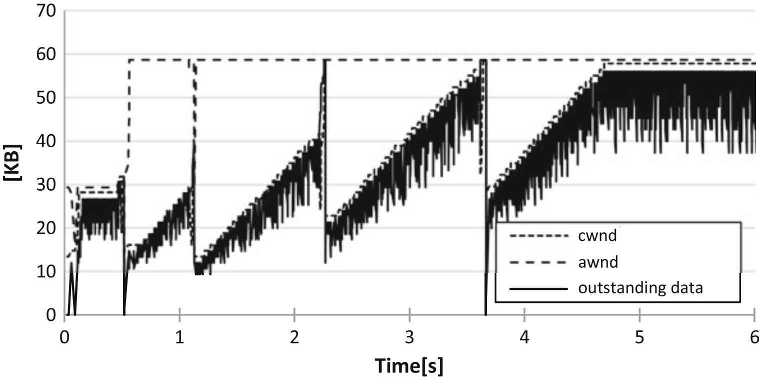


Fig. 3 TCP Reno *cwnd* emulation

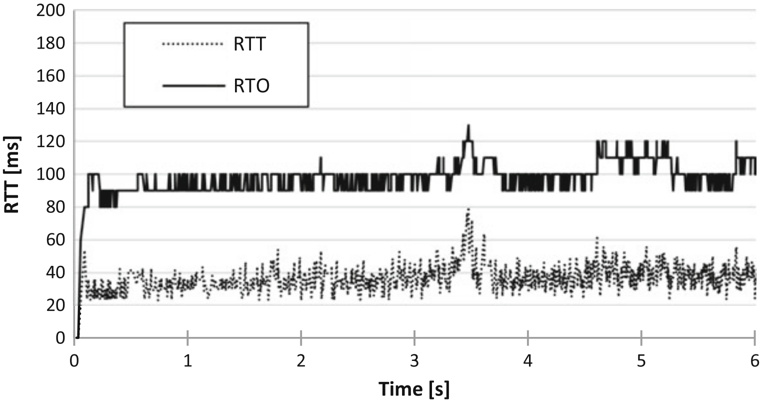


Fig. 4 TCP Reno RTO estimation

3.6 TCP Performance Metrics

Based on RFC 6349, the following metrics are recommended to test the TCP effectiveness.

- *TCP Throughput Ratio W*. This metric is calculated as a percentage ratio of achieved throughput TCP_{th} to the reference throughput C_{REF} and should approach 100 % for good connections.

$$W = \frac{TCP_{th}}{C_{REF}} * 100 \tag{5}$$

- *TCP transmission effectiveness E*. It is a percentage ratio of not retransmitted data to the total amount of data sent during the measurement period and should also approach 100 % for effective connections.

$$E = \frac{D - D_{RET}}{D} * 100 \tag{6}$$

  D_{RET} denotes the amount of data retransmitted during the measurement period.
- *Buffer Delay T*. To calculate this parameter one needs the reference delay RTT_{MIN} calculated beforehand from measurements taken when the network load is minimal. The *tcptrace* tool can be used for this task. Alternatively, RTT_{MIN} may be approximated by the minimal RTT observed during the actual measurement period. Denoting an average RTT observed within the measurement period as RTT_{AVG}, the Buffer Delay can be calculated as:

$$T = \frac{RTT_{AVG} - RTT_{MIN}}{RTT_{MIN}} * 100 \, . \tag{7}$$

  As the name implies, this parameter is related to buffer size in the network nodes and can be interpreted as a measure of buffer load imposed by the measured TCP connection (mostly related to the buffer at the bottleneck link). If we assume that buffer size B conforms to the following formula:

$$B > 2 * C_{REF} * RTT_{MIN} \, , \tag{8}$$

  then the Buffer Delay should be greater than 200 %.

3.7 Root Cause Analysis

For the root cause analysis, we use the emulation of the TCP sender state derived from passive measurements, and the metrics of TCP connection performance

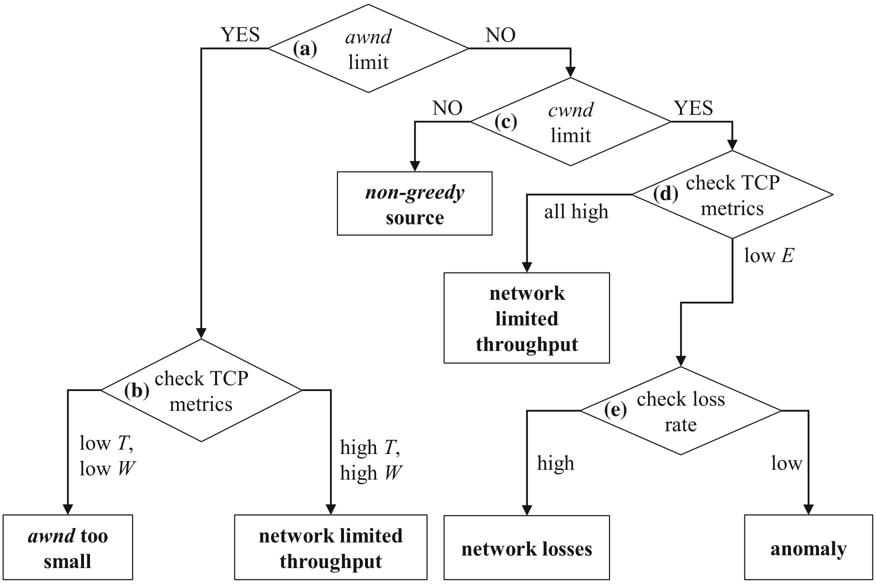


Fig. 5 General algorithm for root cause analysis

calculated on the base of conducted measurements. The general algorithm is depicted in Fig. 5.

In the proposed approach the first task is to check if the throughput is limited by *awnd* (decision block *a* in Fig. 5). This can be done by analyzing the behavior of the outstanding data using emulation of the TCP sender state. If *awnd* seems to be the limiting factor, then it is advised to check the TCP connection metrics (decision block *b*). Low *T* (low buffering) and low *W* (low bandwidth utilization) together with large *E* (lack of retransmissions) support the hypothesis that the advertised *awnd* value is indeed limiting the sender's performance. However, if the *T* and *W* are relatively large, the true limitation may lie in the network itself, and the *awnd* value reached by the sender is large enough for high connection effectiveness.

If neither *awnd* nor *cwnd* (estimated from emulation of TCP sender state) is the limiting factor (decision block *c*) then the achieved throughput results from the internal sender constraints (non-greedy source). Low value of Buffer Delay may additionally support this hypothesis.

If the *cwnd* imposes the limit on achieved throughput, it is advised to check TCP connection metrics (decision block *d*). High effectiveness of transmission together with large Buffer Delay confirm that TCP throughput is limited by a bottleneck link in the network. However, if the level of observed retransmissions is high (low *E*), then the reason behind low throughput may lie in excessive packet loss in the network (decision block *e*) resulting e.g. from faults, bad conditions on wireless access link etc. It has to be noted that TCP retransmissions occur naturally in result of congestion control algorithm continuous attempts to fit the transmission rate to the bottleneck

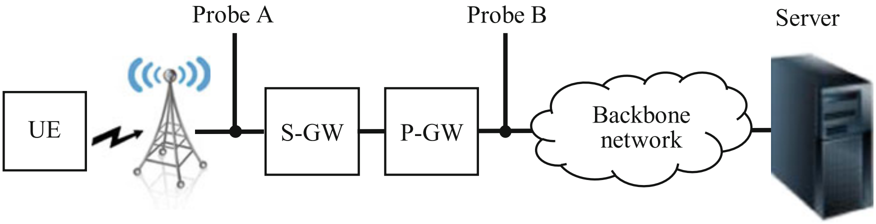


Fig. 6 Measurement setup

bandwidth, but the excessive level of retransmissions is suspicious and has to be checked further.

Finally, the case when transmission effectiveness is low but the packet loss is also low has to be treated as an anomaly that requires further investigation.

3.8 Validation of the Proposed Approach

The proposed approach was validated by conducting measurements in the real network. The measurement setup is depicted in Fig. 6.

Measurements were conducted in commercial 4G mobile network of Orange Poland with real user traffic served in the background. The setup consisted of the UNIX-based web server connected to the backbone network. The S-GW (Serving Gateway) and P-GW (PDN Gateway) are the subcomponents of the Evolved Packet Core of 3GPP's LTE wireless network.

The TCP traffic can be monitored at the server and/or at the mobile device (with *tcpdump*). Additionally, two hardware monitoring probes were installed in the mobile access network. The monitored TCP traffic was saved in *.pcap* format for further processing.

We ran a number of tests based on downloading files from the server to the mobile device (UE). Two types of experiments were carried out. In the first case the files were downloaded from the server in a greedy mode. The server was configured to transmit data with maximum possible rate so that the network available capacity was the only factor limiting the TCP throughput. In the second type of the experiment the socket buffer size at the receiver (the client) was limited below the bandwidth delay product of the network, which is approximately 100 KB (40 ms RTT * 20 Mbps C_{REF}).

In Figs. 7 and 8, the receiver's *awnd* and the amount of outstanding data were obtained directly from the collected TCP traces. The *cwnd* parameter was estimated using the algorithm described in the previous sections. As it can be seen from Fig. 7, the tested TCP connection begins in slow start and within few seconds the *awnd* and *cwnd* parameters reach their maximum sizes. This is possible due to large network buffers that can accommodate thousands of packets. After the next few seconds there

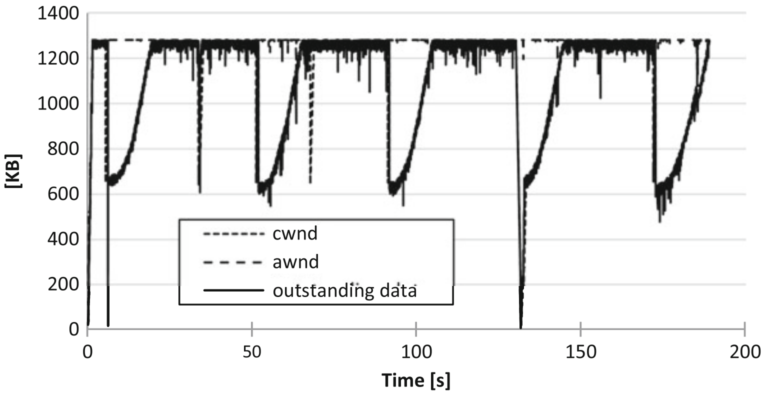


Fig. 7 Network limited TCP connection

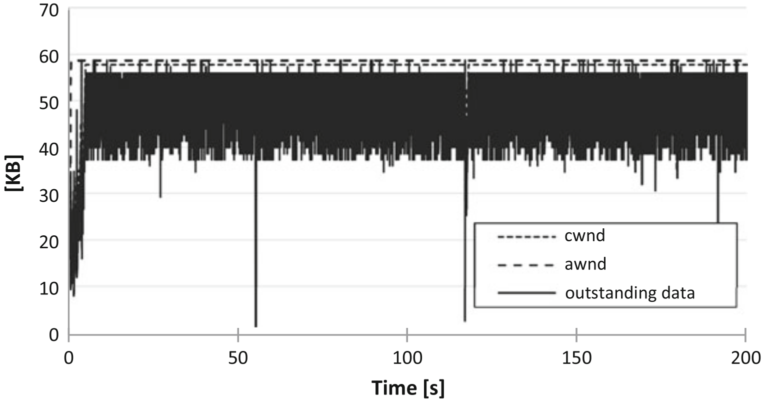


Fig. 8 Receiver limited TCP connection

is a packet drop (signaled by 3 Dup-ACK segments), TCP connection retransmits the lost segment and enters the congestion avoidance phase.

While in congestion avoidance the *cwnd* follows the H-TCP congestion control algorithms. At 130 s time instant we observe a retransmission due to the RTO expiration. TCP falls back to the slow start mode and after reaching *ssthr* (set to 0.5 of the *cwnd* before segment loss) it switches again to congestion avoidance phase. Notice that the estimated *cwnd* follows the amount of measured outstanding data very accurately.

Network limited TCP connection

```
min rtt [ms] = 23.5
avg rtt [ms] = 462
avg out data [B] = 1103400
avg awnd [B] = 1275910
avg cwnd [B] = 1114465
measured throughput [Mbps] = 15.8
outstanding data/rtt [Mbps] = 19
fast retransmits = 6
RTO expirations = 5
TCP efficiency (E) [%] = 99.98
buffer delay (T) [%] = 1866
```

According to the proposed TCP throughput measurement methodology, the tested TCP connection is clearly network limited. The outstanding data follows the *cwnd* closely while the TCP efficiency E is high which means no excessive packet loss inside the network. The buffer delay T is also very high indicating that TCP connection is transmitting a lot of data to the network (see text in the relevant frame). The network capacity C_{REF}, measured with the UDP protocol immediately before starting the test transfers (in the same conditions that influenced maximum throughput achievable in the location during the experiment), is about 18 Mbps. Therefore, the TCP connection in this case utilizes almost 90 % of available capacity (TCP throughput ratio W is also high).

Receiver limited TCP connection

```
min rtt [ms] = 17.5
avg rtt [ms] = 32.4
avg out data [B] = 50956
avg awnd [B] = 58616
avg cwnd [B] = 57615
measured throughput [Mbps] = 12.7
outstanding data/rtt [kbps] = 12.5
fast retransmits = 5
RTO expirations = 0
TCP efficiency (E) [%] = 99.99
buffer delay (T) [%] = 85
```

In the second experiment the socket buffer of the receiver was limited to about 60 KB. In this case the outstanding data follows the *awnd* (see Fig. 8) indicating that the TCP connection is limited by the client. This reasoning is also justified by low value of buffer delay T which is now below 100 %, meaning that TCP is not filling the network with data (see frame). The achieved TCP throughput is about 12.7 Mbps.

4 Conclusions

The paper presents the methodology for identifying the root cause of the TCP connection performance bottlenecks. We have used the Linux kernel source code to implement the algorithm for estimation of the internal TCP connection state. Such approach allows to infer the dynamics of the TCP congestion window which is otherwise unavailable from passive TCP monitoring. The knowledge of the internal TCP state (*cwnd*, *ssthr*, RTO) is essential to understanding the observed behavior of the TCP connection and allows identifying the source of the TCP throughput limitations. In our approach it is used together with the analysis of TCP performance metrics proposed in RFC 6349. Such combined approach, complimented with additional active measurements (probing available capacity, measuring bottleneck buffers) can be helpful in tracing down network problems related to TCP-based applications.

The practical implementation of the above approach requires upfront knowledge of the congestion control algorithm running on the sender side. In most use cases the algorithm can be identified. Our goal was to provide a tool for testing the performance of TCP connection in the scenario when the server and the client setups are at least known. Other cases, when the congestion control algorithm cannot be identified or its source code is unavailable are beyond the scope of this work. It is possible however to implement a set of various congestion control algorithms and to choose automatically the one that provides the best fit to the measured data. Still, the algorithm used by the sender has to be amongst the implemented ones, otherwise the TCP connection will be falsely classified as non-greedy. We may also implement other features of the TCP stack, not only the congestion control algorithms, to widen the possibilities of TCP connection performance analysis in the future.

References

1. Bąk, A., Gajowniczek, P., Zagożdżon, M.: Measurement methodology of TCP performance bottlenecks. In: Proceedings of the 2015 FEDCSiS, Annals of Computer Science and Information Systems, vol. 5, pp. 1149–1156 (2015). doi:10.15439/2015F284
2. Henderson, T., Floyd, S., Gurtov, A., Nishida, Y.: RFC 6582: The NewReno modification to TCP's fast recovery algorithm
3. Wei, D.X., Jin, C., Low, S.H., Hegde, S.: FAST TCP: motivation, architecture, algorithms, performance. IEEE/ACM Trans. Netw. **14**(6), 1246–1259 (2006). doi:10.1109/TNET.2006.886335
4. Xu, L., Harfoush, K., Rhee, I.: Binary increase congestion control for fast, long distance networks. Proc. IEEE INFOCOM **4**, 2514–2524 (2004)
5. Kelly, T.: Scalable TCP: improving performance in highspeed wide area networks. Comput. Commun. Rev. **32**(2) (2003)
6. Jamal, H., Sultan, K.: Performance analysis of TCP congestion control algorithms. Int. J. Comput. Comm. **2**(1) (2008)
7. Ha, S., Rhee, I., Xu, L.: CUBIC: a new TCP-friendly high-speed TCP variant. SIGOPS Oper. Syst. Rev. **42**(5), 64–74 (2008). doi:10.1145/1400097.1400105
8. Leith, D.J., Shorten, R.N., McCullagh, G.: Experimental evaluation of Cubic-TCP. In: Proceedings of PFLDnet (2008)

9. Armitage, G., Stewart, L., Welzl, M., Healy, J.: An independent H-TCP implementation under FreeBSD 7.0—Description and observed behaviour. ACM SIGCOMM Comput. Commun. Rev. **38**(3) (2008)
10. Leith, D., Shorten, R.: H-TCP: TCP for high-speed and long-distance networks. In: Proceedings of PFLDnet (2004)
11. Leith, D.J., Shorten, R.N., Lee, Y.: H-TCP: A framework for congestion control in high-speed and long-distance networks. In: Proceedings of PFLDnet (2005)
12. Floyd, S.: RFC 3649: Highspeed TCP for large congestion windows
13. Floyd, S.: RFC 3742: Limited slow-start for TCP with large congestion windows
14. Tan, K., Song, J., Zhang, Q., Sridharan, M.: A compound TCP approach for high-speed and long distance networks. Proc. INFOCOM **2006**, 1–12 (2006). doi:10.1109/INFOCOM.2006.188
15. Mascolo, S., Casetti, C., Gerla, M., Sanadidi, M.Y., Wang, R.: TCP Westwood: bandwidth estimation for enhanced transport over wireless links. Proc. ACM MOBICOM **2001**, 287–297 (2001)
16. Schiavone, M., Romirer-Maierhofer, P., Ricciato, F., Baiocchi, A.: Towards bottleneck identification in cellular networks via passive TCP monitoring. Lect. Notes Comput. Sci. **8487**, 72–85 (2014)
17. Constantine, B., Forget, G., Geib, R., Schrage, R.: RFC 6349: Framework for TCP throughput testing
18. Afanasyev, A., Tilley, N., Reiher, P., Kleinrock, L.: Host-to-Host congestion control for TCP. IEEE Commun. Surv. Tut. **12**(3), 304–342 (2010)
19. Prasad, R.S., Jain, M., Dovrolis, C.: Socket buffer auto-sizing for high-performance data transfers. J. Grid Comput. **1**(4), 361–376 (2003)
20. Semke, J., Mathis Mahdavi, M.: Automatic TCP buffer tuning computer communication review. ACM SIGCOMM **28**(4) (1998)
21. Gardner, M.K., Feng, W.-C., Fisk, M.: Dynamic right-sizing in FTP (drsFTP): enhancing grid performance in user-space. In: Proceedings of IEEE symposium on high-performance distributed computing (2002)
22. Mathis, M., Reddy, R.: Enabling high performance data transfers. http://www.psc.edu/networking/perftune.html (2003)
23. Fisk, M., Feng, W.: Dynamic right-sizing: TCP flow-control adaptation. In: Proceedings of the 14th Annual ACM/IEEE SC2001 Conference (2001)
24. Weigle, E., Feng, W.: A comparison of TCP automatic tuning techniques for distributed computing. In: Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing (2002)
25. Hirabaru, M.: Impact of bottleneck queue size on TCP protocols and its measurement. IEICE Trans. Commun. **E89-B**(1) (2006)
26. Wang, Yi, Guohan, Lu, Li, Xing: A study of internet packet reordering. Lect. Notes Comput. Sci. **3090**, 350–359 (2004)
27. Jaiswal, S., Iannaccone, G., Diot, C., Kurose, J., Towsley, D.: Measurement and classification of out-of-sequence packets in a tier-1 IP backbone. IEEE/ACM Trans. Netw. **15**(1), 54–66 (2007). doi:10.1109/TNET.2006.890117
28. Mathis, M., Heffner, J.: RFC 4821: packetization layer path MTU discovery
29. Hu, N., LI, L.M., Mao, Z., Steenkiste, P., Wang, J.: Locating internet bottlenecks: algorithms, measurements, and implications. SIGCOMM Comput. Commun. Rev. **34**(4), 41–54 (2004). doi:10.1145/1030194.1015474
30. Hu, N., Steenkiste, P.: Evaluation and characterization of available bandwidth probing techniques. IEEE J. Sel. Areas Commun. **21**(6) (2003)
31. Linux kernel 3.18. https://www.kernel.org/
32. wpdpack library. https://github.com/engina/uip-1.0-win/tree/master/wpdpack

Firefly-Based Universal Synchronization Algorithm in Wireless Sensor Network

Michal Chovanec, Jana Milanová, Lukáš Čechovič, Peter Šarafín, Martin Húdik and Michal Kochláň

Abstract Application areas of Wireless Sensor Network are spread in so many fields that the creation of an universal WSN is a big and challenging problem. But what unites almost all major solutions is the need for communication at some certain time slot or the need for communication between the nodes. Nodes usually have to make measurements, in some cases process the data and send them. Because of the mentioned reasons the synchronization of nodes in network, as well as use of real time communication in network is highly desirable. The detailed explanation of the node level of Firefly-based Universal Synchronization Algorithm and the network level is supported by the simulation results along with the implementation remarks. Then the real-time scheduler for real time operating system in small mobile robotics or WSN with strong modularity is described with features as advanced sleep modes and event driven programming. At the end we represent the basic concept of WSN with use of proposed methods on each node.

M. Chovanec · J. Milanová · L. Čechovič (✉) · P. Šarafín · M. Húdik · M. Kochláň
University of Žilina, Faculty of Management Science and Informatics,
Department of Technical Cybernetics, Univerzitná, 8215/1, 010 26 žilina, Slovakia
e-mail: lukas.cechovic@fri.uniza.sk

M. Chovanec
e-mail: michal.chovanec@fri.uniza.sk

J. Milanová
e-mail: jana.milanova@fri.uniza.sk

P. Šarafín
e-mail: peter.sarafin@fri.uniza.sk

M. Húdik
e-mail: martin.hudik@fri.uniza.sk

M. Kochláň
e-mail: michal.kochlan@fri.uniza.sk

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent
Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_5

1 Introduction

This work is extended version of “Universal Synchronization Algorithm for Wireless Sensor Networks—“FUSA Algorithm”” [1].

Wireless sensor networks (WSN) belong to the category of systems with a great measure of parallelism [2]. In order to effectively utilize the parallelism nature, to ensure real-time communication ability and to focus nodes’ computational power to application-oriented algorithms, it is crucial to synchronize the sensor nodes. In particular, WSN make extensive use of synchronized time in many contexts [2] (e.g. for data fusion, fusion of decisions, hybrid fusions, time division multiple access (TDMA) schedules, synchronized sleep periods, etc.). If the system can be considered to be synchronized, then it is very convenient to deploy simple RTOS running on each node where the individual threads fulfilling a different role. To avoid failure of the functionality of the whole WSN, it is clear that the thread which controls the communication must have the highest priority.

This paper describes a “Firefly-Based Universal Synchronization Algorithm (FUSA)” based on the fireflies synchronization process [1]. The described algorithm is versatile regardless the network topology. This means, hierarchical networks with master nodes that control the synchronization process as well as fully distributed homogeneous-sensor-type networks are usable for the proposed synchronization algorithm. In next section the priority scheduler usable for RTOS is described [3].

1.1 Application Area

WSN represents an application area of a great potential. With the proper algorithms for synchronization and data processing, the WSN can be used in many interesting areas, such as field of health-care [4–6], transportation [7], industry [8, 9], military [10] and many more. Property surveillance, object monitoring, environment monitoring (floods detection, fire detection, illegal logging detection, etc.) including protected areas monitoring is another significant application field of WSN [11, 12]. From functional point of view it is very important not to have only algorithms for monitoring—image recognition, voice recognition [13], but also the proper algorithm for increasing the function potential of the whole network [14].

In order to use the network for reaching the objective, in addition to demand for sufficient supply network nodes [15–17], it is necessary to assure that the nodes are able to communicate always with the necessary instant of time.

Synchronization algorithm demands are application specific. Typically, applications monitoring environment have the following characteristics [14]:

- Energy efficiency—time needed for synchronization, communication window length and active power modes should be are minimized;
- Scalability—usable for different number of nodes;
- Precision—the nodes are able to send the data in proper time;
- Synchronization time—the amount of time needed for synchronization should be as short as possible.

This paper assumes that the application field of the proposed synchronization algorithm is an application for monitoring forests in order to prevent the illegal wood logging situations. All simulations and application remarks are based on this assumption. However, this fact is not in contrary with the versatility of the proposed synchronization algorithm. The mentioned application field is for the illustrative and interpretation purposes.

1.2 Related Work for WSN Synchronization Algorithms

Since the WSN applications are specific, not every synchronization algorithm is suitable for WSN purposes. For example there could be critical limitations on node memory, computation power and communication capabilities. In this paper we mention synchronization algorithms whose nature allows WSN utilization.

Cristian's Algorithm [18] and *Berkeley Algorithm* [19] are considered as essential synchronization algorithms and we will not discuss them. Computer networks very often use *Network Time Protocol (NTP)* [20] for time synchronization. However, the standard computer networks do not suffer from limited energy constraints.

Well known and very often used algorithms in WSN are *Reference Broadcast Synchronization (RBS)* [21] and *Timing Synchronization Protocol for Sensor Network (TPSN)* [22]. In RBS, the master node called the beacon node is used for synchronization. The synchronization of the whole network is performed from the beacon node that sends the reference broadcast towards one-hop-distant nodes from beacon node. Large networks with many sensor and/or actuator nodes are usually divided into smaller virtual networks called clusters. TPSN synchronization algorithm works with synchronization master as well. This master node is elected by all nodes. As soon as the mater node is elected, the spanning tree of the network is created. The children nodes are being synchronized by their parent node. In case any change in the network topology happens (e.g. a node becomes unavailable) a new master has to be elected again.

A reference point and the construction of the network tree is also used in *Tree Structured Referencing Time Synchronization (TSRT)* [23] and *Lightweight Tree-based Synchronization (LTS)* [24].

Other class of synchronization algorithms that use master node for synchronization or the group of master nodes contains *Time Diffusion Synchronization (TDP)* [25] and *ETSP* [26], which use both TPSN and RBS methodology. They switch between the TPSN and RBS based on the threshold value. The hierarchical structure of WSN is also used in [27], where the big accuracy of synchronized clock can be achieved, but only in simulation environment. Some of the algorithms use conditional probability estimation as well [28–30].

2 Synchronization Algorithm—FUSA

The presented algorithm is based on the fireflies synchronization process [31–33] applied into network with fully distributed cooperation and coordination as well as in hierarchical model of WSN.

Since the communication subsystem in active mode consumes the significant, and in many cases the most, energy of all sensor node subsystems, minimizing active RF communication minimizes the energy consumption of the whole node too.

Each node periodically transmits synchronization packet (any data can be used). Let the basic time period be T. By using crystal-based clock, it is possible to set the period for each node precisely. However, each node starts at random time instant. This phenomena results in different timing phase start. Despite having crystal-based clock, small deviations in every clock source create deviations in time phase, thus getting all nodes out of the synchronization. This presents a problem and therefore synchronization algorithms are being used to suppress the unwanted effects.

Let's denote the phase for N nodes by $\phi_n \in< 0, 1 >$. Then, the maximum phase error in the network can be defined as

$$\phi_{max}(t) = max_n(\phi_n(t)) - min_n(\phi_n(t)). \tag{1}$$

This definition can be represented as network synchronization quality.

The network is fully synchronized when $\phi_{max}(t) = 0$. After this point, all nodes are allowed to switch to a sleep mode and they wake up only for a short period of time. In discrete time domain, the best way how to divide the time period T is to divide it into D count of the same parts (frames) T_d, i.e. $T = D \cdot T_d$. For the reason of simple practical implementation, in simulations, the dividing factor D has been equal to $D = 128$. The nodes have transmitted the data in $1s$ time period, which implies 128 Hz interrupt timer frequency. Each node is allowed to transmit data only in its time frame, while the rest of the time frames is assigned to other nodes. Node is expected to respect the radio silence usually by turning itself into the sleep mode.

Each node has a time counter which is incremented periodically in the timer interrupt routine. When it reaches the maximum, then the timer starts decrementing the counter value and at the same time the synchronization packet is transmitted. When the counter reaches the minimum value, the timer starts incrementing. The described process generates triangle wave output, which can be transformed into a phase represented as the sawtooth wave. As presented in [31], the triangle (sawtooth) wave is important and cannot be homogeneous (fe. time = time % 128 will not work).

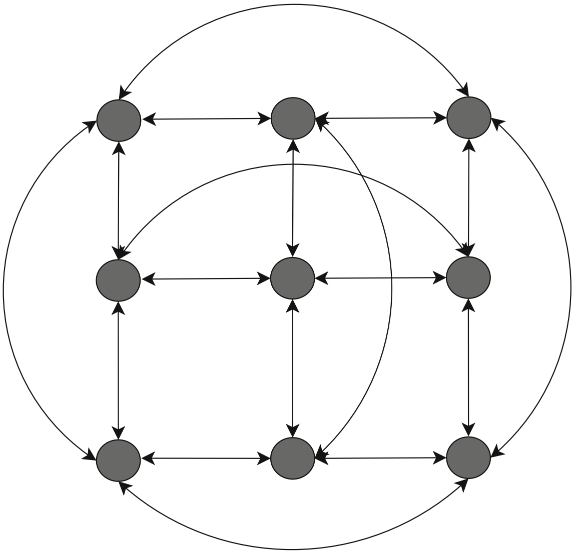


Fig. 1 Example of 3 × 3 grid network also called anuloid topology

Simulations were performed with *64* nodes, in a 8 × 8 static grid network topology, where each node can see only four neighboring nodes. This grid network topology is called an *anuloid grid* (example of 3 × 3 anuloid grid in Fig. 1). Figure 2 demonstrate the network without synchronization—random initial phases. Figure 2a represents phase values of first 8 nodes. Synchronization quality can be evaluated using Fig. 1 and is show in Fig. 2b. It is obvious that the maximum phase error oscillates around the maximum phase value (128) and the average phase is in the center (64).

The source code demonstrates the synchronization process in Ruby programming language:

```
# Algorithm: The node synchronization

def tick(fired)
@fired_tmp = false
if fired
@timer = TIMER_MAX
end
if (@state == 0)
if (@timer >= TIMER_MAX)
@timer-= 1
@state = 1
@fired_tmp = true
else
@timer+= 1
```

```
end
else
if (@timer <= 0)
@timer+= 1
@state = 0
else
@timer-= 1
end
end
```

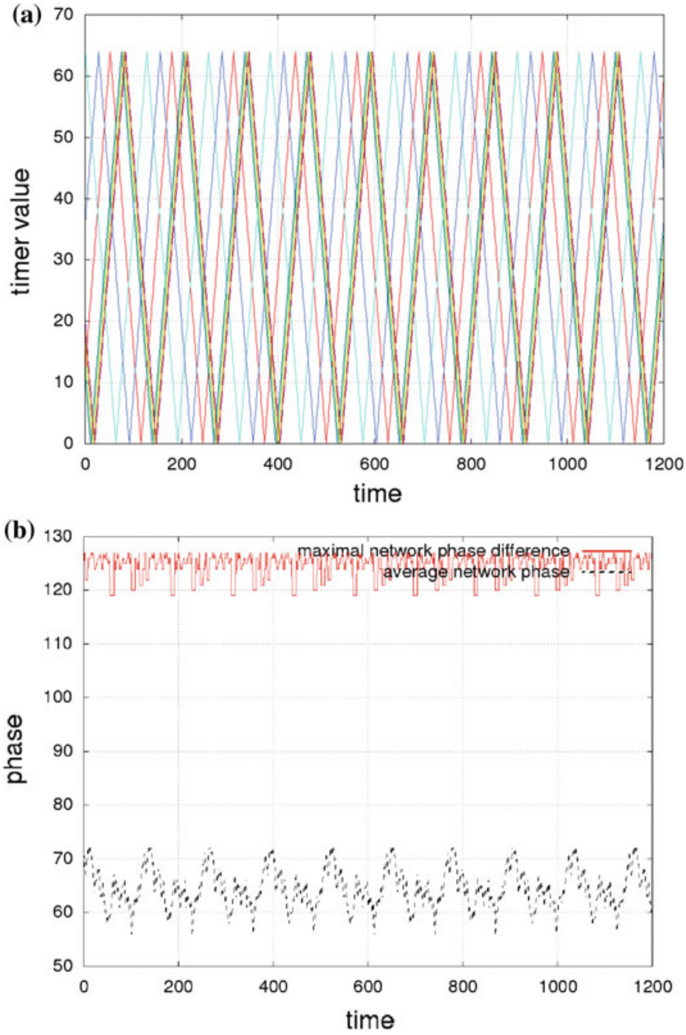


Fig. 2 **a** Phases of nodes without synchronization, **b** Maximum and average phase of the network without synchronization

```
end
```

The method *tick* is called periodically, in discrete time, on each node. The input parameter *fired* has two values:

$$fired = \begin{cases} true\ when & node[j][i+1].fired\ or \\ & node[j][i-1].fired\ or \\ & node[j+1][i].fired\ or \\ & node[j-1][i].fired \\ false & else. \end{cases}$$

When any of the neighboring nodes fires (timer is on the top), this node sets the timer counter to *TIMER_MAX* value. Depending on the state, in the next step, the timer is either decremented, or incremented and compared to the top value. If the counter reaches the maximum value, the node fires, and the state is switched.

```
# Algorithm: Network synchronization

for j in 0..@net.size-1
for i in 0..@net[j].size-1
if (@net[(j+1)%@net.size][i].get_fired)or
(@net[(j-1)%@net.size][i].get_fired)or
(@net[j][(i+1)%@net[j].size].get_fired)or
(@net[j][(i-1)%@net[j].size].get_fired)

fired = true
else
fired = false
end

@net[j][i].tick(fired)
end
end
```

The network synchronization must work in discrete time so *fired* flag is stored in *@fired_tmp* first. After all nodes call the method *tick*, the *fired* flags are updated.

Figure 3 demonstrates the network synchronization process. Figure 3a illustrates the phase of the nodes when synchronization process applies In Fig. 3b we can see fluent decreasing of the phase error, until it reaches 0.

Figures 4, 5, 6 and 7 represent the phase synchronization process and demonstrate the *synchronization wave* in the network in different iterations of the synchronization algorithm. As it can be seen, the waves with the increasing number of synchronization algorithm iterations slightly disappear.

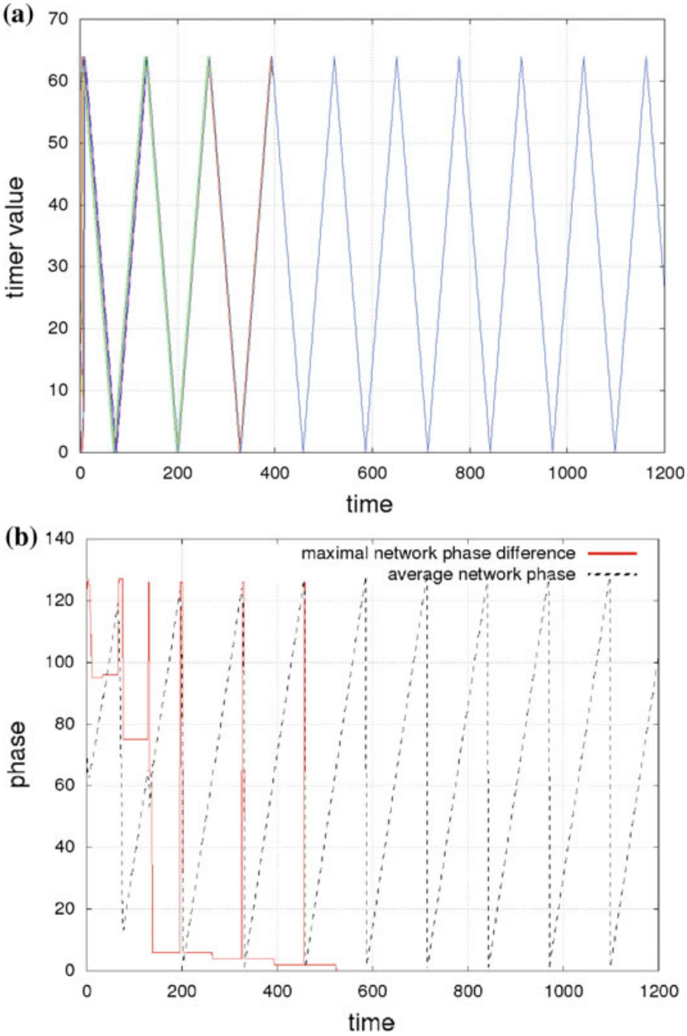


Fig. 3 **a** Phases of nodes with synchronization, **b** Maximum and average phase of the network with synchronization

By comparing Figs. 5 and 6 we can see the *synchronization wave*. When the network is fully synchronized, all phases are the same and we get a straight plane as illustrated in Fig. 7. During the simulations performed on the self-developed simulator, we found out that the proposed synchronization algorithm is fully functional despite of node failure. This has also no effect on the overall WSN operation.

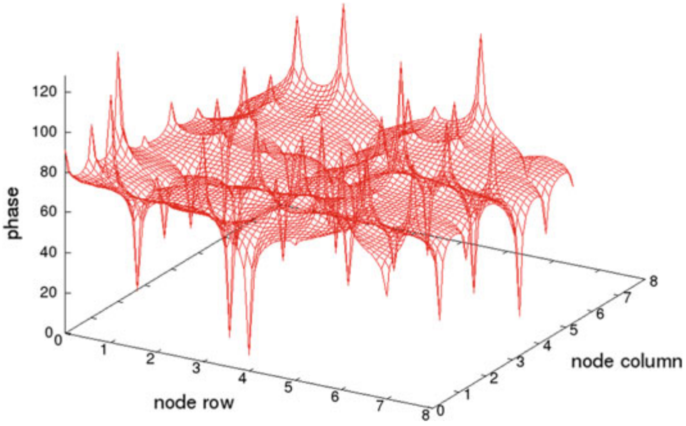


Fig. 4 Network initial phases, iteration 0

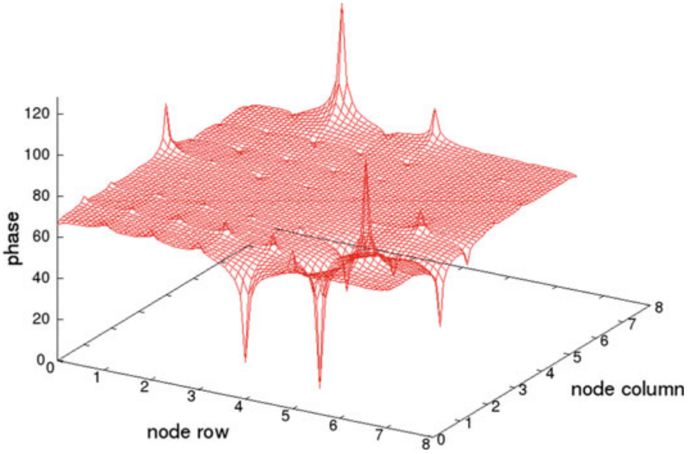


Fig. 5 Network synchronization, iteration 10

3 FUSA—Experimental Evaluation

The experimental evaluation of the proposed synchronization algorithm has been performed on wireless sensor nodes based on Texas Instruments MSP430 family microcontrollers (MSP430F2232, 8 KB FLASH, 512b RAM) (Fig. 8). The communication subsystem (RF) part is based on Texas Instruments transceiver CC1101—a low power transceiver operating at 868 MHz ISM band. Other sensor node subsystems built-in on-board are 3.3 V low-drop regulator, three-axis magnetometer, two LEDs and an UART peripheral for debugging. Further details about the nodes can be found in [34]. Red board in Fig. 8 was used as a power supply and for debugging purposes.

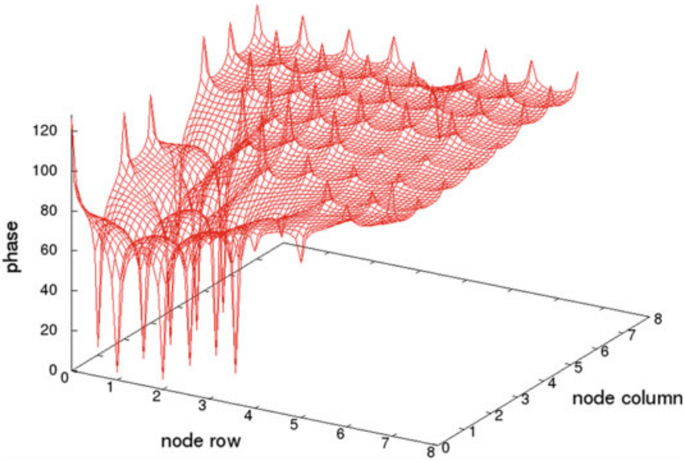


Fig. 6 Network synchronization, iteration 70

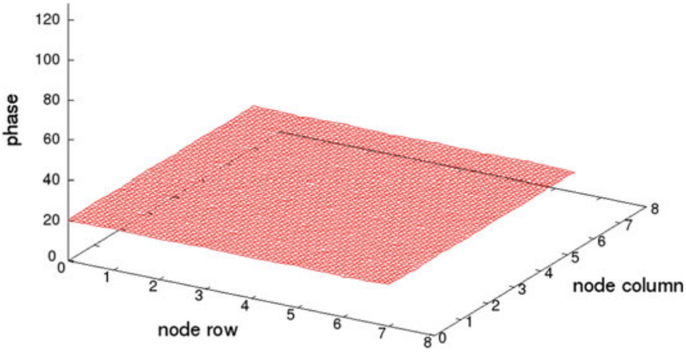


Fig. 7 Network synchronization, iteration 1180

After general input/output peripheral (GPIO) initialization and RF module initialization, the *timer*0*a* interrupt is set to 256 Hz periodic invoke (derived from external 32.768 kHz crystal-based clock source) [1].

The Algorithm *The node synchronization* described in the previous paragraphs is implemented in three program subroutines as follows:

- Synchronization with received packet (in GPIO pin interrupt routine);
- Periodical packet transmission and timer control (in timer interrupt routine);
- Higher-level network functions (in the main loop routine).

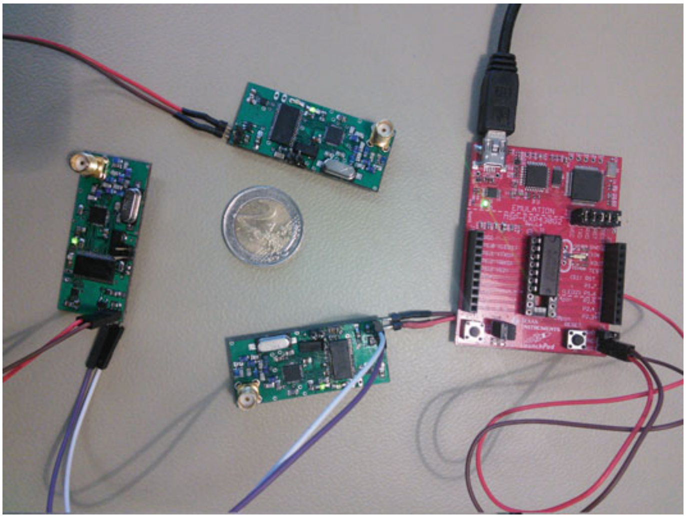


Fig. 8 Testing nodes photo

4 RTOS Scheduling Algorithm for WSN Applications

Real-time scheduler provides added value for embedded software development in the form of strong modularity, reusable code and rapid development [35]. Many embedded applications work without operating system—usually single purpose tasks or interrupt driven tasks. For more complex applications, real time operating system (RTOS) can provide better results when some common problems occur [36, 37]:

- Multiple sensors (or any inputs) reading;
- Multiple control loops with different sampling time;
- Communication (routing, resending);
- Power management;
- System modularity and extension possibilities;
- GUI running on background of the main process.

Main part of OS is the microkernel core. Preemptive multitasking with two options—round robin scheduling or time decrease priority scheduling is implemented. To compare different schedule algorithms (especially real-time processing), we first need to define error function. Consider set of threads as

$$t_i \in T(p, k, s, d, c), \tag{2}$$

where p represents thread priority (lower number—higher priority), k is the counter of thread priority current value, s stands for thread state (running, waiting, created), d states thread deadline time (set by user, usually in ms), c is thread running code (represented as Turing machine).

Let us define thread execution time function as $g(t_i)$ and error function as

$$e = \sum_{i=1}^{Tc} |d_i - g(t_i)|, \tag{3}$$

where Tc is threads count. This function represents the error, which corresponds with the difference between required deadline time and measured time of running thread. Using priorities we can define error as

$$e = \sum_{i=1}^{Tc} |d_i - g(t_i)| \frac{1}{p_i}, \tag{4}$$

where lower p_i means higher priority.

Consider that the faster execution of the thread is not an issue. This fact means that CPU spends remaining time waiting (executing other threads or sleeping). That can be found in (5) and (6).

$$e_i = \begin{cases} d_i - g(t_i) & if\ d_i < g(t_i) \\ 0 & else \end{cases} \tag{5}$$

$$e = \sum_{i=1}^{Tc} |e(t_i)| \frac{1}{p_i} \tag{6}$$

Threads with higher priority (smaller p_i) have bigger influence on the total error. To implement priorities, we define following structure for each thread:

```
/* Thread structure */

struct sThread{
u16 cnt, icnt;
u32 flag;
u32 *sp;
};
```

where *cnt* and *icnt* are counters used for priority scheduling, corresponding with p and k respectively in (2). When thread is created, p and k are set to *priority* value and remain constant (variability during execution is also possible, but not tested yet). Each nonzero k is decremented after each timer interrupt. Thread with smaller k is chosen for the next execution and its k is loaded back to p. Realization in C code is presented on following code.

```
/* Priority scheduler */

u32 i, min_i = 0;

/*find thread with minimum cnt*/
for (i = 0; i < THREADS_MAX_COUNT; i++)
{
if (__thread__[i].cnt <__thread__[min_i].cnt)
min_i=i;

/*decrement counters*/
if (__thread__[i].cnt != 0)
__thread__[i].cnt--;
}

__thread__[min_i].cnt = __thread__[min_i].icnt;
__current_thread__ = min_i;
```

For full function, other common functions like thread creating, waiting or setting into waiting state are implemented [3].

5 Concept of WSN with FUSA and RTOS Scheduling Algorithm

In real operation, the need is to get data from each node after some time of operation. Each node knows the period, when it should send and process measured data. But at the beginning it is not clear, where each period and each time window starts. If all nodes start to send data at the same time, the data can get lose in network, as the node is not receive data in the moment when the other node transmit them.

Because we assume, that each node need to process measured data and send them in right time period, we use proposed algorithm for scheduling the tasks. One process is used for data measuring, other for data processing and the last for data transmit. If the multistage structure of network is used, in operating system is the forth process—data receiving from node from other network level (Fig. 9) [14].

To ensure that the network will work correctly, the communication need to be done at proper time frame. That is why the data transmitting and data receiving have the highest priority in processes for process scheduler. Data measuring is other very important task of the network, where we need to set if some of measurements are more important than other. But in general, process of measurement has higher priority than measurement processing. It is obvious, that size of time window should be set according to time, that need node for processing data. If in some cases node does not have enough time, there are two possibilities according to used task of WSN.

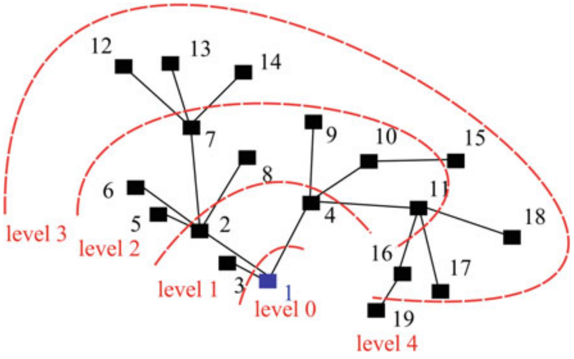


Fig. 9 WSN multistage structure

The node can send last processed measurement, or can send only measurement data with label, that data are not processed. Rest of the packet can be the synchronization information.

6 Conclusion

For the proper function of each WSN, the node synchronization is very important part of proposed solution. In this paper the algorithm for the synchronization was proposed, which is very easy for implementation. This algorithm is not only for hierarchical networks, it is universal and scalable. Functionality of synchronization algorithm was tested on mesh network up to 1024 nodes. Also algorithm of scheduling the processes in operating system was proposed. Combination of scheduling together with FUSA is powerful tool for handling various WSN applications with properties as robustness, easy expandability and fast software development.

Acknowledgements This publication is the result of the project implementation *"WSN pre monitoring a ochranu územia pohoria Malá Fatra"* with grant no. 2013et032 in grant program *"E-Talent 2013"* supported by Tatra banka Foundation Slovakia.

NADÁCIA TATRA BANKY

References

1. Chovanec, M., Púchyová, J., Húdik, M., Kochláň, M.: Universal synchronization algorithm for wireless sensor networks—"FUSA Algorithm". In: Federated Conference on Computer Science and Information Systems, pp. 1001–1007 (2014)
2. Elson, J., Römer, K.: Wireless sensor networks: a new regime for time synchronization. ACM SIGCOMM Comput. Commun. Rev. **33**(1), 149–154 (2003)

3. Chovanec, M., Šarafín, P.: Real-time schedule for mobile robotics and WSN aplications. In: Federated Conference on Computer Science and Information Systems, pp.1199–1202 (2015)
4. Kochláň, M., Hodoň, M., Púchyová, J.: Vital functions monitoring via sensor body area network with smartphone network coordinator. In: MEMSTECH 2013: Perspective Technologies and Methods in MEMS Design, pp. 143–147. Lviv Polytechnic Publishing House, Ukraine (2013)
5. Miček, J., Karpiš, O., Ševčík, P.: Body area network: analysis and application areas. Int. J. Eng. Res. Dev. (IJERD) **6**(8), 22–26 (2013)
6. Púchyová, J., Kochláň, M., Hodoň, M.: Development of Special Smartphone-Based Body Area Network: Energy Requirements. In: Federated Conference on Computer Science and Information Systems (FedCSIS). pp. 915–920. Kraków, Poland (2013)
7. Kochláň, M., Miček, J.: Indoor propagation of 2.4GHz radio signal: propagation models and experimental results. In: 10th International Conference on Digital Technologies, pp. 125–129 (2014)
8. Flamminia, A., Ferraria, P., Mariolia, D., Sisinnia, E., Taronib, A.: Wired and wireless sensor networks for industrial applications. In: 2nd IEEE international workshop on advances in sensors and interfaces, vol. 40, no. 9, pp. 1322–1336 (2009)
9. Gungor, V.C., Hancke., G.P.: Industrial wireles sensor networks: challenges, design principles and technical approaches. IEEE Trans. Ind. Electron. **56**(10) (2009)
10. Karpiš, O., Miček, J.: Sniper localization using WSN. In: International Conference on Military Technologies, pp. 1063–1068. Brno, Czech Republic (2011)
11. Karpiš, O., Juríček, J., Miček, J.: Application of wireless sensor networks for road monitoring. in: 10th IFAC Workshop on Programmable Devices and Embedded Systems, vol. 3, pp. 611–617 (2013)
12. Miček, J., Kapitulík, J.: WSN sensor node for protected area monitoring. In: Federated Conference on Computer Science and Information Systems, pp. 803–807 (2012)
13. Hyben, M., Hodoň, M.: Low-cost command-recognition device. Przegl. Teleinformatyczny **1**(3), 19–28 (2013)
14. Papán, J., Jurečka, Púchyová, J.: WSN for forest monitoring to prevent illegal logging. In: Proceedings of the Federated Conference on Computer Science and Information Systems, pp. 809–812. Wroclaw, Poland (2012)
15. Hofmann, A., Laqua, A., Husar, P.: Piezoelectric based energy management system for powering intelligent implants and prostheses. In: Biomedizinische Technik/Biomedical Engineering, pp. 263–266 (2012)
16. Kochláň, M., Miček, J., Hyben, M.: Wireless sensor network energy harvesting: radio frequency harvesting case study. In: Intelligent Transportation Systems 2013 (ITS 2013), pp. 93–97 (2013)
17. Laqua, D., Husar, P.: Intelligent power management enables autonomous power supply of sensor systems for modern prostheses. In: Biomedizinische Technik/Biomedical Engineering, pp. 247–250 (2012)
18. Cristian, F.: Probabilistic clock synchronization. Distrib. Comput. **3**, 146–158 (1989)
19. Gusella, R., Zatti, S.: The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD. In: IEEE Transactions on Software Engineering, pp. 847–853 (1989)
20. Mills, D.L.: Internet time synchronization: the network time protocol. IEEE Trans. Commun. **COM-39**(10), 1482–1493 (1991)
21. Jeremy, E., Lewis, G., Deborah, E.: Fine-Grained network time synchronization using reference broadcasts. In: Fifth Symposium on Operating Systems Design and Implementation (2002)
22. Ganeriwal, S., Ram, K., Srivastava, M.B.: Timing-Sync protocol for sensor networks. In: First ACM Conference on Embedded Networked Sensor Systems (2003)
23. Rahamatkar, A., Agarwal, A.: A reference based, tree structured time synchronization approach and its analysis in WSN. Int. J. Ad Hoc Sensor Ubiquitous Comput. **2**, 20–31 (2011)
24. Greunen, J.V., Rabaey, J.: Lightweight time synchronization for sensor networks. In: Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications. San Diego, CA (2003)

25. Su, W., Akyildiz, I.F.: Time-diffusion synchronization protocols for sensor networks. IEEE/ACM Trans. Netw. **13**, 384–397 (2005)
26. Shahzad, K., Ali, A., Gohar, N.D.: ETSP: an energy-efficient time synchronization protocol for wireless sensor networks. In: 22nd International Conference on Advanced Information Networking and Applications—Workshops, pp. 971–976 (2008)
27. Albu, R., Labit, Y., Thierry, G., Pascal, B.: An energy-efficient clock synchronization protocol for wireless sensor networks. In: Wireless Days, pp. 1–5 (2010)
28. Bo, C., Enqing, D., Xiaoyang, L., Dejing, Z., Jiaren, W.: A time synchronization algorithm based on bimodal clock frequency estimation. In: 18th Asia-Pacific Conference on Communications, pp. 75–78 (2012)
29. Kim, J., Lee, J., Serpedin, E., Qaraqe, K.: A robust clock synchronization algorithm for wireless sensor networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3512–3515 (2011)
30. Sage, A.P., Husa, G.W.: Algorithms for sequential adaptive estimation of prior statistics. In: IEEE Symposium on Adaptive Processes Decision and Control, pp. 760–769 (1969)
31. Mirollo, R.E., Strogatz, S.H.: Synchronization of pulse-coupled biological oscillators. SIAM J. Appl. Math. **50**, 1645–1662 (1990)
32. Tyrrell, A., Auer, G., Bettstetter, C.: Fireflies as Role Models for Synchronization in Ad Hoc Networks, pp. 1–7. Bio-Inspired Models of Network, Information and Computing Systems (2006)
33. Tyrrell, A., Auer, G., Bettstetter, C.: Firefly synchronization in ad hoc networks. In: 3rd MiNEMA Workshop. Lueven, Belgium (2006)
34. Hodoň, M., Chovanec, M., Hyben, M.: Intelligent traffic-safety mirror. Studia Informatica Universalis **11**(1), 87–101 (2013)
35. Hentzen, W.: The Software Developer's Guide, 3rd edn. (2002)
36. Gajaweera, N.: Wireless sensor networks. http://www.ent.mrt.ac.lk/dialog/documents/ERU-2-wsn.ppt
37. Stankovic, J.A., Wood, A.D., He, T.: Realistic applications for wireless sensor networks. http://www.ent.mrt.ac.lk/dialog/documents/ERU-2-wsn.ppt

Comparison of Various Marker Localization Methods

Peter Vestenický, Tomáš Mravec and Martin Vestenický

Abstract The presented paper is focused on analysis of two methods of marker localization. The markers are passive RFID transponders (without or with identification chip) consisting of tuned LC circuit and being used to mark and trace underground networks such as cables and pipes. Localization of the marker is based on evaluation of signal amplitude received from the excited marker, i.e. it is RSSI based localization method. The excitation of marker can be periodically repeated or continuous. In the first case the localization process consists of two stages—excitation and receiving of marker damped oscillations, in the second case the amplitude of continuously generated excitation signal is decreased by vicinity of the marker. Localization principle based on continuous marker excitation is analysed for serial or parallel resonant circuit of locator antenna. Both localization principles are mathematically analysed by modelling of their circuits using differential equations. The results of analysis are used to compare all methods and to evaluate their suitability for practical utilization. Whereas the markers have various working frequencies the analyses were done for all of them.

Keywords RFID · Localization · Mutual inductance · Marker

P. Vestenický · T. Mravec (✉)
Faculty of Electrical Engineering, Department of Control
and Information Systems, University of Žilina, Univerzitná 8215/1,
01026 Žilina, Slovakia
e-mail: tomas.mravec@fel.uniza.sk

P. Vestenický
e-mail: peter.vestenicky@fel.uniza.sk

M. Vestenický
Faculty of Electrical Engineering, Department of Telecommunications
and Multimedia, University of Žilina, Univerzitná 8215/1, 01026 Žilina, Slovakia
e-mail: martin.vestenicky@fel.uniza.sk

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent
Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_6

1 Introduction

Inductively coupled RFID (Radio Frequency Identification) systems [1] are now being widely used in many industrial applications. For example, the marking of goods by RFID technology enables the traceability of goods which is helpful to control the whole logistic chain from production to sale. In addition to these applications, the RFID transponders are being used for marking of underground facilities location. Such RFID transponders are called "markers". The marker is a passive RFID transponder consisting of a tuned LC circuit without identification chip (1-bit) or with identification chip tuned on low frequency in 77−170 kHz band. This work is an extended version of the paper published in [2].

For localization of some older underground facilities (cables, pipes etc.) a signal can be injected into their continual metal conductor and this signal can then be received on terrain surface and the cable or pipe can be traced. Today's underground facilities are mostly constructed from plastic material so this simple localization and tracing method cannot be used, therefore in this case the marking of underground objects by RFID markers is the only useable method.

The unknown position of marker under the terrain surface can be estimated by a localization device (locator). Moreover, the depth of marker can be estimated by RSSI (Received Signal Strength Indication) similarly as described in [3, 4]. In [5] the authors describe marker localization methods based on marker damped oscillations and on continuous generating of magnetic field. This work extends the analyses of RSSI based marker localization methods. The method based on damped oscillations of marker is extended by introduction of separate damping and sensing resistors and the analysis of method based on continuous marker excitation is done by applying differential equations instead of algebraic equations to analyse the transient phenomena in sensing circuit.

2 Related Works

The localization of moving underground objects (for example animals) based on inductive coupling is described in [6]. This application assumes the use of sensor network consisting of transmitting coils fixed on terrain surface and the moving underground object equipped with receiver collects data transmitted from these coils.

Indoor localization based on triaxial coils applied in both transmitter and receiver with very low working frequency 2.5 kHz is published in [7]. Low frequency magnetic field is suitable for underground localization purposes, too, because it is not affected by ground properties.

Another approach is presented in [8]. This work assumes localization based on UHF RFID tags in mining industry, but it is performed in mining tunnels and localization from the terrain surface is not assumed because the UHF signals do not propagate through layer of ground.

The mathematical analyses of the marker localization methods were presented in [2] where only serial connection of antenna tuning circuit was analysed. This paper extends the analyses by analysing of parallel antenna tuning circuit and by comparison of both antenna arrangement sensitivity to the marker vicinity. The presented results are numerically solved for all of the standard working frequencies of the markers. Moreover, in [2] an equation for mutual inductance calculation taken from [9] was used. This equation seems to be inaccurate therefore new equation for mutual inductance calculation was used and all numerical results were recalculated using the Eq. (2).

3 Mutual Inductance

An important quantity in the models presented in next chapters is the mutual inductance M between marker and locator coils. The mutual inductance can be calculated by many methods which have high demands on calculations [10]. These methods solve the mutual inductance calculations by the elliptic integrals of the first or second kind or by expressing of these integrals via infinite series.

The magnetic field intensity H generated by the first circular coil on its axis is given by the following formula [9]

$$H = \frac{N_R \cdot I \cdot r_R^2}{2 \cdot (x^2 + r_R^2)^{\frac{3}{2}}}. \tag{1}$$

Then, assuming that the second circular coil is placed on the axis of the first coil at the distance x and the magnetic field generated by the first coil in the place of the second coil is homogenous, the mutual inductance M can be calculated from the next simple equation

$$M = \frac{\pi \mu_0}{2} \frac{N_R N_T r_R^2 r_T^2 \cos \theta}{(r_R^2 + x^2)^{\frac{3}{2}}} \tag{2}$$

where N_R is number of turns of the first coil (RFID locator antenna), N_T is number of turns of the second coil (marker coil), I is current flowing through coil, r_R is radius of locator antenna coil, r_T is radius of marker coil, x is distance between the locator antenna and the marker and θ is the angle between coil of locator antenna and marker coil (if $\theta = 0°$ then coils are parallel). Note that the Eq. (2) is different from the equation used for analyses in [2] because the resulting mutual inductance M given by (2) is more accurate.

4 Mathematical Model Based on Damped Oscillation

This principle of localization assumes that the localization device periodically excites the marker LC circuit and in the pauses between excitation periods the response from marker in form of its damped oscillations is received. The simpler model with one resistor was analysed in [5]. In this chapter a more complex model with separate damping and sensing resistors will be analysed. The first resistor R_D is used for fast damping of locator $L_R C_R$ tuned circuit oscillations and its value can be calculated from the Eq. (3). The second resistor R_M is used for current sensing in the receiving stage of localization. The model is shown in Fig. 1.

$$R_D = 2\sqrt{\frac{L_R}{C_R}} - R_R \tag{3}$$

For this model the next system of equations (4) can be derived:

$$\begin{aligned} & L_R \frac{di_1(t)}{dt} + [(1 - MER(t))MOD(t)R_D + MER(t)MOD(t)R_D + R_R]i_1(t) \\ & + \frac{1}{C_R}\int_0^t i_1(\tau)d\tau - M\frac{di_2(t)}{dt} = (1 - MOD(t))u_1(t) \\ & L_T\frac{di_2(t)}{dt} + R_T i_2(t) + \frac{1}{C_T}\int_0^t i_2(\tau)d\tau - M\frac{di_1(t)}{dt} = 0 \end{aligned} \tag{4}$$

The modulation function $MOD(t)$ is given by Eq. (5) and the sensing (measuring) resistor R_M is switched by function $MER(t)$ given by the Eq. (6).

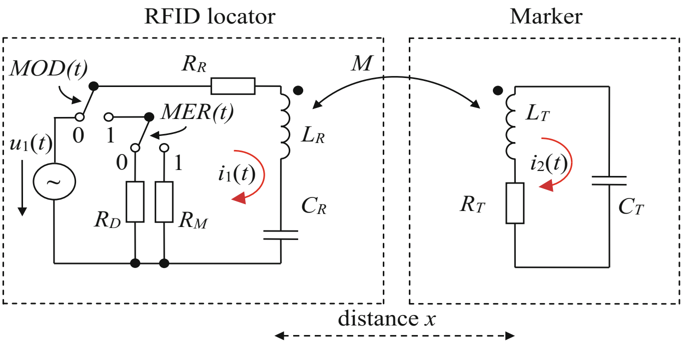


Fig. 1 Model of localization with separate damping and measuring resistors

$$MOD(t) = \frac{\mathrm{Sign}\left(-\sin\frac{2\pi ft}{250}\right)+1}{2} \tag{5}$$

$$MER(t) = \frac{\mathrm{Sign}\left(-\sin\frac{2\pi f(t-\Delta T)}{250}\right)+1}{2} \tag{6}$$

i.e. it is binary square signal with a frequency 250 times lower than the frequency of excitation signal source and shifted (delayed) by ΔT in time against the modulation function $MOD(t)$. During the small delay ΔT the oscillations of locator antenna tuned circuit has to be damped. The excitation signal source is assumed harmonic, i.e.

$$u_1(t) = U_1\ \sin(2\pi ft) \tag{7}$$

The system of integrodifferential equations (4) was numerically solved after its conversion to the 1st order system of differential equations (8) by substitution $x_1(t) = i_1(t)$, $x_2(t) = \mathrm{d}i_1(t)/\mathrm{d}t$, $x_3(t) = i_2(t)$, $x_4(t) = \mathrm{d}i_2(t)/\mathrm{d}t$, $\omega = 2\pi f$. Then we get:

$$\begin{aligned}
\frac{dx_1(t)}{dt} &= x_2(t)\\
\frac{dx_2(t)}{dt} &= -a_1x_1(t)\ + a_2(1-MOD(t))\omega U_1\cos(\omega t) - a_3x_3(t) - a_4x_4(t)\\
&\quad - a_2[R_R + [1-MER(t)]MOD(t)R_D + MER(t)MOD(t)R_M]x_2(t)\\
\frac{dx_3(t)}{dt} &= x_4(t)\\
\frac{dx_4(t)}{dt} &= -b_1x_1(t) + b_2(1-MOD(t))\omega U_1\cos(\omega t) - b_3x_3(t) - b_4x_4(t)\\
&\quad - b_2[R_R + [1-MER(t)]MOD(t)R_D + MER(t)MOD(t)R_M]x_2(t)
\end{aligned} \tag{8}$$

where the individual coefficients a_1, a_2, a_3, a_4 and b_1, b_2, b_3, b_4 are given by:

$$\begin{aligned}
a_1 &= \frac{L_T}{C_R(L_RL_T-M^2)} & b_1 &= \frac{M}{C_R(L_RL_T-M^2)}\\
a_2 &= \frac{L_T}{L_RL_T-M^2} & b_2 &= \frac{M}{L_RL_T-M^2}\\
a_3 &= \frac{M}{C_T(L_RL_T-M^2)} & b_3 &= \frac{L_R}{C_T(L_RL_T-M^2)}\\
a_4 &= \frac{MR_T}{L_RL_T-M^2} & b_4 &= \frac{L_RR_T}{L_RL_T-M^2}
\end{aligned} \tag{9}$$

The used values of R_R, L_R, C_R, N_R and R_T, L_T, C_T, N_T are listed in the Table 1.

Table 1 Values of components used in numerical calculations

R_R	L_R	C_R	N_R	R_T	L_T	C_T	N_T
15.7 Ω	1 mH	1.411 nF	23	7.85 Ω	1 mH	1.411 nF	38

Note that the values of L_R, C_R and L_T, C_T were selected so that the corresponding resonant frequencies are $f_R = f_T = 134$ kHz.

The radiuses of both coils used in numerical calculations are $r_R = r_T = 0.1$ m, distance between them is $x = 0.4$ m and the angle $\theta = 0°$. Corresponding mutual inductance M is then calculated from (2). Amplitude of the excitation signal is $U_1 = 10$ V, its frequency f is 134 kHz.

The damping resistor $R_D = 1668\ \Omega$ calculated from (3) in this case ensures the minimum time of $L_R C_R$ transient response. The used values of the sensing resistor are $R_M = 100\ \Omega$ for comparison with the results obtained in [5] and $R_M = 1\ \Omega$ to maximize the current response $i_1(t)$ from marker.

The time course of the current $i_1(t)$ for model of marker localization with two resistors is shown in Figs. 2 and 3 for $R_D = 1668\ \Omega$, $R_M = 100\ \Omega$ and $R_M = 1\ \Omega$, respectively.

Moreover, the dependence of maximum current amplitude I_{1_Max} on the distance x was calculated for this model and it is shown in Fig. 4. Note that the current maximum was calculated in the time interval after the transients of the excitation current decay. For comparison of the presented analysis results and the results from simplified model calculated in [5] the case when $R_D = R_M$ was calculated, too. This case is identical with the simplified model and the comparison is shown in Fig. 4.

The mathematical model given by Eqs. (4)–(9) was solved not only for one working frequency (134 kHz) but also for other marker working frequencies ($f = f_R = f_T$) in the range from 83–169.8 kHz [5] by varying the capacitances C_R, C_T and the damping resistor R_D for comparison purposes. The resulting dependencies of maximum amplitudes of current responses are shown in Fig. 5. This figure shows that differences exist among various working frequencies of markers and that the response from the marker with greater working frequency has greater

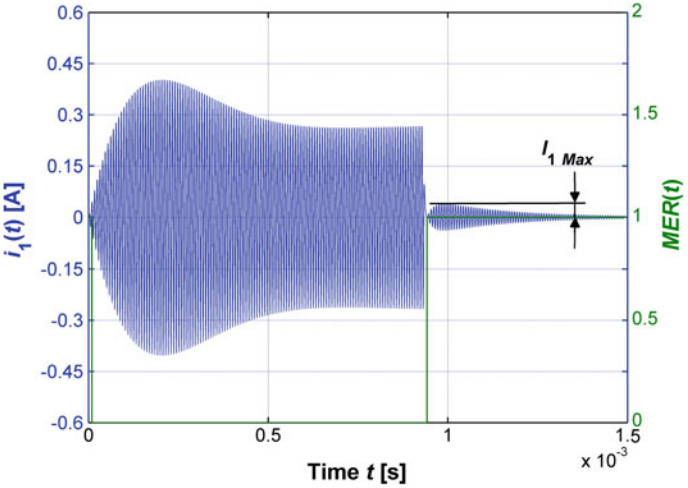


Fig. 2 Time course of the current $i_1(t)$ for $R_D = 1668\ \Omega$ and $R_M = 100\ \Omega$

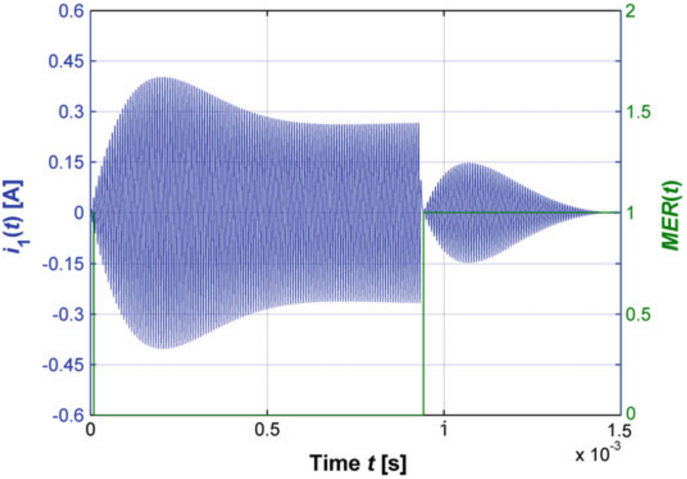


Fig. 3 Time course of the current $i_1(t)$ for $R_D = 1668\ \Omega$ and $R_M = 1\ \Omega$

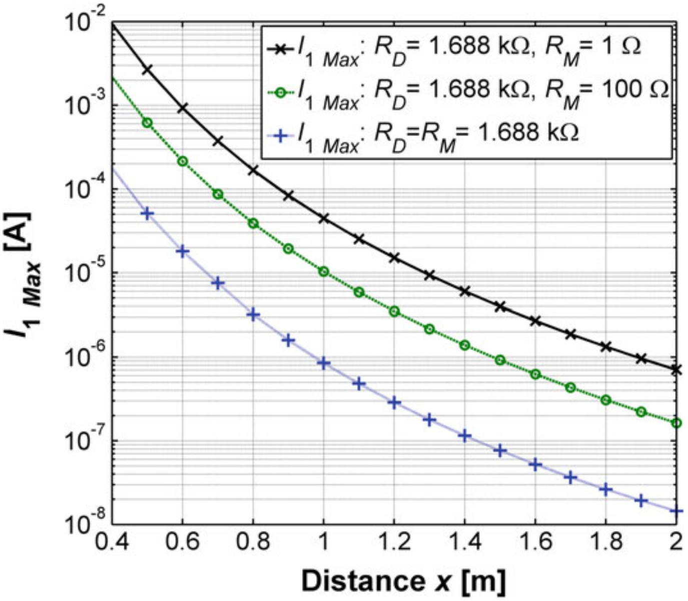


Fig. 4 Maximum amplitude of current response from marker for damped oscillation model, $f = f_R = f_T = 134$ kHz

amplitude. In practice, this theoretical advantage of localization of marker with greater working frequencies is partially limited because the parasitic phenomena (especially a skin effect and a self-resonance of the coil L_R) partially decrease the amplitude of current $i_1(t)$.

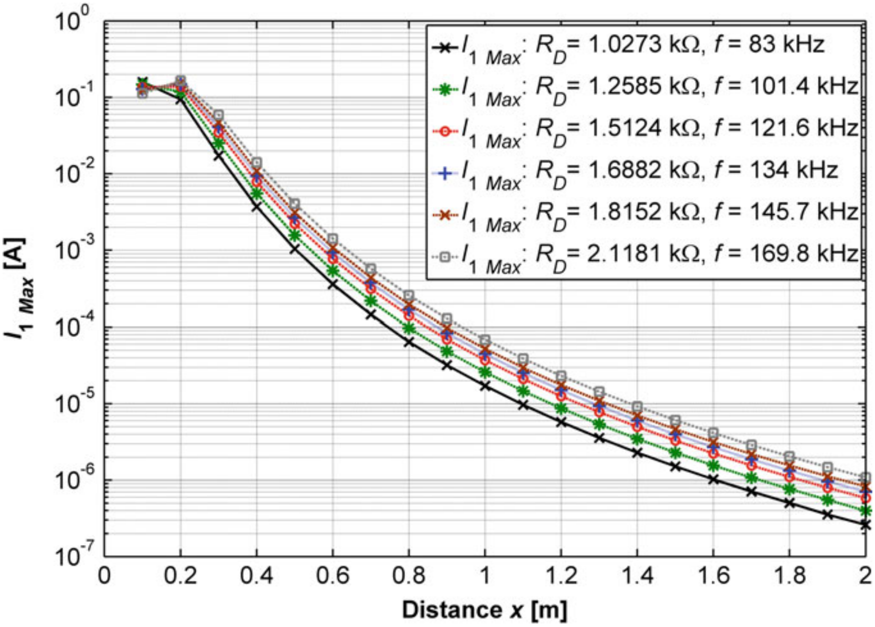
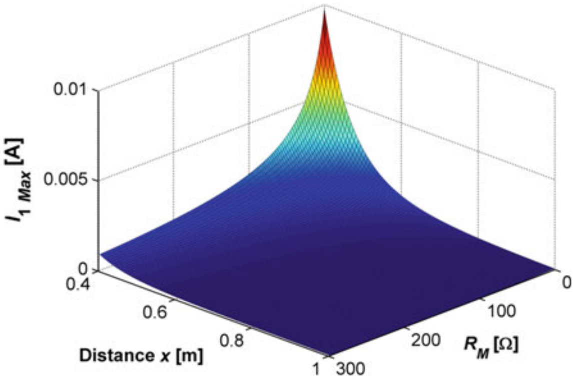


Fig. 5 Maximum amplitude of current response from marker for damped oscillation model calculated for various working frequencies, $R_M = 1\ \Omega$

The sensing (measuring) resistor R_M influences the maximum current amplitude I_{1_Max} because this resistors acts as additional damping resistor for L_RC_R circuit so that the ideal situation is when $R_M \rightarrow 0$ (see Fig. 6).

Fig. 6 Dependence of maximum current amplitude on the measuring resistor R_M and distance x



5 Mathematical Model Based on Continuous Excitation of Marker

This principle of marker localization is based on the continuous generating of the magnetic field into the space. In dependence on the distance x between coils the marker resonant circuit influences on the current $i_1(t)$ or voltage $u_{CR}(t)$ in locator antenna circuit in the case of serial or parallel antenna circuit, respectively. This models can be analysed by using of complex impedances of its components as it was performed in [5]. This approach cannot explain the transient phenomena in LC circuits because the results of calculations based on the complex impedances give only steady state solutions. Therefore new models based on differential equations were created.

5.1 Serial Arrangement of the Antenna

The schematic diagram of this model is similar as in previous model with the exception of switches and damping and sensing resistors (Fig. 7). The serial resonant circuits are placed in the localization device and in the transponder (marker), too.

The case of serial antenna circuit can be modelled by the next system of integrodifferential equations:

$$\begin{aligned} L_R\frac{di_1(t)}{dt}+R_Ri_1(t)+\frac{1}{C_R}\int_0^t i_1(\tau)d\tau-M\frac{di_2(t)}{dt}&=u_1(t),\\ L_T\frac{di_2(t)}{dt}+R_Ti_2(t)+\frac{1}{C_T}\int_0^t i_2(\tau)d\tau-M\frac{di_1(t)}{dt}&=0. \end{aligned} \tag{10}$$

The used signal source is given by Eq. (7). Similar as in previous chapter the system of integrodifferential equations (10) was transformed by substitution

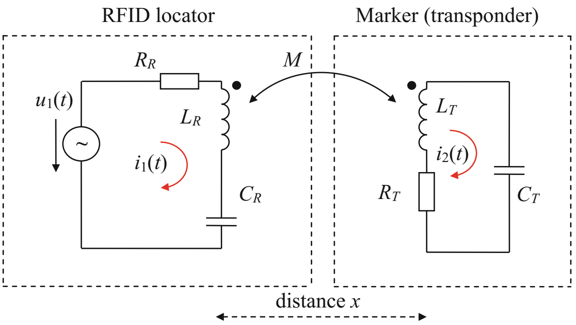


Fig. 7 Model of localization with continuous marker excitation and serial antenna circuit

$x_1(t) = i_1(t)$, $x_2(t) = di_1(t)/dt$, $x_3(t) = i_2(t)$, $x_4(t) = di_2(t)/\mathrm{d}t$, $\omega = 2\pi f$ into the 1st order system of differential equations (11) which can be easily solved by standard mathematic software. Individual coefficients are given by (12). Note that these coefficients are different from the coefficients given by (9).

$$\begin{aligned} \frac{dx_1(t)}{dt} &= x_2(t) \\ \frac{dx_2(t)}{dt} &= a_1 x_1(t) + a_2 x_2(t) + a_3 x_3(t) + a_4 x_4(t) - a_5 \omega U_1 \cos(\omega t) \\ \frac{dx_3(t)}{dt} &= x_4(t) \\ \frac{dx_4(t)}{dt} &= b_1 x_1(t) + b_2 x_2(t) + b_3 x_3(t) + b_4 x_4(t) - b_5 \omega U_1 \cos(\omega t) \end{aligned} \tag{11}$$

$$\begin{array}{ll} a_1 = \frac{L_T}{C_R(M^2 - L_R L_T)} & b_1 = \frac{M}{C_R(M^2 - L_R L_T)} \\ a_2 = \frac{L_T R_R}{M^2 - L_R L_T} & b_2 = \frac{R_R M}{M^2 - L_R L_T} \\ a_3 = \frac{M}{C_T(M^2 - L_R L_T)} & b_3 = \frac{L_R}{C_T(M^2 - L_R L_T)} \\ a_4 = \frac{M R_T}{M^2 - L_R L_T} & b_4 = \frac{L_R R_T}{M^2 - L_R L_T} \\ a_5 = \frac{L_T}{M^2 - L_R L_T} & b_5 = \frac{M}{M^2 - L_R L_T} \end{array} \tag{12}$$

In this case the detection of marker vicinity is more complicated as in the previous chapter because there is not the time stage in which only the response from excited marker can be received but the excitation signal in locator circuit is always present and combined with the signal from marker. The symptom of marker vicinity is decreasing of steady state current amplitude in locator circuit. When the distance between marker and locator is big ($x \rightarrow \infty$) then the mutual inductance calculated from (2) is very small ($M \rightarrow 0$). In this case the current $i_1(t)$ has maximum value I_{1max}, which is measured in steady state after the time $t = 1.9$ ms. The marker vicinity then causes current drop ΔI_1 (13) which can be calculated as difference between steady value of current $i_1(t)$ (see Fig. 8) and its maximum I_{1_max} measured in the same time point when no marker is nearby the locator.

$$\Delta I_1 = |I_{1\max}| - |I_1| \tag{13}$$

The time dependence of current $i_1(t)$ in the serial resonant circuit $L_R C_R$ (Fig. 8) was numerically calculated from the Eq. (11) for the same parameters as used in previous chapter. The dependence of relative current drop ΔI_{1r} (relative to the $I_{1\max}$) versus distance x is shown in Fig. 9 for various working frequencies of whole system.

The results calculated by analysis based on differential equations (11) were compared with the results obtained by analysis based on algebraic equations and complex impedances from [5] and both analyses give the same results.

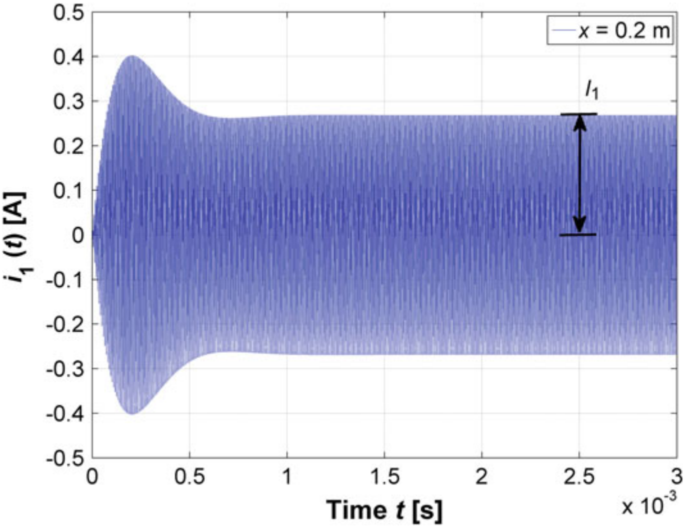


Fig. 8 The time dependence of current $i_1(t)$, distance $x = 0.2$ m

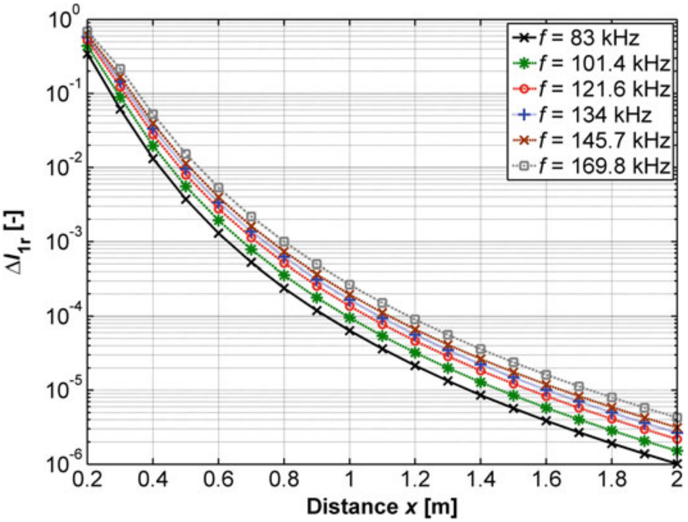


Fig. 9 Current drop ΔI_{1r} versus distance x for various working frequencies

5.2 Parallel Arrangement of the Antenna

The model is derived from the previous case replacing the antenna serial resonant circuit by parallel circuit as it is shown in Fig. 10. The principle of marker vicinity detection is similar as in the previous model—if the marker is nearby the locator

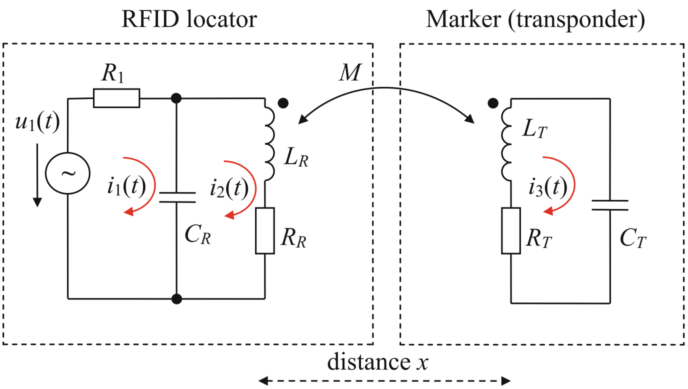


Fig. 10 Model of localization with continuous marker excitation and parallel antenna circuit

antenna then it influences the voltage in antenna resonant circuit. This model can be described by the following system of differential equations:

$$\begin{aligned}
&R_1 i_1(t) + \frac{1}{C_R}\int_0^t i_1(\tau)d\tau - \frac{1}{C_R}\int_0^t i_2(\tau)d\tau = u_1(t)\\
&L_R\frac{di_2(t)}{dt} + R_R i_2(t) + \frac{1}{C_R}\int_0^t i_2(\tau)d\tau - \frac{1}{C_R}\int_0^t i_1(\tau)d\tau - M\frac{di_3(t)}{dt} = 0\\
&L_T\frac{di_3(t)}{dt} + R_T i_3(t) + \frac{1}{C_T}\int_0^t i_3(\tau)d\tau - M\frac{di_2(t)}{dt} = 0
\end{aligned}\tag{14}$$

Assume that the signal source in the Fig. 10 is harmonic, corresponding to (7). The system of integrodifferential equations was transformed by substitution $x_1(t) = i_1(t)$, $x_2(t) = i_2(t)$, $x_3(t) = di_2(t)/\mathrm{d}t$, $x_4(t) = i_3(t)$, $x_5(t) = di_3(t)/\mathrm{d}t$. Then we get the 1st order system of differential equations (15) where individual coefficients a_1, a_2, a_3, a_4, a_5 and b_1, b_2, b_3, b_4, a_5 are given by (16).

$$\begin{aligned}
\frac{dx_1(t)}{dt} &= \frac{\omega U_1\cos(\omega t)}{R_1} - \frac{1}{C_R R_1}x_1(t) + \frac{1}{C_R R_1}x_2(t)\\
\frac{dx_2(t)}{dt} &= x_3(t)\\
\frac{dx_3(t)}{dt} &= -a_1x_1(t) + a_2x_2(t) + a_3x_3(t) + a_4x_4(t) + a_5x_5(t)\\
\frac{dx_4(t)}{dt} &= x_5(t)\\
\frac{dx_5(t)}{dt} &= -b_1x_1(t) + b_2x_2(t) + b_3x_3(t) + b_4x_4(t) + b_5x_5(t)
\end{aligned}\tag{15}$$

$$\begin{aligned} a_1 &= a_2 = \frac{L_T}{C_R(M^2 - L_R L_T)} & b_1 &= b_2 = \frac{M}{C_R(M^2 - L_R L_T)} \\ a_3 &= \frac{R_L L_T}{M^2 - L_R L_T} & b_3 &= \frac{R_L M}{M^2 - L_R L_T} \\ a_4 &= \frac{M}{C_T(M^2 - L_R L_T)} & b_4 &= \frac{L_R}{C_T(M^2 - L_R L_T)} \\ a_5 &= \frac{R_T M}{M^2 - L_R L_T} & b_5 &= \frac{R_T L_R}{M^2 - L_R L_T} \end{aligned} \tag{16}$$

Similar as in previous chapter if the marker is not in the locator reach i.e. $x \to \infty$ then from (2) it results that the mutual inductance $M \to 0$. Then the voltage $u_{CR}(t)$ has maximum value U_{CRmax}, which is measured in steady state after the time reaches $t = 1.9$ ms. The proximity of marker then can be detected when the steady value of voltage U_{CR} in time $t = 2.5$ ms is decreased by ΔU_{CR}:

$$\Delta U_{CR} = |U_{CR\max}| - |U_{CR}| \tag{17}$$

The time dependence of voltage $u_{\mathrm{CR}}(t)$ in the RFID locator circuit (Fig. 11) was numerically calculated from the system (15) for these parameters of the whole model, which are similar as in the previous chapter:

- Voltage of signal generator $u_1 = 10$ V, frequency $f = 134$ kHz, i.e. $\omega = 2\pi f = 841.9$ krad/s
- Distance between coils $x = 0.2$ m, angle $\theta = 0°$
- Radius of coils $r_R = 0.1$ m, $r_T = 0.1$ m
- Resistor $R_1 = 100$ kΩ

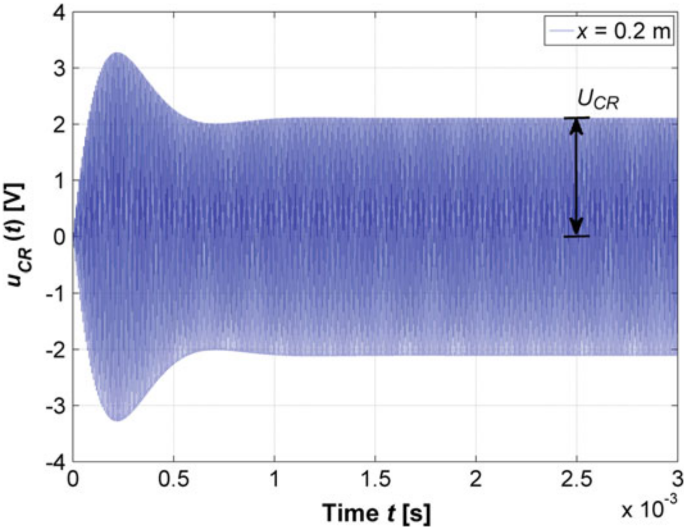


Fig. 11 The time dependence of voltage $u_{CR}(t)$, distance $x = 0.2$ m

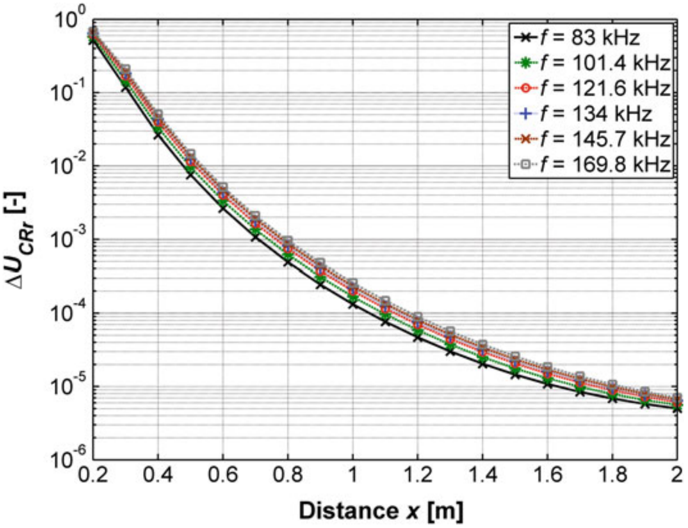


Fig. 12 Voltage drop Δu_{CRr} versus distance x for various working frequencies

- $C_R = 1.411$ nF, $L_R = 1$ mH, $R_R = 5\ \Omega$, i.e. antenna of locator has resonant frequency $f_R = 134$ kHz, quality factor of L_RC_R tuned circuit is $Q_R = 69$, calculated from (18)
- $C_T = 1.411$ nF, $L_T = 1$ mH, $R_T = 7.85\ \Omega$, i.e. resonant frequency of marker is $f_R = 134$ kHz, quality factor of L_TC_T tuned circuit is $Q_T = 107$.

Moreover, the relative voltage change ΔU_{CRr} (relative to the U_{CRmax}) was calculated for various working frequencies of locator—marker system to compare localization sensitivity in the case of various frequencies. This comparison is shown in Fig. 12.

$$Q_R = \frac{1}{R_1\sqrt{\frac{C_R}{L_R}} + \frac{1}{R_R}\sqrt{\frac{L_R}{C_R}}} \tag{18}$$

5.3 Comparison of Both Antenna Circuits

The direct comparison of serial and parallel antenna circuits is not possible due to different physical quantities signalling the presence of marker (current in the serial circuit and voltage in the parallel circuit). The comparison can be done if relative values related to maximum of corresponding quantity will be used. The relative values of current ΔI_{1r} and voltage ΔU_{CRr} drops can be simply calculated from Eqs. (19) and (20). Their dependences on distance x are shown in Fig. 13 for the frequency $f = f_R = f_T = 134$ kHz.

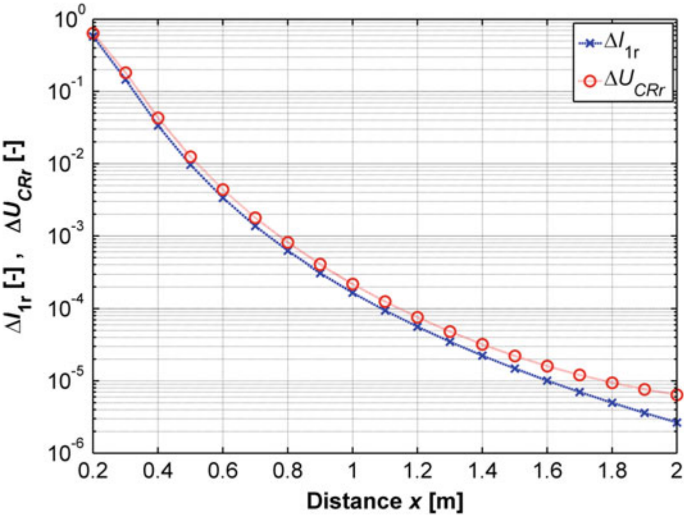


Fig. 13 Comparison of normalized changes (drops) for both antenna circuits

$$\Delta U_{CRr} = 1 - \left(\frac{U_{CR}}{U_{CR\max}}\right) \tag{19}$$

$$\Delta I_{1r} = 1 - \left(\frac{I_1}{I_{1\max}}\right) \tag{20}$$

6 Conclusion

Presented paper extends the mathematical analyses of marker localization principles. Two possible marker localization methods were analysed. For practical use the first method seems to be appropriate because the signal in receiving stage of localization can be directly processed.

The second method based on continuous marker excitation would require more complicated signal processing in locator. This complication is caused by “mixing” the excitation and response signals so that the marker vicinity cannot be simply detected by signal presence detection as in the first case but by detection of signal drop. Moreover the transient phenomena occur as documented in Figs. 8 and 11.

Another criterion for evaluation of presented localization methods is based on their sensitivity to the marker vicinity. From comparison in Fig. 14 the method based on the continuous marker excitation seems to be more sensitive to the marker vicinity. Because the markers are typical near field application, the current

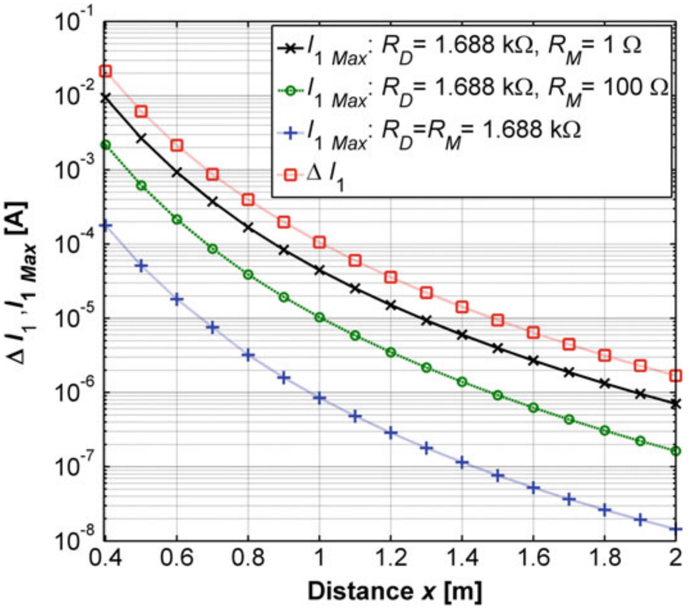


Fig. 14 Comparison of localization models for frequency $f = f_R = f_T = 134$ kHz

amplitude in both cases decreases very rapidly when the distance between marker and locator increases.

If we compare the same localization method but with various working frequencies as it is shown in Figs. 5, 9 and 12 the sensitivity to the marker vicinity increases with increasing working frequency. But the practical limitations exist—signals of higher frequencies are more attenuated by the influence of ground, especially if the ground contains water. Next practical limitation is caused by self-resonance effect of coils which limits the using of coils in serial resonant circuits.

The next research will be focused on the design of hardware needed to perform series of measurements so that the results of performed analyses will be compared with real measured data.

Acknowledgements This publication is the result of the project implementation: "Centre of excellence for systems and services of intelligent transport II", ITMS 26220120050 supported by the Research & Development Operational Programme funded by the ERDF. "Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ."

Európska únia
Európsky fond regionálneho rozvoja

Operačný program VÝSKUM a VÝVOJ

Agentúra
Ministerstva školstva, vedy, výskumu a športu SR
pre štrukturálne fondy EÚ

References

1. Finkenzeller, K.: RFID Handbook Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication, 3rd edn. John Wiley and Sons, Ltd. Chichester, UK (2010). ISBN 978-0-470-66512-1. http://dx.doi.org/10.1002/9780470665121
2. Vestenický, P., Mravec, T., Vestenický, M.: Analysis of inductively coupled RFID marker localization methods. In: FedCSIS 2015, Federated Conference on Computer Science and Information Systems, Lodz, pp. 1291–1295 (2015). ISBN 978-83-60810-66-8. http://dx.doi.org/10.15439/2015F298
3. Ahmad, M.Y., Mohan, A. S.: RFID reader localization using passive RFID tags. In: APMC 2009, Asia Pacific Microwave Conference, pp. 606–609, Singapore (2009). ISBN 978-1-4244-2802-1. http://dx.doi.org/10.1109/APMC.2009.5384152
4. Van Haute, T., Rossey, J., Becue, P., De Poorter, E., Moerman, I., Demeester, P.: A hybrid indoor localization solution using a generic architectural framework for sparse distributed wireless sensor networks. In: FedCSIS 2014, Federated Conference on Computer Science and Information Systems, Warsaw, pp. 1009–1015 (2014). ISBN 978-83-60810-58-3. http://dx.doi.org/10.15439/2014F20
5. Vestenický, P., Mravec, T., Vestenický, M.: Mathematical modelling of single-bit passive RFID marker localization methods. In: ELEKTRO 2014, 10th International Conference, Rajecké Teplice, pp. 504–507 (2014). ISBN 978-1-4799-3720-2. http://dx.doi.org/10.1109/ELEKTRO.2014.6848946
6. Markham, A., Trigoni, N., Macdonald, D. W., Ellwood, S. A.: Underground localization in 3-D using magneto-inductive tracking. IEEE Sens. J. **12**(6), 1809–1816 (2012). ISSN 1530-437X. http://dx.doi.org/10.1109/JSEN.2011.2178064
7. Abrudan, T. E., Markham, A., Trigoni, N.: Poster abstract: a case for magneto-inductive indoor localization. In: EWSN 2014, The 11th European Conference on Wireless Sensor Networks, University of Oxford, Oxford, pp. 18–19 (2014). ISBN 978-3-319-04651-8
8. Hautcoeur, J., Talbi, L., Nedil, M.: High gain RFID tag antenna for the underground localization applications at 915 MHz band. In: APSURSI 2013, IEEE Antennas and Propagation Society International Symposium, Orlando, pp. 1488–1489 (2013). ISBN 978-1-4673-5315-1. http://dx.doi.org/10.1109/APS.2013.6711403
9. Marin, S.A.: RFID made easy. Application note AN411, EM Microelectronic—Marin SA (2002). http://www.emmicroelectronic.com/webfiles/product/rfid/an/an411.pdf
10. Rosa, E. B., Grover, F. W.: Scientific Papers of the Bureau of Standards No. 169. In: Formulas and Tables for the Calculation of Mutual and Self-Inductance, vol. 8, no. 1 (1916)

Decomposition Scheme for Flow Design in Photonic Data Transport Networks

Andrzej Bąk and Mateusz Dzida

Abstract Development of sophisticated photonic transmission systems enabled evolution of photonic data transport networks towards cost-efficient and energy-efficient platforms capable to carry enormous traffic. Given access to technologically advanced equipment, network operator faces a series of decision problems related to how to efficiently use this technology. In this paper, we propose a mathematical model of network design problem applicable in the context of modern photonic network with wavelength division multiplexing (WDM). The mathematical models presented in the paper are given by means of Mixed-Integer-Programmes (MIPs). We first discuss a straight-forward formulation incorporating link-flow variables. The analysis of several real-size problem instances revealed that a need for a more advanced modeling approach, which will deliver problem formulations tractable by available hardware and solvers. Appropriate model was developed using Dantzing-Wolfe decomposition method. The resulting model uses link-flow based variables and incorporates significantly less constraints and variables if compared to the non-decomposed version.

Keywords WDM · Flow optimisation · Mixed-Integer-Programme

A. Bąk (✉) · M. Dzida
Optimax, ul. Wolbromska 19 m A, 03-680 Warsaw, Poland
e-mail: abak@poczta.pl; bak@tele.pw.edu.pl

M. Dzida
e-mail: dzida.mateuszmd@orange.com

A. Bąk
Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland

M. Dzida
Orange, ul. Sw. Barbary 2, 00-686 Warsaw, Poland

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent
Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_7

1 Introduction

This work is an extended version of [9], and presents an application of the Dantzig-Wolfe decomposition scheme to the photonic transport network flow design problem presented in the original paper.

Recent advances in the photonic networking enabled rapid growth of the transmission rates in the modern photonic data transport networks. Thus, photonic data transport networks became considerable alternative for traditional electric-based transmission systems, and are more and more widely deployed in the Autonomous Systems composing the Internet.

An important area of research in the domain of photonic data transport networks is associated with development of functional models of photonic networks and mathematical models of the decision problems associated with designing such networks.

Modeling telecommunication network is not a trivial task. On one hand, development of functional and mathematical models requires detailed knowledge of transmission technology and networking protocols. Thus, such models must be sufficiently detailed to accurately represent costs of equipment. On the other hand, it is necessary to consider a number of specific aspects that may further turn into design constraints. Most important of these include:

- architecture of switching equipment and transmission,
- mechanisms for traffic grooming and consolidation,
- network reconfiguration in case of failure,
- physical effects associated with signal propagation.

In the balance of this paper, being an extension of our work [9], we propose a mathematical (optimization) model of the photonic transport network flow design problem that takes into account all those mentioned aspects. Developed model is expressed in terms of integer programming. It refers to generic input data, including: network topology, infrastructure, and considered products. In particular, on one hand, input data must determine full cost characteristics of the considered equipment, usually in form of cost of particular expansion cards. On the other hand, input data are supposed to include locations of client devices and their demand for data transport services. It is therefore assumed that knowledge possessed by a network operator about demand structure is certain. In practice, knowing exact demand for transport services can be difficult, and sometimes even impossible. Still, we assume that through appropriate statistical methodology, it is possible to determine demand value with reasonable degree of confidence.

Paper is organized as follows. In Sect. 2 we define a mathematical model associated with designing flows in photonic data transport network. Assumptions and construction of network graph are discussed in Sects. 2.1 and 2.2, respectively. Considered flow design problem is formulated as mixed-integer programme in Sect. 2.3. Further, we investigate modeling specific aspects of the photonic data transport networks, related to: consistency of client flow at technology level Sect. 2.4, redundancy Sect. 2.5, impairments Sect. 2.6, and network cost Sect. 2.7. Then, in Sect. 2.8 we discuss application of the Dantzig-Wolfe decomposition scheme to the WDM flow design problem. Paper is summarized with estimation of formulations complexity in Sect. 3, and conclusions in Sect. 4.

2 Flow Design

In this section we consider flow design problem related to OTN/WDM photonic network. More information on architecture and interfaces of Optical Transport Networks (OTN) can be found in ITU-T recommendations [1, 2]. Considered flow design problem is formulated in terms of mathematical programming. Having given basic WDM network topology and set of traffic demands to be realized, OTN/WDM flow design problem is aimed at identifying a flow distribution and composition of expandable WDM components (e.g., muxponders, transponders, multiplexers, etc.) leading to optimized value of certain objective function. In particular, feasible solution of the considered problem identifies design of the Optical Network Element (ONE) nodes in terms of number, type, and configuration of expansion cards necessary to realize traffic demands, and associated cost.

Considered problem is described in the literature as Routing Wavelength Assignment (RWA) problem. It is commonly considered in combination with objective function maximizing the number of concurrent connections. Example integer linear programming formulation of this problem can be found in [22]. Independently in [6, 10] it was proved that RWA problem is $\mathcal{NP}$-complete. Formulations proposed in the literature [5, 7, 11, 14–18, 20, 21, 23, 24] differ from formulation proposed in the following in terms of graph construction. Namely, in this paper it is assumed that each λ channel constitutes separate edge in the network graph. Such assumption is not common in other works, but it allows to simplify formulation, and increase problem flexibility through graph construction. Moreover, classical RWA problem is concerned with routing and λ selection only. Here, problem is extended with consideration of the access side. This extension is motivated by usage of objective function related to cost of elastic expansion cards.

2.1 Assumptions

Traffic demands are assumed to be known in advance, e.g., they can be sourced from some external business forecast and measurement tool. As demand variation is out of the scope at considered network design problem, demand volumes may be additionally adjusted with some security margin. Each traffic demand is defined by triple: source, destination, and bandwidth volume. Demand source and destination are external clients connected to local ONE nodes through intra-office or short-haul black&white fibers.

Client devices and ONE devices are installed within Points of Presence (PoPs) of a network operator, and each client device is connected to uniquely defined ONE, usually in the same PoP. Even if in some PoP, ONE device is not installed, client localized in such PoP must be unambiguously assigned and connected to one ONE in one of the other PoPs.

After installing full suite of channel multiplexers, ONE device is capable to handle N channels, equal to its maximum capacity. Each channel has precisely defined central frequency and width. Central frequencies of consecutive channels are supposed to be compatible with one of optical grids defined by ITU-T.

2.2 Network Graph

Network topology at the simplest level defines locations, configuration, and type of network elements, and arrangement of long-haul fibers connecting network elements. Depending on required level of granularity, network topology can be more or less detailed. At level of details required by flow optimization, this simple topology needs to be extended with deeper insight into composition of network elements. For this purpose, we define a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ composed of set of nodes $\mathcal{V}$ and set of edges $\mathcal{E}$. Graph composition is used in the following as a basic modeling methodology. It allows to formulate considered flow design problem in terms of multi-commodity flow optimization.

2.2.1 Graph Nodes

In general, node set $\mathcal{V}$ can refer to four types of physical elements (cards or whole devices):

- optical network element—device responsible for multiplexing, switching, and (optionally) converting colorful λ signals ($\mathcal{O}$),
- transponder—expansion card responsible for adopting colorless tributary signals and modulating them as colorful λ signals ($\mathcal{T}$),
- muxponders—expansion card responsible for concatenating multiple colorless signals into higher-order colorful signals ($\mathcal{M}$),
- client—non-WDM device, consuming OTN services ($\mathcal{C}$).

In order to model switching and converting colorful λ signals, each ONE is represented in the network graph $\mathcal{G}$ by a set of graph nodes (referred to as *colorful nodes*), each associated with exactly one λ and one direction towards adjacent ONE. Accordingly, number of colorful graph nodes associated with single ONE is equal to $N \times D$, where D is the number of ONE neighbors. Similarly, basic graph of long-haul fiber connections is replicated, so there exists N (equal to number of λ's) parallel subgraphs, each topologically isomorphic with original network graph. If ONEs are capable to convert λ frequencies, all colorful nodes associated with single ONE needs to be interconnected. For example, if in the feasible solution, such artificial link is crossed on path between colorful nodes associated with λ_1 and λ_2, it means that signal incoming to the related ONE at λ_1 is transmitted out through λ_2. Example subgraph of colorful nodes associated with two neighboring 3-direction ONEs in reconfigurable optical add-drop multiplexer (ROADM) configuration is presented in Fig. 1. Tunable and reconfigurable optical add-drop multiplexer (T&ROADM) counterpart extends the ROADM subgraph with full mesh connections between colorful nodes inside ONE, as presented in Fig. 2.

Transponder and concentrator are sometimes combined as one expansion card – muxponder. If not combined, clients can be connected to transponders two-fold: through direct connections or indirectly through hierarchy of compatible concentrators. In the network graph, stand-alone transponders are associated with a subset of graph nodes $\mathcal{T}$, where each graph node is associated with one transponder type and one ONE location. Associated subgraph is presented in Fig. 3. In the figure, there are three transponder types (say 10, 40, and 100 Gbps) and two clients. Transponder graph nodes representing each transponder type in one location are connected to all colorful nodes.

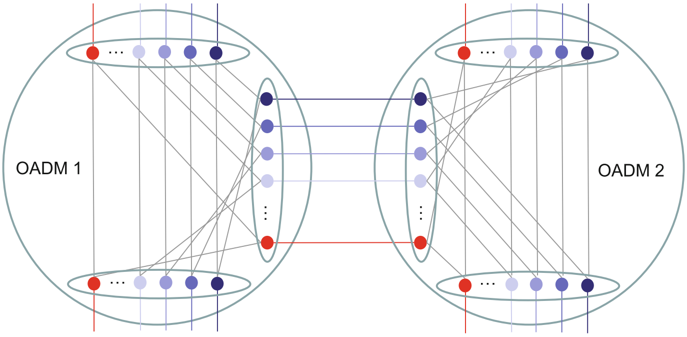


Fig. 1 ROADM subgraph

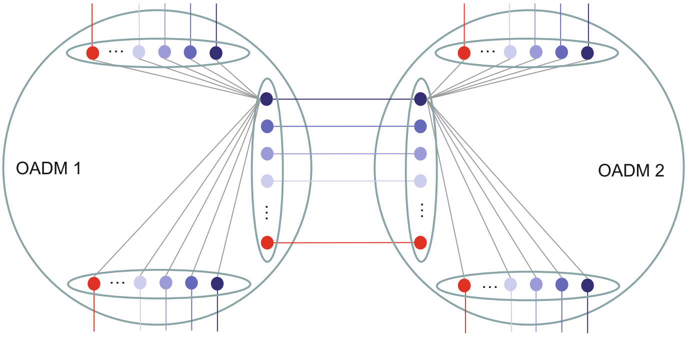


Fig. 2 T&ROADM subgraph

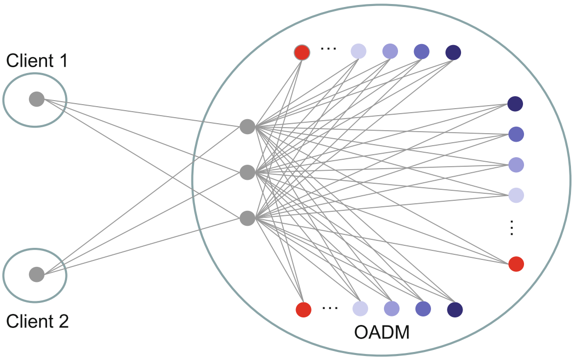


Fig. 3 Transponder subgraph

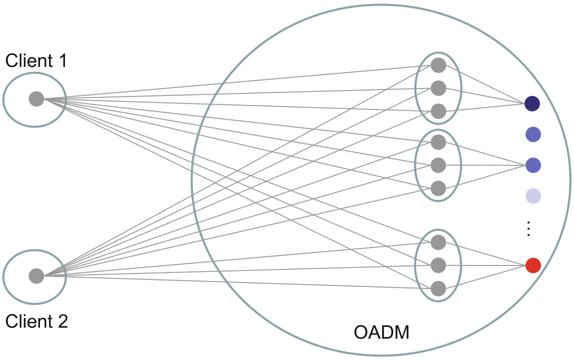


Fig. 4 Muxponder subgraph

Muxponders form subset of graph nodes $\mathcal{M}$, where each muxponder type is replicated N times, so each colorful node can be connected to its own unique suite of muxponders. Despite graph contains all potential muxponder cards, only some subset of cards may be required in the optimal solution. All muxponder graph nodes are connected with graph nodes representing compatible interfaces in the client devices, as it is presented in Fig. 4. Whole muxponder hierarchy is represented by single graph node, which means that multiple different connection types (different transport technology modules) are represented as parallel graph links, each associated with one transport technology type.

Client devices are not replicated in the network graph, and there exists exactly one graph node associated with each physical client device.

2.2.2 Graph Edges

Nodes $\mathcal{V}$ are connected by set of edges $\mathcal{F}$, referring to physical connections:

- long-haul fibers connecting ONEs ($\mathcal{H}$),
- intra-office fibers connecting line ports in client devices and tributary ports in muxponders/transponders,
- patch-cords and back-plane wiring connecting line ports in transponders and tributary ports in ONE multiplexers.

All enumerated types of connections: fibers, patch-cords, and wiring are further described by common term link. Set $\mathcal{F}$ is assumed to be a superset of the link set defined by the basic network topology. In particular, it contains replicated edges between adjacent colorful nodes. Finally, not all edges contained in $\mathcal{F}$ will be deployed, because some graph nodes represent non-existent components and link deployment will depend on card installation. A subset of these potential links will be selected for deployment or activation.

2.3 Mathematical Formulation

Based on introduced network graph definition, the WDM flow design problem can be formulated as below mixed integer programme.

Object Sets

$\mathcal{E}$	(directed) demands (e.g., IP links)
$\mathcal{V}$	nodes
$\mathcal{O} \subset \mathcal{V}$	colorful ONE nodes
$\mathcal{T} \subset \mathcal{V}$	transponders
$\mathcal{M} \subset \mathcal{V}$	muxponders
$\mathcal{C} \subset \mathcal{V}$	clients (e.g., IP routers)
$\mathcal{F} = \mathcal{H} \cup \mathcal{L}$	(directed) edges (WDM links)
$\mathcal{G} \subset \mathcal{F}$	edges associated with transponder links
$\mathcal{H} \subset \mathcal{F}$	edges associated with long-haul links
$\mathcal{L} \subset \mathcal{F}$	edges associated with intra-office links
$\mathcal{A}_v \subset \mathcal{F}$	edges outgoing from node $v \in \mathcal{V}$
$\mathcal{B}_v \subset \mathcal{F}$	edges incoming to node $v \in \mathcal{V}$
$\mathcal{P}_v \subset \mathcal{F}$	edges associated with add-drop (trib.) ports in colorful ONE nodes $v \in \mathcal{O}$
$\mathcal{Q}$	data transmission technologies

Predefined Objects

$a(e) \in \mathcal{C}$	originating client node (source) of demand $e \in \mathcal{E}$
$b(e) \in \mathcal{C}$	terminating client node (sink) of demand $e \in \mathcal{E}$
$a(f) \in \mathcal{V}$	originating client node (source) of edge $f \in \mathcal{F}$
$b(f) \in \mathcal{V}$	terminating client node (sink) of edge $f \in \mathcal{F}$
$\alpha(fe) \in \mathcal{L}$	terminating line related to originating link $f \in \mathcal{L}$ with regard to demand $e \in \mathcal{E}$

Constants

c_e volume of demand $e \in \mathcal{E}$

l_f capacity module of link $f \in \mathcal{F}$

t_v equal to the maximum number of active tributary links of muxponder, if $v \in \mathcal{M}$
equal to 1, if $v \in \mathcal{O}$

n_f equal to N, if $f \in \mathcal{G}$
equal to 1, if $f \in \mathcal{F} \backslash \mathcal{G}$

Variables

$s_{fe} \in \{0, 1\}$ variable equal to 1 if demand $e \in \mathcal{E}$ is realized on link $f \in \mathcal{F}$, and 0 otherwise

$z_f \in \mathbb{Z}$ variable equal to the number of transport modules on link $f \in \mathcal{F}$

Constraints

$$\sum\nolimits_{f \in \mathcal{A}_v} l_f s_{fe} = c_e \qquad e \in \mathcal{E}, v = a(e) \in \mathcal{C} \tag{1a}$$

$$\sum\nolimits_{f \in \mathcal{B}_v} l_f s_{fe} = c_e \qquad e \in \mathcal{E}, v = b(e) \in \mathcal{C} \tag{1b}$$

$$\sum\nolimits_{f \in \mathcal{A}_v} s_{fe} = \sum\nolimits_{f \in \mathcal{B}_v} s_{fe} \qquad e \in \mathcal{E}, v \in \mathcal{V} \backslash \{a(e), b(e)\} \tag{1c}$$

$$\sum\nolimits_{e \in \mathcal{E}} s_{fe} \le z_f \qquad f \in \mathcal{L} \tag{1d}$$

$$\sum\nolimits_{e \in \mathcal{E}} s_{fe} \le M z_f \qquad f \in \mathcal{H} \tag{1e}$$

$$\sum\nolimits_{f \in \mathcal{A}_v} z_f = \sum\nolimits_{f \in \mathcal{B}_v} z_f \qquad v \in \mathcal{O} \tag{1f}$$

$$\sum\nolimits_{f \in \mathcal{B}_v} z_f \le t_v \qquad v \in \mathcal{M} \cup \mathcal{O} \tag{1g}$$

$$\sum\nolimits_{f \in \mathcal{A}_v} z_f \le 1 \qquad v \in \mathcal{M} \cup \mathcal{O} \tag{1h}$$

$$z_f \le n_f \qquad f \in \mathcal{F}. \tag{1i}$$

Presented formulation is a modified form of classical formulation of multicommodity flow optimization problem (see [4, 12]). In this formulation, flow distribution is described by values of binary variables $\boldsymbol{s}$ representing flows on particular network links. Having given feasible values of $\boldsymbol{s}$ one can easily reconstruct particular paths selected to carry traffic.

Integer variables $\boldsymbol{z}$ determine in general number of transmission modules on particular links. However, in case of links associated with nodes $v \in \mathcal{M} \cup \mathcal{O}$ this number is strictly binary (due to constraints (1h) and (1i)). For the rest, variable $\boldsymbol{z}$ is integer (due to constraints (1g) and (1i)).

Due to classical flow conservation constraints (see [12]), in relation to specific demand, in all nodes, except end nodes of this demand, the total volume of incoming flows must be balanced by total volume of outgoing flows. Formulation (1) involves two groups of flow conservation constrains: constraints (1a)–(1c) related to variables $\boldsymbol{s}$ and constraints (1f) related to variables $\boldsymbol{z}$.

Usage of particular network links, including all types of inter-card patch-cords and back-plane wiring, by flows determines consumption of transport modules (their number is expressed by variables z), according to constraints (1d) and (1e).

Constraints (1g) assure that only one transponder or muxponder can be coupled with each channel tributary port in ONE multiplexer. Similarly, number of active tributary and line links connected to muxponder ports are limited by constraints (1g) and (1h), respectively.

Formulation (1) gathers constraints related to using WDM transport to carry client traffic. Based on this formulation, in the following we consider a number of its extensions and composition of objective function related to the overall cost associated with WDM transport.

2.4 L2 Technology

To express that each demand can be realized using homogenous L2 technology, like Gigabit Ethernet, FC800, STM64, binary variable vector $\boldsymbol{k}$ was introduced. Non-zero value of variable k_{ge} enforces through constraints (2a) that demand $e \in \mathcal{E}$ can be realized using only links compliant with technology $g \in \mathcal{Q}$. If one technology (say $g \in \mathcal{Q}$) is selected (value k_{ge} is 1), links associated with other technologies cannot be used, what is assured by constrains (2b).

$$\sum\nolimits_{f \in \mathcal{R}_g} s_{fe} \le |\mathcal{R}_g| k_{ge} \qquad g \in \mathcal{Q}, e \in \mathcal{E} \tag{2a}$$

$$\sum\nolimits_{g \in \mathcal{Q}} k_{ge} \le 1 \qquad e \in \mathcal{E}. \tag{2b}$$

Above, $\mathcal{R}_g \subset \mathcal{F}$ denotes set of edges associated with technology $g \in \mathcal{Q}$.

2.5 Redundancy

To provide uninterrupted services, able to survive failures of optical network elements and fiber connections, client devices need additional bandwidth, allocated along paths not affected by considered failures. Additional bandwidth, required by protection, is associated with certain level of resource redundancy. Redundant resources are either not used in the nominal network state or can be used for transmitting low priority traffic, preempted in case of failure occurrence.

In case of the WDM networks, redundant resources can be provided either at client digital signal level (called client protection) or photonic signal level (called photonic protection). In the former case, client device is responsible for activating redundant resources. Redundant resources cover optical channels allocated along protection path, and transponder/muxponder cards and ports. In the latter case, specialized protection cards are required. Such protection cards split power of the protected optical signal between multiple (usually two) ports connected to different add-drop ports within multiplexer subsystem. In both cases, the nominal and protection paths should be topologically disjoint with regard to failure occurrence, so under any failure at least one of the paths survives.

Let set $\mathcal{F}_i, i \in \mathcal{I}$ represents an arbitrary set of links that share risk of failure. Such group is described in the literature as Shared Risk Link Group (SRLG) or in general Shared Risk Resource Group (SRRG) [19]. Each SRLG associated with single link $f \in \mathcal{F}$ failure contains exactly one element. Each SRLG associated with node $v \in \mathcal{O}$ failure contains all adjacent links, i.e., $\mathcal{A}_v \cup \mathcal{B}_v$. SRLG should be constructed case by case in relation to specific needs and requirements of a network operator. Specific composition of SRLG thus remains out of scope of this paper.

In order to determine capacity c_{ei} allocated to demand e and available during failure state i, constraints (3a)–(3b) should be added to the problem formulation (1):

$$0 \le c_e - c_{ei} \le M r_{ei} \qquad e \in \mathcal{E}, i \in \mathcal{I} \tag{3a}$$

$$0 \le c_{ei} \le M(1 - r_{ei}) \qquad e \in \mathcal{E}, i \in \mathcal{I} \tag{3b}$$

$$0 \le r_{ei} \le \sum\nolimits_{f \in \mathcal{F}_i} s_{fe} \le M r_{ei} \qquad e \in \mathcal{E}, i \in \mathcal{I}. \tag{3c}$$

Above, each variable c_{ei} expresses volume of link flow associated with demand e in failure state i. Value of variable r_{ei} indicates if link e is available throughout failure state i. Consequently, if for some pair (e, i) $r_{ei} = 1$ then associated c_{ei} is equal to c_e, and c_{ei} is zero otherwise. In according to constraints (3c), value of r_{ei} is positive if and only if at least one link f realizing demand e is affected by failure i, i.e., when $\sum_{f \in \mathcal{F}_i} s_{fe} \ge 0$.

Redundancy required by protection mechanisms can be also modeled through multiplication of demand volume to be realized by the transport WDM/OTN network, and additional constraints assuring that only fraction of demand volume is transmitted through specific resources (network element or link). Protection method associated with described resource redundancy requirement is commonly described in the literature as path diversity [8]. To assure introduced requirement constraints (1a)–(1b) must be rewritten as:

$$\sum\nolimits_{f \in \mathcal{A}_v} l_f s_{fe} = 2c_e \qquad e \in \mathcal{E}, v = a(e) \tag{4a}$$

$$\sum\nolimits_{f \in \mathcal{B}_v} l_f s_{fe} = 2c_e \qquad e \in \mathcal{E}, v = b(e) \tag{4b}$$

$$\sum\nolimits_{f \in \mathcal{F}_i} l_f s_{fe} \le c_e \qquad e \in \mathcal{E}, i \in \mathcal{I} \tag{4c}$$

Constraints (4c) assure that any link flow do not exceed demand volume. In result, each demand must be realized on at least two disjoint paths.

2.6 Optical Impairments

Optical impairments (described in Sect. 2.6) affecting optical signals, transmitted through photonic network, can be modeled in the form of three limitations:

- fiber length limit,
- hop-count limit,
- noise accumulation limit.

Chromatic dispersion is responsible for spreading duration of optical signal peaks. As a linear effect, proportional to the total length of fiber, chromatic dispersion can be eliminated by using dispersion compensation modules (DCM, also called dispersion compensation units—DCU). DCM modules are supposed to introduced chromatic dispersion in reverse direction than dispersion introduced by regular fiber. DCM modules compensate thus chromatic dispersion related to central frequency of the optical signal. Consequently, dispersion affecting frequencies far from the central frequency are not compensated completely. Amount of uncompensated chromatic dispersion is called residual dispersion. Residual dispersion is an important factor that limits the total length of fiber traversed by optical signal. Maximum admissible fiber length depends on transponder type and characteristics.

Optical signal propagating through fiber medium is attenuated. Fiber attenuation is proportional to the total length of fiber spans crossed by the signal. To restore signal power to the level required by photo-detector, optical signal is amplified by so called Linearized Optical Fiber Amplifier (LOFA) cards localized in selected points along the path. However, LOFA cards, beside signal amplification, introduce some portion of noise. To keep signal quality at high level, network operator should control the total amount of introduced noise. On one hand, low noise level can be assured by hop-count constraints. On the other hand, noise characteristic of active optical elements, as mentioned LOFA cards, can be expressed in terms of Optical Signal to Noise Ratio (OSNR) value. Total value of OSNR is proportional to partial OSNR of particular elements in the path.

All introduced in this section limitations can be associated with additive metrics: length, hop-count, and inverse OSNR. Accordingly, all can be modeled similarly by set of so-called shortest path constraints. Formulation of shortest path constraints is based on path length variables $\boldsymbol{p} = (p_v \,:\, v \in \mathcal{O})$. Each p_v represents the length of the shortest path from v with respect to weight system $\boldsymbol{q}$. Then, for each link f outgoing from node v ($a(f) = v$) contained in the shortest path crossed by edge the following shortest path condition must hold.

$$p_{a(f)} + q_f = p_{b(f)} \tag{5}$$

Condition (5) is commonly used by the shortest path algorithms to validate if the path traversing edge f is shorter than the shortest path found so far. For our purposes we adopt condition (5) to formulate shortest path constraints:

$$p_{a(f)} + q_f - p_{b(f)} = 0 \text{ if value of } z_f \text{ is } 1 \quad f \in \mathcal{E} \tag{6a}$$

$$p_{a(f)} + q_f - p_{b(f)} \geq m \text{ if value of } z_f \text{ is } 0 \quad f \in \mathcal{E} \tag{6b}$$

Conditions (6a)–(6b) state that if and only if edge f is contained in the shortest path to $b(f)$, length of this path must be equal to sum of the length of a shortest path to $a(f)$ and weight q_f. Otherwise; the value of the expression $p_{a(f)} + q_f - p_{b(f)}$ must be greater or equal to m, which is the smallest difference between lengths of two paths. Accordingly, length limitation constraints can be formulated as follows:

$$m(1 - z_f) \leq p_{a(f)} + q_f - p_{b(f)} \leq Mz_f \qquad f \in \mathcal{E} \tag{7a}$$

$$p_v \leq p^* \qquad v \in \mathcal{V} \tag{7b}$$

Considered limitations require additional constraints (7b) to enforce that values of required parameters remain under maximum admissible level p^*. Weight system $\boldsymbol{q}$ is constant, and is supposed to express value of required parameter:

- link length,
- number of active elements associated with link (usually one),
- inverse OSNR associated with link.

2.7 Network Cost

Cost of WDM transport is mostly related to the number and type of used elastic expansion cards: channel multiplexers, transponders, and muxponders. Cost related to installation of transponder and muxponder cards can be expressed as follows:

$$\sum\nolimits_{v \in \mathcal{O}} \sum\nolimits_{f \in \mathcal{P}_v} \frac{1}{2} g_v z_f \tag{8}$$

Above, unitary cost related to card associated with node $v \in \mathcal{M} \cup \mathcal{T}$ is given by constant g_v.

To calculate cost related to installation of multi-stage multiplexer expansion cards we need to introduce additional variables and constrains. Binary variable m_j associated with multiplexer $j \in \mathcal{J}$ states if card is installed or not. Variable is positive if at least one channel associated with this particular multiplexer is used. This relation is expressed by constrains (9). Set of colorful channel links and cost associated with multiplexer $j \in \mathcal{J}$ are given by $\mathcal{S}_j$ and h_j, respectively.

$$\sum\nolimits_{f \in \mathcal{S}_j} z_f \leq |\mathcal{S}_j| m_j \qquad j \in \mathcal{J} \tag{9}$$

Finally, with respect to (9) and other constraints defined above, the objective function can be formulated as:

$$\boldsymbol{min} F(\boldsymbol{z}, \boldsymbol{m}) = \sum\nolimits_{v \in \mathcal{O}} \sum\nolimits_{f \in \mathcal{P}_v} \frac{1}{2} g_v z_f + \sum\nolimits_{j \in \mathcal{J}} h_j m_j \tag{10}$$

Objective function (10) is related to minimization of number of expansion cards.

2.8 Datzig-Wolfe Based Decomposition of WDM Flow Design Problem

In this section we discuss application of the Dantzig-Wolfe decomposition scheme to the WDM flow design problem given as problem (1). According to the principles of Dantzig-Wolfe decomposition, mathematical programme can be reformulated in terms of extreme points of polytope defined by subset of programme constraints. In such case, original programme variables are replaced by variables associated with particular extreme points. In result, programme needs reformulation (so some of original constrains can be omitted, some need to be rewritten using new variables, and some new must be introduced). Although, the number of extreme points can be very large in general, there exist techniques allowing to restrict number of variables introduced into reformulated programme.

Let convex polytope $\mathcal{S}$ by defined by flow conservation constraints (1a)–(1c). Flow conservation constraints are aimed at enforcing flows allocation along end-to-end paths between traffic sources and destinations. Extreme point (called also vertex) of a polytope is a point that by definition cannot be expressed as a linear combination of other points contained in the polytope. In case of polytope $\mathcal{S}$, extreme points represent flows along homogeneous end-to-end paths originated in $a(e)$ and terminated in $b(e)$ (for every IP link $e \in \mathcal{E}$). Thus, applying Dantzig-Wolfe principle, we can rewrite formulation (1) in terms of extreme points representing path flows. New formulation exploits flow variables associated with end-to-end paths $\boldsymbol{x} = \{x_p : p \in \mathcal{P}\}$, where $\mathcal{P}$ represents set of all paths. Path p realizing IP link e is represented by the set of its links, $\mathcal{F}_p \subseteq \mathcal{F}$, and joins the end nodes of link e. We define following path sets:

- $\mathcal{P}_e \subset \mathcal{P}$ representing subset of paths that can be used to realize IP link $e \in \mathcal{E}$,
- $\mathcal{P}_f \subset \mathcal{P}$ representing subset of paths that cross WDM link $f \in \mathcal{F}$,
- $\mathcal{P}_i \subset \mathcal{P}$ representing subset of paths that are available in SRLG $i \in \mathcal{I}$,
- $\mathcal{P}_g \subset \mathcal{P}$ representing subset of paths that use technology $g \in \mathcal{Q}$,
- $\mathcal{P}_{ef} = \mathcal{P}_e \cap \mathcal{P}_f$,
- $\mathcal{P}_{ei} = \mathcal{P}_e \cap \mathcal{P}_i$,
- $\mathcal{P}_{eg} = \mathcal{P}_e \cap \mathcal{P}_g$,

Each variable in the path flow formulation represents a vertex solution of the polyhedron defined by flow conservation constraints (1a)–(1c) for a specific IP link e. Thus, any admissible path satisfy the corresponding set of flow conservation constraints by definition. As a result, flow conservation constraints can be eliminated from path flow formulation. Capacity constraints and other constraints must be reformulated using the path flow variables.

According to the Dantzig-Wolfe principle, any flow pattern defined by set of flow conservation constraints can be expressed as a linear combination of end-to-end path flows associated with extreme points in polytope $\mathcal{S}$. In practice, linear combination of path variables associated with IP link e can be expressed as $\sum_{p\in\mathcal{P}_e} x_p$. Thus, path flow variables $\boldsymbol{x}$ are related to link flow variables $\boldsymbol{s}$ through relation:

$$s_{fe} = \sum_{p\in\mathcal{P}_{ef}} x_p \qquad f\in\mathcal{F}, e\in\mathcal{E} \qquad (11)$$

As the total flow realized by all WDM paths, associated with IP link e, must be equal (or greater) to required value c_e (according to flow conservation constraints), any point of $\mathcal{S}$ can be equivalently defined by the following convexity constraints.

$$\sum_{p\in\mathcal{P}_e} l_p x_p \geq c_e \qquad e\in\mathcal{E} \qquad (12)$$

Next, we reformulate WDM flow design formulation by eliminating link flow variables $\boldsymbol{s}$. For this purpose we substitute link flow variables by path flow variables, according to (11). Consequently, constraints (3c) can be transformed to:

$$0 \leq r_{ei} \leq \sum_{f\in\mathcal{F}_i} \sum_{p\in\mathcal{P}_{ef}} x_p \leq M r_{ei} \qquad e\in\mathcal{E}, i\in\mathcal{I} \qquad (13)$$

Observe, that $\sum_{f\in\mathcal{F}_i} \sum_{p\in\mathcal{P}_{ef}} x_p$ is equivalent to $\sum_{p\in\mathcal{P}_e\setminus\mathcal{P}_{ei}} x_p$. Accordingly, constraints (13) can be rewritten as:

$$0 \leq r_{ei} \leq \sum_{p\in\mathcal{P}_e\setminus\mathcal{P}_{ei}} x_p \leq M r_{ei} \qquad e\in\mathcal{E}, i\in\mathcal{I} \qquad (14)$$

Relation between link flow variables $\boldsymbol{s}$ and decision variables $\boldsymbol{z}$ is given by:

$$\sum_{e\in\mathcal{E}} s_{fe} \leq z_f \qquad f\in\mathcal{L} \qquad (15a)$$

$$\sum_{e\in\mathcal{E}} s_{fe} \leq M z_f \qquad f\in\mathcal{H} \qquad (15b)$$

Applying substitution (11) in (15), we arrive at:

$$\sum_{e\in\mathcal{E}} \sum_{p\in\mathcal{P}_{ef}} x_p \leq z_f \qquad f\in\mathcal{L} \qquad (16a)$$

$$\sum_{e\in\mathcal{E}} \sum_{p\in\mathcal{P}_{ef}} x_p \leq M z_f \qquad f\in\mathcal{H} \qquad (16b)$$

which in practice are equivalent to:

$$\sum\nolimits_{p\in\mathcal{P}_f} x_p \le z_f \qquad f\in\mathcal{L} \tag{17a}$$

$$\sum\nolimits_{p\in\mathcal{P}_f} x_p \le Mz_f \qquad f\in\mathcal{H} \tag{17b}$$

The only constraints left, that also has to be rewritten using path-flow notation is constraint (2a) which together with constraint (2b) assure that only one technology is used to establish all WDM paths supporting a specific IP link. Again, applying (11) to constraint (2a) we get:

$$\sum\nolimits_{f\in\mathcal{R}_g} \sum\nolimits_{p\in\mathcal{P}_{ef}} x_p \le Mk_{ge} \qquad g\in\mathcal{Q}, e\in\mathcal{E} \tag{18}$$

which is equivalent to (19).

$$\sum\nolimits_{p\in\mathcal{P}_{eg}} x_p \le Mk_{ge} \qquad g\in\mathcal{Q}, e\in\mathcal{E} \tag{19}$$

Formulation (1) of the WDM flow design problem with assigned objective function (10) constraints assuring homogenous usage of L2 layer technology (2), and redundancy constraints (3) can be finally rewritten using notation of path flows:

$$\boldsymbol{minimize} \sum\nolimits_{v\in\mathcal{O}} \sum\nolimits_{f\in\mathcal{P}_v} \frac{1}{2} g_v z_f + \sum\nolimits_{j\in\mathcal{J}} h_j m_j \tag{20a}$$

subject to

$$0 \le c_e - c_{ei} \le Mr_{ei} \qquad e\in\mathcal{E}, i\in\mathcal{I} \tag{20b}$$

$$0 \le c_{ei} \le M(1-r_{ei}) \qquad e\in\mathcal{E}, i\in\mathcal{I} \tag{20c}$$

$$0 \le r_{ei} \le \sum\nolimits_{p\in\mathcal{P}_e\setminus\mathcal{P}_{ei}} x_p \le Mr_{ei} \qquad e\in\mathcal{E}, i\in\mathcal{I} \tag{20d}$$

$$\sum\nolimits_{p\in\mathcal{P}_e} l_p x_p = c_e \qquad e\in\mathcal{E} \tag{20e}$$

$$\sum\nolimits_{p\in\mathcal{P}_f} x_p \le z_f \qquad f\in\mathcal{L} \tag{20f}$$

$$\sum\nolimits_{p\in\mathcal{P}_f} x_p \le Mz_f \qquad f\in\mathcal{H} \tag{20g}$$

$$\sum\nolimits_{f\in\mathcal{A}_v} z_f = \sum\nolimits_{f\in\mathcal{B}_v} z_f \qquad v\in\mathcal{O} \tag{20h}$$

$$\sum\nolimits_{f\in\mathcal{B}_v} z_f \le t_v \qquad v\in\mathcal{M}\cup\mathcal{O} \tag{20i}$$

$$\sum\nolimits_{f\in\mathcal{A}_v} z_f \le 1 \qquad v\in\mathcal{M}\cup\mathcal{O} \tag{20j}$$

$$z_f \le n_f \qquad f\in\mathcal{F} \tag{20k}$$

$$\sum\nolimits_{p\in\mathcal{P}_{eg}} x_p \le |\mathcal{R}_g| k_{ge} \qquad g\in\mathcal{Q}, e\in\mathcal{E} \tag{20l}$$

$$\sum\nolimits_{g\in\mathcal{Q}} k_{ge} \le 1 \qquad e\in\mathcal{E} \tag{20m}$$

Formulation (20) can be further simplified by combining constraints (20b)–(20e). The resulting formulation reads:

$$\boldsymbol{minimize} \sum\nolimits_{v\in\mathcal{O}} \sum\nolimits_{f\in\mathcal{P}_v} \frac{1}{2} g_v z_f + \sum\nolimits_{j\in\mathcal{J}} h_j m_j \tag{21a}$$

subject to

$$\sum\nolimits_{p\in\mathcal{P}_{ei}} l_p x_p = c_{ei} \qquad e\in\mathcal{E}, i\in\mathcal{I} \tag{21b}$$

$$\sum\nolimits_{p\in\mathcal{P}_f} x_p \le z_f \qquad f\in\mathcal{L} \tag{21c}$$

$$\sum\nolimits_{p\in\mathcal{P}_f} x_p \le M z_f \qquad f\in\mathcal{H} \tag{21d}$$

$$\sum\nolimits_{f\in\mathcal{A}_v} z_f = \sum\nolimits_{f\in\mathcal{B}_v} z_f \qquad v\in\mathcal{O} \tag{21e}$$

$$\sum\nolimits_{f\in\mathcal{B}_v} z_f \le t_v \qquad v\in\mathcal{M}\cup\mathcal{O} \tag{21f}$$

$$\sum\nolimits_{f\in\mathcal{A}_v} z_f \le 1 \qquad v\in\mathcal{M}\cup\mathcal{O} \tag{21g}$$

$$z_f \le n_f \qquad f\in\mathcal{F} \tag{21h}$$

$$\sum\nolimits_{p\in\mathcal{P}_{eg}} x_p \le |\mathcal{R}_g| k_{ge} \qquad e\in\mathcal{E}, g\in\mathcal{Q} \tag{21i}$$

$$\sum\nolimits_{g\in\mathcal{Q}} k_{ge} \le 1 \qquad e\in\mathcal{E} \tag{21j}$$

Certainly, the above path flow formulation works well only if the lists of paths ($\mathcal{P}_e, e\in\mathcal{E}$) contain all paths required for obtaining an optimal solution. This is, however, not easy to ensure, since we cannot put all the (simple) paths onto the lists, because the number of paths grows exponentially with the size of network graph. Thus, complexity of formulation (21) highly depends on the number of path flow variables contained in given formulation. However, techniques like column generation or its MIP counterpart branch-and-price allow to resolve this programme using only a subset of path flow variables, explicitly introduced into problem formulation. Column generation exploits observation that other path flow variables, not introduced into problem formulation, are equal to zero. In column generation approach, however, one needs to resolve a series of so called pricing problems which select new paths to be added to the set of candidate paths. In the case of our problem, the pricing problems are $\mathcal{NP}$-hard and applying this approach for large problem instances might not be an efficient approach. Moreover, even if a pricing problem is easy and can be solved in relatively short period of time, sometimes it appears that the number of generated paths is enormous, although not all generated paths are necessary. Thus, the alternative approaches based on pre-computing paths can be considered. These methods have been thoroughly discussed and presented in deliverable of the referenced ODIN project [13]. In the report, authors propose the algorithms for pre-computing candidate lists of primary paths most likely containing optimal solution paths. This should allow for an efficient resolving of path flow formulation (21). Two approaches for pre-computing paths are considered, generating the k-shortest paths, and generating paths using ε-approximation devised by Garg and Könemann. Still, whenever column generation technique is to be applied, precomputed candidate lists of primary paths allow to reduce the original hard pricing problems to the polynomial-time shortest path problem.

Despite the above discussed difficulties, the path flow formulation (21) has important advantages. First of all, the number of variables and constraints is of the order of N^2 (recall that N is the number of nodes), provided the lengths of the path lists ($\mathcal{P}_e, e \in \mathcal{E}$) are bounded (this is a reasonable assumption in many cases). In the link flow formulation (1) both the number of variables and the number of constraints are proportional to N^3. Second, the link flow formulation (1) does not provide any control over the type of paths that can be used (it assumes that all paths are used), unless special constraints are introduced. For example, the limit (say equal to k), imposed on path lengths in terms of some metric function, can be easily taken into account in the path flow formulation (21)—we just generate path lists containing all simple paths fulfilling the assumed length-limit in sense of particular metric function (e.g., fiber attenuation). On the contrary, using the link flow formulation (1) we have to use additional variables and constraints in the form of constraints (7a) and (7b). It turns out that in the case of length-limit MIP formulations, the linear relaxation of the adjusted formulation (i.e., the formulation in which $z_f \in \mathbb{R}^+$ instead of $z_f \in \mathbb{C}^+$) can give very bad lower bounds with respect to the linear relaxation of the corresponding mixed-integer programming formulation with appropriate path lists. The reason is that with relaxed variables z_f paths longer that k can be used in solutions of the linear relaxation of (1) with (7a) and (7b).

3 Complexity

Throughout this section we estimate complexity of the considered formulations of the WDM flow design problem. Complexity estimation is based on calculation of the numbers of variables and constrains necessary to formulate the considered WDM flow design problem in relation to the selected instances of network instances defined in the SNDLib library [3]. Referenced network instances are characterized in Table 1, where particular columns contain the numbers of network nodes, network links, and traffic demands, respectively.

Further, assuming 80-channel WDM technology and three types of transponders (10Gbps, 40Gbps, 100Gbps) we calculate the numbers of particular types of graph elements related to the considered network instances. Calculation results are presented in Table 2. When considering the path-flow based formulations, we assume that candidate paths lists will contain at most 10 paths per each demand. This assumption reflects our experience—in most of the practical size optimization problems considered so far, at most 10 paths were used in the final problem formulation (they were generated using column generation technique). For that problems, the globally optimal solution could be proved or the solution obtained was very close to optimum.

Table 1 Characteristics of the selected network instances

Network instance	Nodes	Links	Demands
Abilene	12	15	132
Atlanta	15	22	210
Brain	161	332	14311
Cost266	37	57	1332
Geant	22	36	462
Germany50	50	88	662
Giul39	39	172	1471
France	25	45	300
Janos-us	26	84	650
Janos-us-ca	39	122	1482

Table 2 Characteristics of the network graphs

Network instance	$\lvert\mathcal{O}\rvert$	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{L}\rvert$	$\lvert\mathcal{H}\rvert$	$\lvert\mathcal{F}\rvert$
Abilene	2400	12	1200	36	1236
Atlanta	3520	15	1760	45	1805
Brain	53120	161	26560	483	27043
Cost266	9120	37	4560	111	4671
Geant	5760	22	2880	66	2946
Germany50	14080	50	7040	150	7190
Giul39	27520	39	13760	117	13877
France	7200	25	3600	75	3675
Janos-us	13440	26	6720	78	6798
Janos-us-ca	19520	39	9760	117	9877

Finally, Table 3 contains the numbers of constrains and variables necessary to formulate the considered WDM flow design problems in relation to the selected SNDLib network instances. We present the numbers for both types of formulations—link-flow and path-flow based. Number of constraints is contained in a range from 19 thousands to 26 millions for the link-flow notation, and from 12 thousands to 5 millions for path-flow notation. Number of variables is larger and is contained in a range from 164 thousands to 387 millions for link-flow notation and 2 thousands to 170 thousands for path-flow notation. We can conclude the size of the link-flow based formulation is enormous with respect to numbers of constraints and variables and make in practice the considered formulations numerically intractable for resolving with exact optimization methods. More promising are path-based flow formulations, where the number of constraints required is approximately 2 to 4 times smaller, and what is more important, the number of variables is two-order of magnitude smaller.

Table 3 Characteristics of the formulations

Network instance	Link-flow formulation		Path-flow formulation	
	Constraints	Variables	Constraints	Variables
Abilene	19704	164388	12180	2556
Atlanta	38335	380855	19630	3905
Brain	26151647	387039416	5021942	170153
Cost266	484005	6226443	117954	17991
Geant	114852	1363998	41652	7566
Germany50	372608	4766970	117524	13810
Giul39	1263603	20426944	369210	28587
France	96825	1106175	43650	6675
Janos-us	307484	4425498	111116	13298
Janos-us-ca	956135	14647591	265046	24697

This allows to conclude, that the efficient method for resolving the types of problems considered in the paper, should be looked for in the area of path-based flow formulations, that should be resolved using column-generation technique.

4 Conclusion

Paper investigates mathematical modeling of photonic networks applying wavelength division multiplexing. It contains mathematical models of multicommodity flows in WDM network, first given as link-flow based formulation, and more tractable by currently available solvers and hardware, path-flow based formulation. The path-based flow formulation was developed as a consequence of hardness of link-flow based formulation containing, even for small network instances (composed of several devices) several thousands of constraints and variables. In order to reduce complexity of the proposed model to numerically tractable level, authors applied the Dantzig-Wolfe based decomposition which resulted in significant reduction in number of constraints and variables used. This result will allow for the development of efficient resolution methods applicable even for large network instances. In the future, it will be one of the main research topics carried out within the ODIN project.

Acknowledgements This work presents results of realization of first phase of ongoing project "Optymalizacja przepływów w wielowarstwowych sieciach dostepu do Internetu w technologii DWDM/IP/MPLS" (ODIN). Project is co-founded by The National Centre for Research and Development, Poland.

References

1. Architecture of optical transport networks: Recommendation ITU-T G.872
2. Interfaces for the optical transport network: Recommendation ITU-T G.709
3. SNDlib 1.0—Survivable network design data library (2005). http://sndlib.zib.de
4. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network flows: theory, algorithms, and applications. Prentice Hall (1993)
5. Aparicio-Pardo, R., Klinkowski, M., Garcia-Manrubia, B., Pavon-Marino, P., Careglio, D.: Offine impairment-aware rwa and regenerator placement in translucent optical networks. J. Lightwave Technol. **29**(3) (2011). http://dx.doi.org/10.1109/JLT.2010.2098393
6. Chlamtac, I., Ganz, A., Karmi, G.: Lightpath communications: an approach to high bandwidth optical WAN's. IEEE Trans. Commun. **40**(7), 1171–1182 (1992). http://dx.doi.org/10.1109/26.153361
7. Christodoulopoulos, K., Manousakis, K., Varvarigos, E.: Considering physical layer impairments in offine RWA. IEEE Netw. **23** (2009). http://dx.doi.org/10.1109/MNET.2009.4939260
8. Dzida, M., Sliwinski, T., Zagozdzon, M., Ogryczak, W., Pioro, M.: Path generation for a class survivable network design problems. In: NGI 2008 Conference on Next Generation Internet Networks, Cracow, Poland (2008). http://dx.doi.org/10.1109/NGI.2008.11
9. Dzida, M., Bak, A.: Flow design in photonic data transport network. In: Ganzha, M., Maciaszek, L. (ed.) Proceedings of the 2015 Federated Conference on Computer Science and Information Systems. Annals of Computer Science and Information Systems, vol. 5, pp. 471–482. IEEE (2015). http://dx.doi.org/10.15439/2015F148
10. Evan, S., Itai, A., Shamir, A.: On the complexity of timetable and multicommodity flow problems. SIAM J. Comput. **5**, 691–703 (1976). http://dx.doi.org/10.1137/0205048
11. Manousakis, K., Christodoulopoulos, K., Kamitsas, E., Tomkos, I., Varvarigos, E.: Offine impairment-aware routing and wavelength assignment algorithms in translucent WDM optical networks. J. Lightwave Technol. **27**(12) (2009). http://dx.doi.org/10.1109/JLT.2009.2021534
12. Minoux, M.: Mathematical Programming: Theory and Algorithms. John Wiley & Sons (1986)
13. Optimax: Optimization of multicommodity flows in optical internet access networks using WDM/IP/MPLS technology: Decomposition (2015). http://optimax-net.pl/raporty?download=2:decomposition
14. Pavon-Marino, P., Azodolmolky, S., Aparicio-Pardo, R., Garcia-Manrubia, B., Pointurier, Y., Angelou, M., Sole-Pareta, J., Garcia-Haro, J., Tomkos, I.: Offine impairment aware RWA algorithms for cross-layer planning of optical networks. J. Lightwave Technol. **27**(12) (2009). http://dx.doi.org/10.1109/JLT.2009.2018291
15. Saradhi, C., Subramaniam, S.: Physical layer impairment aware routing (PLIAR) in WDM optical networks: issues and challenges. IEEE Commun. Surv. Tutor. **11**(4) (2009). http://dx.doi.org/10.1109/SURV.2009.090407
16. Sengezer, N., Karasan, E.: Static lightpath establishment in multilayer traffc engineering under physical layer impairments. IEEE/OSA J. Opt. Commun. Netw. **2**(9) (2010). http://dx.doi.org/10.1364/JOCN.2.000662
17. Sengezer, N., Karasan, E.: Multi-layer virtual topology design in optical networks under physical layer impairments and multi-hour traffc demand. EEE/OSA J. Opt. Commun. Netw. **4**(2) (2012). http://dx.doi.org/10.1364/JOCN.4.000078
18. Sole, J., Subramaniam, S., Careglio, D., Spadaro, S.: Cross-layer approaches for planning and operating impairment-aware optical networks. In: Proceedings the IEEE 100 (2012). http://dx.doi.org/10.1109/JPROC.2012.2185669
19. Strand, J., Chiu, A., Tkach, R.: Issues for routing in the optical layer. IEEE Commun. Mag. (2001). http://dx.doi.org/10.1109/35.900635
20. Varvarigos, E., Manousakis, K., Christodoulopoulos, K.: Cross layer optimization of static lightpath demands in transparent WDM optical networks. In: IEEE Information Theory Workshop on Networking and Information Theory (2009). http://dx.doi.org/10.1109/ITWNIT.2009.5158553

21. Varvarigos, E., Manousakis, K., Christodoulopoulos, K.: Offlne routing and wavelength assignment in transparent WDM networks. IEEE/ACM Trans. Netw. **18**(5) (2010). http://dx.doi.org/10.1109/TNET.2010.2044585
22. Zang, H., Jue, J., Mukherjee, B.: A review of routing and wavelength assignment approaches for wavelength routed optical WDM networks. Opt. Netw. Mag. (2000)
23. Zhai, Y., Askarian, A., Subramaniam, S., Pointurier, Y., Brandt-pearce, M.: Cross-layer approach to survivable DWDM network design. IEEE/OSA J. Opt. Commun. Netw. **2**(6) (2010). http://dx.doi.org/10.1364/JOCN.2.000319
24. Zhang, W., Tang, J., Nygard, K., Wang, C.: REPARE: Regenerator placement and routing establishment in translucent networks. In: IEEE Global Telecommunications Conference GLOBECOM (2009). http://dx.doi.org/10.1109/GLOCOM.2009.5425649

Part II
Network Security

Secure Transmission in Wireless Sensors' Domain Supported by the TPM

Janusz Furtak and Jan Chudzikiewicz

Abstract Wireless sensor networks are an essential component of the fast-growing Internet of Things. The nodes of such network usually have small energy resources and do not have big computing power. The big challenge is to secure transmissions between nodes of the network and continuous authentication of nodes in data link layer of such network. This paper presents a proposal to solve this kind of problem using TPM in the domain of sensors. A model of wireless sensor network as well as operations associated with authentication in the sensors domain are presented. Additionally, an implementation of selected operations in the sensors domain is described. The test environment including the construction of nodes equipped with the TPM and obtained results related to the transmission delay time and power consumption are presented.

Keywords Security in WSN · Sensor authentication · Trusted platform module · Internet of things

1 Introduction

Nowadays, electronic communication is widespread. Through electronic links the data is exchanged between the people over long distances and in a short period of time. More and more often data is exchanged not only between the people, but also between devices. The Internet of Things becomes a common phenomenon. In all these applications the Wireless Sensor Networks (WSN) are widely used. This work is an extended and updated version of [1].

The nodes of WSN are devices called sensor nodes. The sensor node usually consists of one or more sensors, microcontroller, transceiver, and power source

J. Furtak (✉) · J. Chudzikiewicz
Military University of Technology, Warsaw, Poland
e-mail: janusz.furtak@wat.edu.pl; jfurtak@wat.edu.pl

J. Chudzikiewicz
e-mail: jan.chudzikiewicz@wat.edu.pl

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_8

[2, 3]. Most sensor nodes are capable of operating unattended for a long period of time reaching even a few years. In such applications, the biggest challenge is the efficient use of energy. This can be achieved by: minimizing energy consumption, proper power management and energy harvesting [4]. However, there are applications (e.g. military operations, actions of the police, rescue and others), in which the most important goal to achieve is safety and effective execution of the given task. In such cases maintenance-free operation time is not a priority task. In some cases, the life time of the sensor node may be restricted to few days or hours. In such applications the power consumption may not be a critical parameter. Applying a data encryption and an authentication of the sensor nodes, it is possible to achieve a suitable level of confidentiality and reliability of data as well as proper security level against attacks.

Security mechanisms implemented in WSN require cooperation between the sensor nodes due to the decentralized nature of the network and common absence of any infrastructure [5, 6]. The sensor nodes have to store a sensitive authentication data (e.g. cryptographic keys). Moreover, enemies/adversaries can introduce a fake sensor nodes aimed at impersonating the original ones. Given these safety requirements and limited node resources WSN, researchers have proposed various solutions, such as: secure and efficient routing protocols [7, 8], secure data aggregation protocols [9–13], and additional security mechanisms supported by the Trusted Platform Module (TPM) [1, 14–17].

Requirements for sensor nodes used in military applications are specific. Such sensor nodes are usually used to perform a single task and are expected to properly function in a relatively short time (e.g. only several hours or days). During this time they must ensure proper operation, secure data transmission, and be resistant to read or manipulate the data (especially cryptographic keys) stored in their resources in case of a takeover by an enemy. Transmitted data from the sensor nodes are usually directed to the commander. Due to a relatively short operating time of the sensor node, in practice there is no limitations on energy consumption for sensor nodes. Examples of such applications are shown in Fig. 1, but our attention will be focused on the application shown in the middle of Fig. 1.

Taking into account the specific requirements for sensor nodes in military applications the secure method of transmitting and storing data in WSNs is proposed in the paper. The presented method bases on the Trusted Platform Module (TPM). A TPM is used for secure storing the necessary data to authenticate the nodes, and generate symmetric keys, and asymmetric keys (private/public).

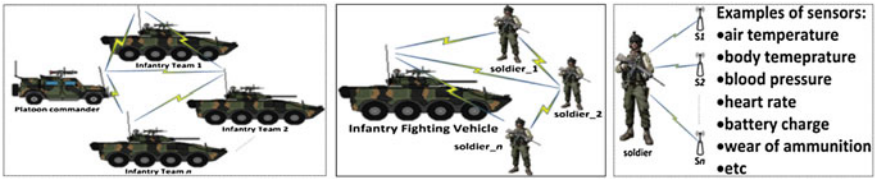


Fig. 1 WSN in military applications

Another, but also a very important problem is the safe transfer of data from WSN to customers connected to the Internet. The security mechanisms of transmission in the network layer (e.g. IP Sec) and higher layers (e.g. SSL, TLS) are well known. The biggest problem with the integration of security mechanisms on the Internet and WSN, in which XBee technology is normally used, is a small frame size, i.e. 127 bytes. The proposal to resolve this problem is presented in [13]. In this concept there are used: capabilities of IPv6 and 6LoWPAN, an interesting way of headers compression and encryption of data at the data link layer. However this concept does not provide the method of generating, distributing and storing cryptographic keys. Way of managing cryptographic keys described in this paper can be used in a method presented in [14].

In the second section there are presented basic definitions and a proposed model of WSN. In the third section the basic data structures used in the sensor nodes and data stored in the resource of sensor node are defined. The fourth section shortly describes procedures for ensuring proper authentication of sensor nodes in domain and correct data transfer between sensor nodes. Moreover it describes in detail a certain operations in sensors' domain. In the section a few experiments with selected operations in sensors' domain and obtained results are showed. Finally, a few concluding remarks are presented.

2 The Model of Wireless Sensor Network with Authentication[1]

A network of wireless sensors shown in the Fig. 2 creates the sensors' domain. In the domain there is exactly one Master node (M node), which acts as the security authority for all nodes in a domain. M node manages the credentials of all nodes of domain that participate in the exchange of data in the domain. M node is also the recipient of the data, which originate from Slave nodes ($\mathrm{S_j}$ nodes) or replica of Master nodes ($\mathrm{rM_k}$ nodes).

At first, each S node is registered in the domain of sensors and authenticated by M node. After that, the node can be a source of data. There can be several nodes in the domain plying the role replica of master (rM_k node), which will be able to take over the role of M node in case of its failure. The role of rM can be performed by properly equipped S node after establishing rM role for him.

During the registration process of S node, establishing rM role for S node and while sending data from S node and from rM node to M node the transmission is encrypted. S nodes store their encryption keys in their secure resources. M node and rM nodes store in their resources also the keys of all nodes in sensors' domain description. This description is encrypted using a key of M node or suitably by key

[1] The model of WSN with authentication and concept of authentication in WSN was presented on *2014 Federated Conference on Computer Science and Information Systems* [16].

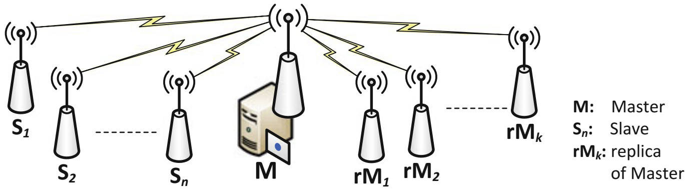


Fig. 2 The structure of sensors' domain

of rM nodes. Each domain node is equipped with a Trusted Platform Module (TPM),[2] which is used to support all the processes related to the generation of keys, the secure storage of node data (in particular encryption keys) and the process of encryption/decryption of data.

From the viewpoint of authentication procedures, nodes M and rM for nodes S are the same. Node rM can become a new node M of domain after changing its role, due to proven inactivity of the previous node M. In this case the node, which has so far acted as a node M, becomes a node rM, node S, or is removed.

When the sensor does not function, is turned off or damaged, it is assumed that this node is in a non-active state, and when the sensor is functioning, then the node is in the active state.

3 Resources of Sensors

Each sensor is equipped with a TPM. In non-volatile memory of TPM are stored the necessary data to authenticate the node in domain. Access to the memory is protected by Endorsment Key and Storage Root Key of the module. Additionally in resources of sensors playing the role of M or rM the description of the domain and descriptions of remaining domain nodes are stored. The data stored in resources of nodes are shown on Fig. 3.[3]

Content of credentials stored in TPM non-volatile memory of each node:

- EK (*Endorsment Key*)—key pair (private/public) generated in the development phase of the TPM—the private part of the key never leaves the module and is impossible to be read;
- SRK (*Storage Root Key*)—key pair (private/public) generated during the process of taking over ownership of the TPM in the procedure of initiating the

[2] TPM is an implementation of a standard developed by the Trusted Computing Group [18]. This module is designed to support the cryptographic procedures and protocols that can be used for securing data [19, 20].

[3] The data shown on Fig. 4, and further have been partially modified during the implementation of the method described in [1, 14].

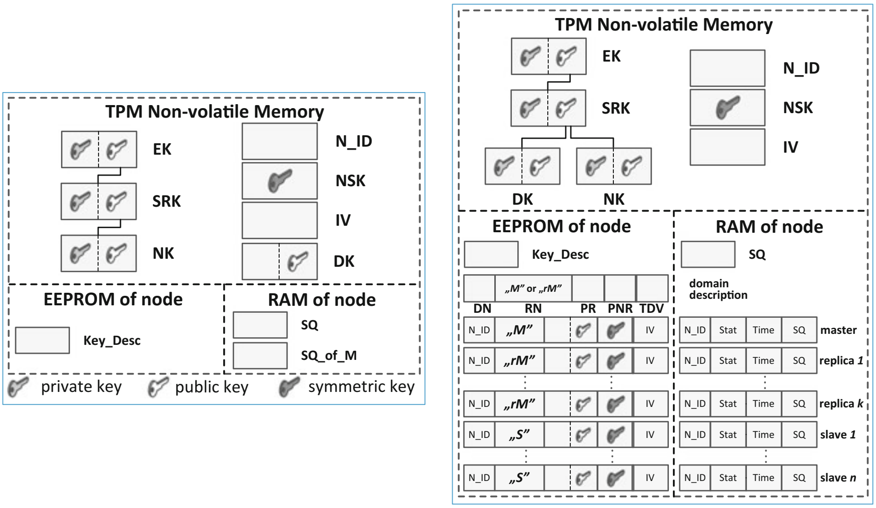


Fig. 3 The data stored on S node (*left*), M and rM node (*right*)

node; private part of the key is bound by public part of EK, and access to the key is protected by secret of module owner;

- NK (*Node Key*)—key pair (private/public) of node generated during the procedure for registering the node in the domain of sensors; private part of the key is bound by public part of SRK;
- DK (*Domain Key*)—key pair (private/public) of sensors' domain; generated in the process of creating the domain of sensors and establishing the role of M in the domain for the first node; each S node stores only public part of DK;
- N_ID (*Node ID*)—ID of the sensor;
- NSK (*Node Symmetric Key*)—symmetric key to the cryptographic protection of data transmission between S node and M node; generated during the procedure of registering the node in the domain and renovated in the regeneration procedure of the node credentials;
- IV—initiating vector for encryption using NSK key in Cipher Block Chaining mode;
- SQ—the sequence number of the last sent frame from S node (modified after each message);
- SQ_of_M—the sequence number of the last sent frame from M node (modified after each message).

Nodes, which are to play the role of M or rM, must be equipped with additional memory (e.g. EEPROM), which is intended to store the description of the domain and descriptions of remaining domain nodes as follows:

- **Domain description** (the structure of the data is showed on right side of Fig. 3):
  - DN (*Domain Name*)—the name of domain;

- RN (*Role of Node*)—determines whether the data are the resource of master node or the resource of replica of master; it is synonymous with the role it plays in the domain; may have values from the set {*M, rM*};
- PR (*Period of Replication*)—the time after which the rM node is required to establish communication with M node and refresh the domain data;
- PNR (*Period of Non-success Replication*)—the time after which rM node is obliged to repeat the attempt to establish communication with M node if the previous attempt to refresh the domain data was not successful;
- TDV (*Time of Data Validity*)—after this time, if the rM node failed to refresh the data, the domain data are invalid and node becomes an S node.

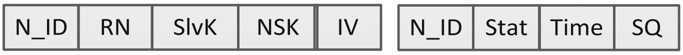
- **Description of domain nodes**. Description of each node consists of two parts. (the structure of the data is showed on Fig. 4). The first part includes the following data:

  - N_ID (*Node ID*)—ID of the sensor;
  - RN (*Role of Node*)—the role played by the node in the domain; it can take values from the set {*M, rM, S*};
  - SlvK—public part of an asymmetric key of N_ID node in sensors' domain;
  - NSK—symmetric key to encrypt the data sent from this node to M node; obtained during the procedure for registering the node in the domain and renovated in the procedure for the regeneration of S node credentials;
  - IV—initiating vector for encryption using NSK key in Cipher Block Chaining mode;

  The second part consists of the following status data:

  - N_ID (*Node ID*)—ID of the sensor;
  - Stat—status of the node; it can take one of the values: *non-active(-1)*, *active (0)*, *active non-confirmed (n)*, where *n* is the number of consecutive unsuccessful attempts to establish communication with the node
  - Time—moment of the last and the effective transmission;
  - SQ—the sequence number of the last sent frame (modified after each message).

The data: EK, SRK, NK, DK, N_ID, NSK and IV are stored in non-volatile memory of the TPM. Domain description and descriptions of nodes stored in EEPROM are secured using the NSK key and IV vector of M node. The status data of the nodes, except the N_ID field, are also encrypted using the NK key and IV vector of M node. All status data of domain nodes are organized as a list and are saved in RAM. The key description (Key_Desc) and sequential number of M node are stored as a plain text. After each power cycle the data stored in RAM must be refreshed.



N_ID	RN	SlvK	NSK	IV

N_ID	Stat	Time	SQ

Fig. 4 The data structure describing a sensor node

4 Operations in the Wireless Sensor Network with Authentication

The concept of authentication in WSNs using TPM was presented in [16]. The main assumptions of this concept were: to ensure proper authentication of sensors in the domain, the correct cryptographic protection of data transmission between the sensors, and increasing network resilience to DoS attacks (especially to a replay attack). The concept includes the following procedures:

1. Initiation of the sensors' domain covering the initiating of M node.
2. S node registration in the domain of sensors.
3. Transfer of the sensor data from S node to M node.
4. Removing rM or S node from the sensors' domain.
5. Authentication procedure of the node.
6. Integration test of nodes in sensors' domain.
7. Regeneration of S node credentials.
8. Giving the role rM in the domain for S node.
9. Resources update of rM node based on resources of M node.
10. Changing the node role from rM to M.
11. Election of the node to fulfill M node's functions after the failure of the previous M node.
12. Integration test of resources of M and rM nodes.

In this study in the following sections the procedures listed in first three paragraphs are comprehensively described. Implementation details of the procedures are described in the next section.

4.1 Initiation of the Sensors' Domain

This procedure is intended to create the domain of sensors and to initiate the node that will play the M role in the domain.

Input data: M node owner secret,[4] M node NK usage secret,[5] sensors' domain name (DN), M node identifier (N_ID) and time periods (i.e. PR, PNR and TDV) associated with the operation of nodes rM.

During the initiation of sensors' domain are performed the following steps:

1. Take over ownership of the TPM and generation of SRK key.

[4] Node owner secret is a mandatory parameter for procedure of taking over ownership of the TPM. The owner secret is stored within TPM non-volatile memory for future owner-authorized commands.

[5] Node NK usage secret is a mandatory parameter for procedure of generating NK key for node. The secret is stored within TPM non-volatile memory for future NK authorization use.

2. Generate asymmetric key NK (NK attributes: *Binding*, *Non-Migratable*, *Authority_always*, SRK is a parent of NK) and put it into the root of trust stored in the TPM of M node.
3. Generate the data for M node:
   - generate asymmetric key DK for sensors' domain and put it into the root of trust stored in the TPM of M node (DK attributes: *Storage*, *Migratable*, *Authority_always)*, SRK is a parent of DK; later public part of DK will be used by S node to bind the data which will be sent from S node to M node during registration procedure;
   - generate symmetric key (NSK—size 32 bytes) and initialization vector (IV —size 16 bytes) for AES cryptography;
   - generate sequential number SQ for M node;
   - put M node data into non-volatile memory of the TPM of M node.
4. Prepare the domain description, which includes the DN, RN, PR, PNR, TDV data and then encrypt this description using the NSK key and IV vector. The RN field should have a content of "M".
5. Prepare the M node description and then encrypt it using the NSK key, and IV. The fields of the description should have the following values:
   - N_ID = input data N_ID (the field is not encrypted);
   - RN = "*M*";
   - SlvK = public part of the node NK key;
   - NSK = the node NSK key;
   - IV = initiating vector for NSK key;
   - Stat = 0;
   - Time = current time;
   - SQ = random number from the range < 0; 65535 >.
6. Save the M node description in EEPROM and M node status data in RAM.

4.2 S Node Registration in the Domain of Sensors

In the procedure of S node registration in the domain is required that during this procedure the node is connected with the node M via the serial interface.[6]

Input data: S node owner secret, S node NK usage secret, S node identifier (N_ID).

During the S node registration in the domain of sensors are performed the following steps:

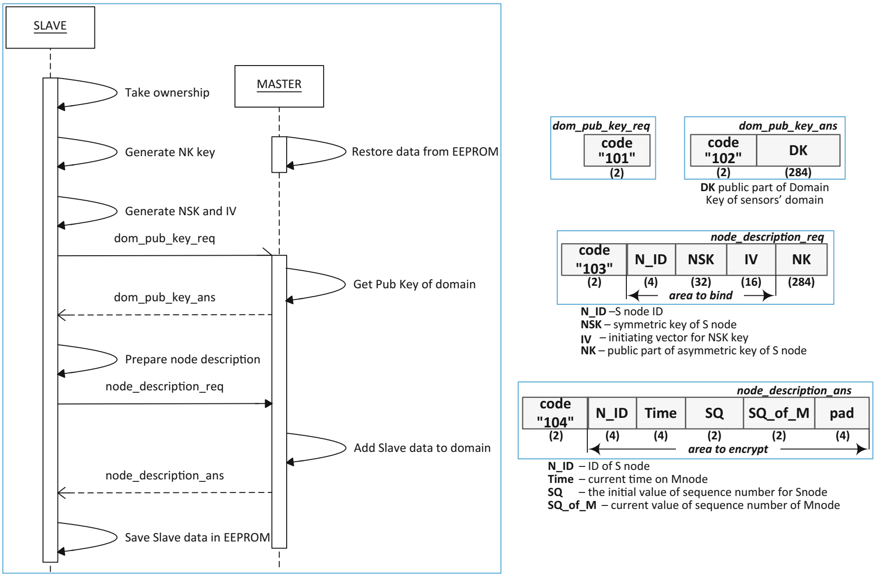
[6] If it was not possible to use the serial interface, in order to ensure the safety of the registration procedure, it is required to develop additional ways of mutual authentication of nodes involved in the registration process.

1. Connect S node to serial port of M node.
2. On S node: take over ownership of the TPM and generate SRK key.
3. Generate asymmetric key NK of S node (NK attributes: *Binding*, *Non-Migratable*, *Authority_always*; SRK is a parent of NK) and put it into the root of trust stored in the TPM of S node.
4. Generate the data for S node:
    - generate symmetric key (NSK—size 32 bytes), initialization vector (IV—size 16 bytes) for AES cryptography;
    - obtain the public part of the DK key from non-volatile memory of the TPM of M node—send a *dom_pub_key_req* packet from S node to M node through the serial line and receive from M node a *dom_pub_key_ans* packet;
    - put S node data into TPM non-volatile memory of S node.
5. Prepare S node description, bind it using public part of DK and send it to M node using *node_description_req* packet.
6. On M node: unbind the data from *node_description_req* packet using the private part of DK key, prepare the S node description, then encrypt this description using NSK key and IV vector of M node and save the S node description in M node resources. The fields of the S node description should have the following values:
    - N_ID = input data N_ID (the field is not encrypted);
    - RN = "S";
    - SlvK = public part of the S node NK key which is being registered;
    - NSK = the NSK key of node which is being registered;
    - IV = initiating vector for NSK key;
    - Stat = 0;
    - Time = current time;
    - SQ = random number from the range < 0; 65535 >.
7. Send a confirmation of registration to the node S (*node_description_ans* packet). The confirmation contains N_ID, Time and SQ and is encrypted using NSK key and IV vector of node S.
8. Disconnect the S node from serial port of M node.

The sequence diagram presenting cooperation between S node and M node during the registration process of S node in sensors' domain and the structure of used packets are showed on Fig. 5.

4.3 Transfer the Sensor Data from S Node to M Node

During normal operation, the data are received from the sensor of S node and are sent to M node. The sequence diagram presenting cooperation between M node and



DK public part of Domain Key of sensors' domain

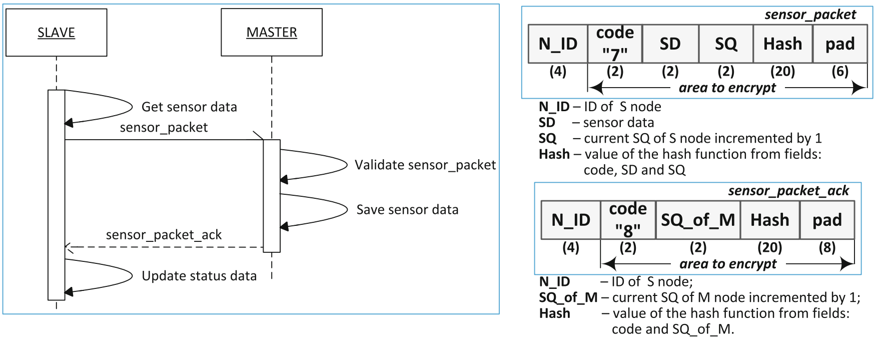
N_ID –S node ID
NSK – symmetric key of S node
IV – initiating vector for NSK key
NK – public part of asymmetric key of S node

N_ID – ID of S node
Time – current time on Mnode
SQ – the initial value of sequence number for Snode
SQ_of_M – current value of sequence number of Mnode

Fig. 5 The sequence diagram of S node registration process and structure of used packets

S node during the transfer of sensor data from S node to M node and the structure of the used packets are showed on Fig. 6.

To send the data *sensor_packet* frame is used. All fields of the frame except N_ID field are encrypted using NSK key of S node. After receiving the frame on M node, the fields SQ and hash are validated. If successful *sensor_packet_ack* frame is sent to S node as a confirmation of receipt of data. All fields of the frame except N_ID field are encrypted using NSK key of S node.



N_ID – ID of S node
SD – sensor data
SQ – current SQ of S node incremented by 1
Hash – value of the hash function from fields: code, SD and SQ

N_ID – ID of S node;
SQ_of_M – current SQ of M node incremented by 1;
Hash – value of the hash function from fields: code and SQ_of_M.

Fig. 6 The sequence diagram of the transfer of sensor data node from S node to M node and structure of the used packets

4.4 The Laboratory Stand to Examine Authentication Procedures in WSN

The laboratory stand to examine the authentication procedures in WSN utilizing TPM was developed. The laboratory stand (showed in Fig. 7) includes two sensors equipped with TPM—one M node and one S node.

Components of M node and S node are the same. Block diagram of these nodes and a view of exemplary sensor node is shown on Fig. 8. Sensor node used in the experiments was built with the following components (numbers in the list correspond to the numbers on the figure):

1. Arduino Mega2560R3—based on microcontroller ATmega2560 (clock 16 MHz, 256 KB Flash, 8 KB of SRAM, 4 KB of EEPROM, 16 analog inputs, 4 UARTs (hardware serial ports).
2. XBee 1mW Wire Antenna Series 1—wireless communication module.
3. Adapter XBee Shield (communicates XBee module with Arduino by Serial 0).
4. Power bank—9 V power supply.

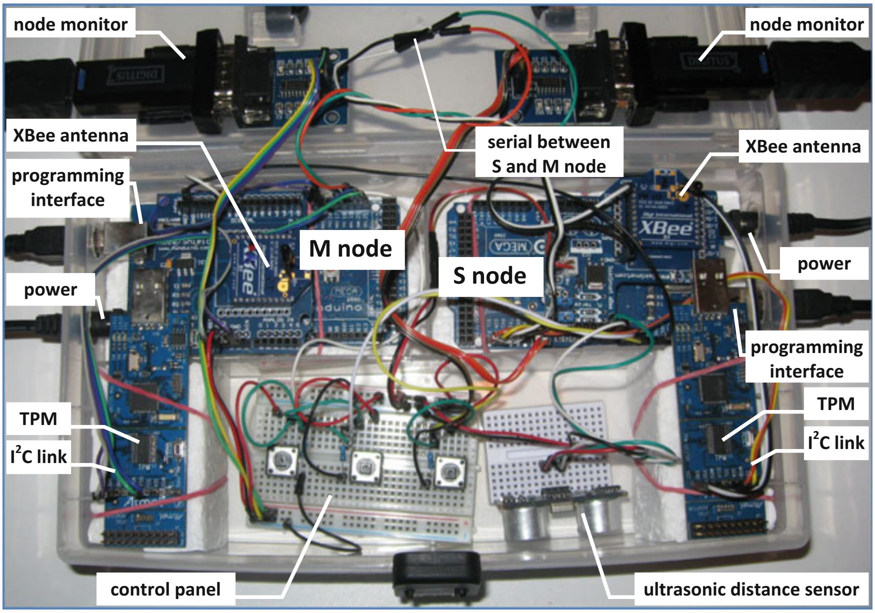


Fig. 7 View of laboratory stand

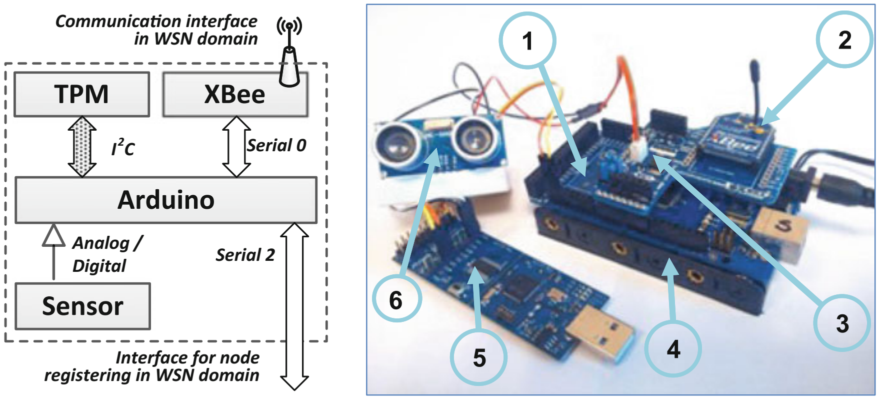


Fig. 8 Block diagram [16] and view of an exemplary sensor node [1]

5. TPM[7]—detachable part of hardware component of Atmel I^{2}C/SPI Demonstration Kit connected to Arduino through the I^{2}C Interface (in experiments was used part showed on Fig. 9).
6. Ultrasonic distance sensor—2–400 cm non-contact measurement.

The procedures of key generation are time-consuming and require a large amount of memory. For instance a description of one asymmetric key occupies 784 bytes, whereas M node has to store three such descriptions. The resources of Arduino microcontroller used in node, which plays the M role are insufficient to perform all functions by the M node. The most troublesome issue is lack of the memory, in particular the size of the SRAM, which is only 8kB. Therefore two software images for M node were prepared: “Master Init” and “Master Work”. Functions performed by “Master Init” software are:

- taking over ownership of the TPM;
- preparing M node data;
- preparing description of sensors’ domain;
- adding M node description to sensors’ domain;
- saving the generated authentication data in non-volatile memory (in EEPROM and NVRAM of TPM).

Functions performed by “Master Work” software are:

- restoring the data stored in non-volatile memory (from EEPROM and NVRAM of TPM);
- support of the registration process of S node in the domain of sensors—including adding a node description to the sensors’ domain description;

[7] Used module additionally meets the requirements described in Security Policy for Atmel TPM [21], which says that authentication mechanisms meet the strength requirements of FIPS 140-2, Level 2 [22].

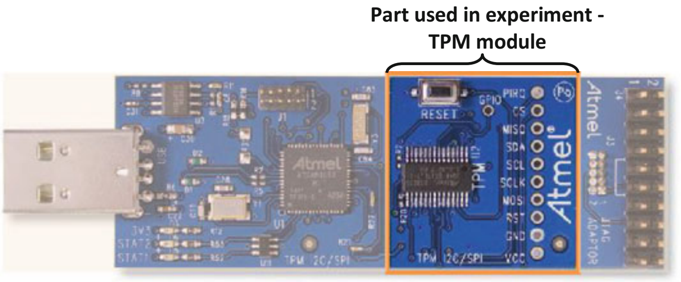

Part used in experiment - TPM module

Fig. 9 Atmel I2C/SPI Demonstration Kit

- receiving data from the sensors installed on the S node.

Functions performed by "Slave" software are:

- taking over the ownership of the TPM;
- preparing S node data—generating asymmetric key (NK) of S node, symmetric key (NSK) and initialization vector (IV) for S node;
- registration of S node in sensors' domain consisting of: acquisition of public key (DK) of sensors' domain from M node, preparation of S node description and transferring this description to M node;
- saving the generated authentication data in non-volatile memory (in EEPROM and NVRAM of TPM);
- sending to M node the data from the sensors installed on the S node.

The necessity of splitting the M node software into two parts forces the following behavior. Firstly, the software "Master Init" must be loaded into the controller of the M node, following the initialization procedure of the M node. In the next step, the "Master Work" software should be loaded into the controller of the M node. This method is inconvenient and time-consuming, but fortunately the change of the controller software has to be done only once. Further regular functioning of the M node does not require any software changes.

4.5 Experiment and Results

In laboratory stand was conducted an experiment consisting of the following stages:

1. Initiation of M node.
2. Registering of the S node in the domain of sensors.
3. Transferring sensor data from S node to M node:

   (a) sending the first frame from node S to M as a plain text i.e. without encryption.

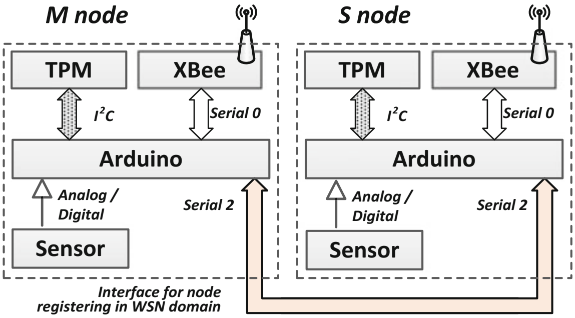


Fig. 10 Block diagram of M node and S node during the procedure of S node registering [1]

(b) sending the second frame (the sensor data is the same) from S to M node with AES encryption using NSK and IV of S node.

STAGE 1. Initiating of M node.

The entire first stage is initiated and implemented autonomously on the node that will act as the Master. On M node the "Master init" software is loaded. After this step the TPM is initiated and node ownership is acquired. The description of M node is written in non-volatile memory of TPM. Moreover, encrypted[8] description of sensors' domain, in which one node is registered (i.e. Master), is created. Examples of encrypted description of sensors domain for M node, which were created as a result of this step, you can find in [1].

STAGE 2. Registering of the S node in the domain of sensors.

Before the beginning of the second stage the software of M node should be replaced by a "Master work". Afterwards S node should be connected to the M node over a Serial link as shown in Fig. 10 [1].

In the first three steps of the stage TPM of S node is initiated, node ownership is acquired and the root of trust on the S node is created. Then direct connection to the M node by a serial link is needed to transfer the public part of the DK. Worth mentioning is that the DK is transferred as a plain text. In the next step NSK and IV is randomly generated and put into non-volatile memory of S node. Then the N_ID, NSK, IV and public part of the NK key are bound using public part of the DK key and transferred to M node through the serial link. On the basis of these data M node prepares a description of the node S and attach it to the sensors' domain description.

In the last step, confirmation of the S node registration (encrypted using NSK key of S node) is sent to S node. Due to the fact that the S node is registered, it should be disconnected from the serial link connecting it to the M node.

STAGE 3.

The S node is ready to transfer its sensor data by XBee interface—serial link used in stage 2 is disconnected. In experiment takes part, in addition to S and M

[8] In description of sensors' domain all fields (with the exception of node IDs) are encrypted using the NSK key and IV vector of M node.

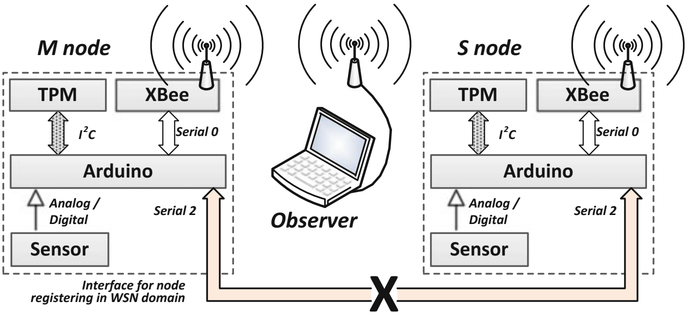
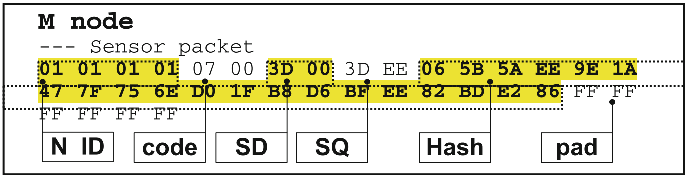


Fig. 11 Block diagram of M node, S node and observer during transferring data between S node and M node



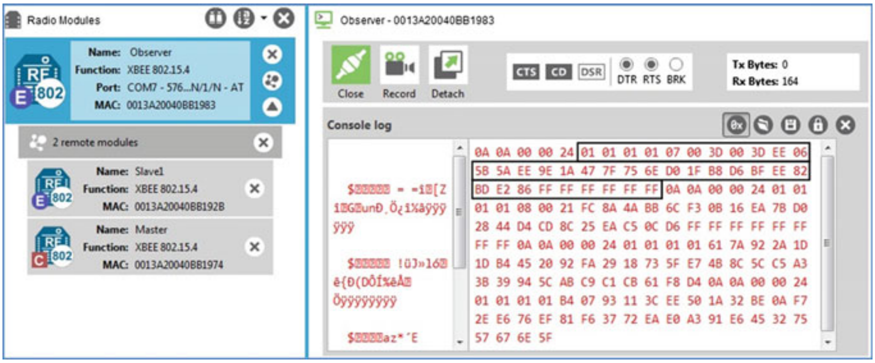
```
M node
--- Sensor packet
01 01 01 01 07 00 3D 00 3D EE 06 5B 5A EE 9E 1A
47 7F 75 6E D0 1F B8 D6 BF EE 82 BD E2 86 FF FF
FF FF FF FF
```

Fig. 12 Data received by M node with nonencrypted transmission

node, Observer station equipped with Xbee interface as shown on Fig. 11. This node is designed to intercept the data transmission between nodes S and M.

Data received by M node (Fig. 12) and Observer station (Fig. 13) during transmission without encryption are equivalent to each other.

Data received by the M node and the Observer station during encrypted transmission are equal as well. For the M node, the NSK key and the IV vector of the S



Name: Observer
Function: XBEE 802.15.4
Port: COM7 - 576...N/1/N - AT
MAC: 0013A20040BB1983

Name: Slave1
Function: XBEE 802.15.4
MAC: 0013A20040BB192B

Name: Master
Function: XBEE 802.15.4
MAC: 0013A20040BB1974

Tx Bytes: 0
Rx Bytes: 164

```
0A 0A 00 00 24 01 01 01 01 07 00 3D 00 3D EE 06
5B 5A EE 9E 1A 47 7F 75 6E D0 1F B8 D6 BF EE 82
BD E2 86 FF FF FF FF FF FF 0A 0A 00 00 24 01 01
01 01 08 00 21 FC 8A 4A BB 6C F3 0B 16 EA 7B D0
28 44 D4 CD 8C 25 EA C5 0C D6 FF FF FF FF FF FF
FF FF 0A 0A 00 00 24 01 01 01 01 61 7A 92 2A 1D
1D B4 45 20 92 FA 29 18 73 5F E7 4B 8C 5C C5 A3
3B 39 94 5C AB C9 C1 CB 61 F8 D4 0A 0A 00 00 24
01 01 01 01 B4 07 93 11 3C EE 50 1A 32 BE 0A F7
2E E6 76 EF 81 F6 37 72 EA E0 A3 91 E6 45 32 75
57 67 6E 5F
```

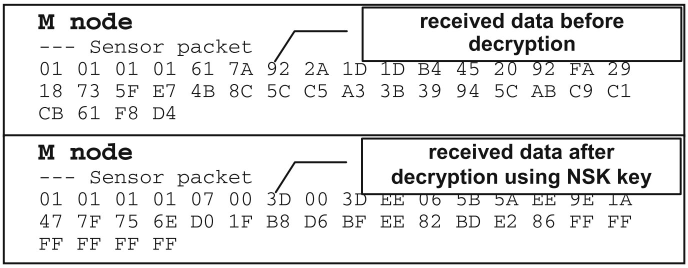
Fig. 13 Data received by Observer station with nonencrypted transmission

node are known and therefore, M node will be able to decrypt the fields from the received frame. The result is showed on Figs. 14 and 15.

The experiment shows that the data transferred between nodes S and M are secured cryptographically. Although the resistance to the attacks of the proposed solution requires confirmation through a relevant research, we can state without a doubt that even if unauthorized nodes, that are not registered in the domain of sensors, would intercept the data, they would not be able to use them directly and without delay.

During the experiment was verified the correctness of the following procedures: initiating the M node, registering the S node in the resource of the M node and the accuracy of data transfer (non-encrypted and encrypted) between S node and M node. In addition, the data transfer time between the M node and the S node and the level of resource utilization of these nodes had been measured.

Time of initialization of the M node, initialization and registration of the S node was relatively long and ranged from 20 to 65 s. The time depends on time-consuming process of generating the asymmetric keys by the TPM. It should be noted that these procedures for each node are performed only once. The data read from the S node sensor are transferred repeatedly. Transfer time of the data to



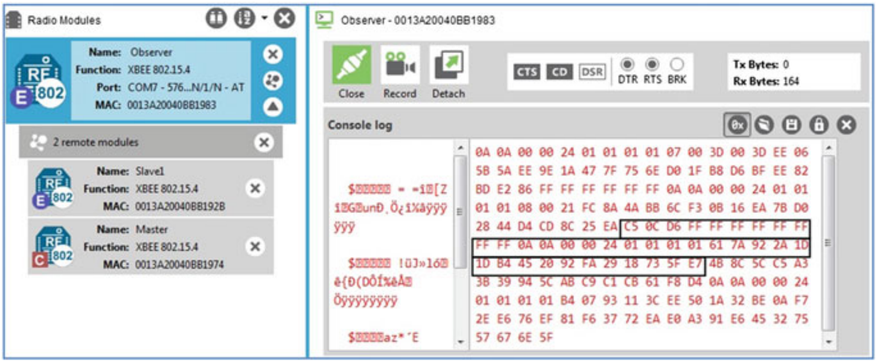
```
M node
--- Sensor packet
01 01 01 01 61 7A 92 2A 1D 1D B4 45 20 92 FA 29
18 73 5F E7 4B 8C 5C C5 A3 3B 39 94 5C AB C9 C1
CB 61 F8 D4
```

received data before decryption

```
M node
--- Sensor packet
01 01 01 01 07 00 3D 00 3D EE 06 5B 5A EE 9E 1A
47 7F 75 6E D0 1F B8 D6 BF EE 82 BD E2 86 FF FF
FF FF FF FF
```

received data after decryption using NSK key

Fig. 14 Obtained data from the encrypted transmission before and after decryption



Radio Modules

Name: Observer
Function: XBEE 802.15.4
Port: COM7 - 576...N/1/N - AT
MAC: 0013A20040BB1983

2 remote modules

Name: Slave1
Function: XBEE 802.15.4
MAC: 0013A20040BB192B

Name: Master
Function: XBEE 802.15.4
MAC: 0013A20040BB1974

Observer - 0013A20040BB1983

Tx Bytes: 0
Rx Bytes: 164

Console log

```
  $□□□□□ - -1□[Z      0A 0A 00 00 24 01 01 01 01 07 00 3D 00 3D EE 06
1□G□unÐ.Õ¿1%âÿÿÿ      5B 5A EE 9E 1A 47 7F 75 6E D0 1F B8 D6 BF EE 82
ÿÿÿ                   BD E2 86 FF FF FF FF FF FF 0A 0A 00 00 24 01 01
                      01 01 08 00 21 FC 8A 4A BB 6C F3 0B 16 EA 7B D0
  $□□□□□ !□J»16□      28 44 D4 CD 8C 25 EA C5 0C D6 FF FF FF FF FF FF
è{Ð(DÔÍ%êÅ□           FF FF 0A 0A 00 00 24 01 01 01 01 61 7A 92 2A 1D
Öÿÿÿÿÿÿÿ              1D B4 45 20 92 FA 29 18 73 5F E7 4B 8C 5C C5 A3
                      3B 39 94 5C AB C9 C1 CB 61 F8 D4 0A 0A 00 00 24
  $□□□□az*´E          01 01 01 01 B4 07 93 11 3C EE 50 1A 32 BE 0A F7
                      2E E6 76 EF 81 F6 37 72 EA E0 A3 91 E6 45 32 75
                      57 67 6E 5F
```

Fig. 15 Data received by Observer station with encrypted transmission

the M node and transfer time of the confirmation of the data are shown in Tables 1 and 2. The experiment was performed for two cases. In the first case the sensor data was transmitted without encryption. In the second case the same sensor data was encrypted using AES-256-CBC scheme.

The current consumption of the node components are shown in Fig. 16. The size of the current consumption almost does not depend on whether the sensor works, whether it is idle.

Resource utilization by the M node and the S node in the experiment are shown in Table 3. EEPROM memory size gives the opportunity to build a network of WSN consisting of no more than 11 nodes.

Table 1 Transfer time of data acquired from the sensor

SD [mm]	Sensor_packet		Slave		Transfer [ms]	Master		Total ovehead[a]
	Protect	Length [B]	Prepare [ms]	Difference [ms]		Support [ms]	Difference [ms]	
61	No	36	48,56	**95,30**	359,40	281,59	**116,68**	**58,98 %**
	AES-256-CBC	36	143,86		359,40	398,27		

[a] Total ovehead—the quotient of the sum delay times of encryption and decryption frame to the frame transfer time

Table 2 Acknowledgment receipt transfer time of data from the sensor

SD [mm]	Sensor_packet_ack		Master		Transfer [ms]	Slave		Total overhead
	Protect	Length [B]	Prepare [ms]	Difference [ms]		Support [ms]	Difference [ms]	
61	No	36	2,12	**116,25**	351,41	210,40	**95,82**	**60,35 %**
	AES-256-CBC	36	118,37		351,39	306,22		

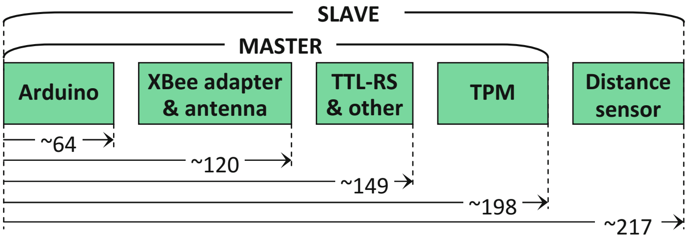


Fig. 16 Current consumption by node components [mA]

Table 3 Utilization of resources in different modes

Sensor mode	Flash memory		SRAM		EEPROM		
	Size [B]	Sketch [b] (ultilizing)	Size [B]	Min usage [b] (ultilizing)	Size [B]	Number of keys	Max size of WSN
Master init	258 048	37208 (**14** %)	8192	3725 (**45** %)	4096	3	11
Master work	258 048	36 568 (**14** %)	8192	4436 (**54** %)	4096	3	11
Slave	258 048	41 608 (**16** %)	8192	3972 (**48** %)	4096	2	–

4.6 Insights

Encryption/decryption of frames causes a delay of about 95 ms at the S node and 116 ms at the M node and is equal approximately 60 % of the frame transfer time (~350 ms). The difference of the delays appears due to the fact that the S node is to handle the frame reads of its AES key from NVRAM of TPM. In contrast, M node to handle the frame has to first read its AES key from NVRAM of TPM, and then using it, decrypts the AES key of the S node from the domain description.

The length of non encrypted "*sensor_packet*" and "*sensor_packet ack*" differ only by two bytes, but the time for preparing these frames is significantly different (e.g. in mode without encryption it is 48,56 – 2,12 = 46,44 ms). This difference stems from the fact that for the M node the identifier of the S node is available in the frame, whereas the S node has to read it from the NVRAM of TPM. It can be assumed that read time of one piece of data from the TPM NVRAM is around 50 ms.

The biggest challenge in the study, was the preparation of software for nodes so as the resources of nodes (especially RAM size) were sufficient for proper operation of this software. For this reason the dynamic memory allocation was used.

During the creation of this software in order to increase the robustness of the software on attack, we assumed that an encryption keys will stay resident in RAM only if the encryption/decryption procedures will be performed. Therefore, before any such procedure, the keys must be read from NVRAM of TPM and, in the case of the M node, additionally from the encrypted description of sensors' domain stored in EEPROM. After each such procedure appropriate area of RAM is always zeroed.

Consideration of both of these assumptions causes an increase of delay in the preparation and handling of data transfer. The values of the measured delays that are the result of the encryption of the data are relatively large and quite strongly differ, for example from the results presented in [23]. However, it is worth noting that the main purpose of the experiments presented in the paper was to verify the correct functioning of the proposed solutions, and the examination of time delays was only an additional element. The results of the delay measurement will be a reference point in next experiments with improved solutions.

5 Conclusion

This paper presents the model, concept of authentication in sensors' domain and implementation of securing transmissions between nodes of WSN. For this purpose, the mechanisms provided by the TPM are used. In this paper were presented only the most important operations in sensors domain: nodes initiation and transfer data between the nodes. Particular attention was paid to secure the transmission and to secure the nodes of network. In all procedures hardware support provided by the TPM was used. If you apply all the requirements specified in the security requirements for cryptographic modules (FIPS 140-2), the level of securing the data is high. The effect is, however, comes at a price relatively high power consumption and requires usage of modules, that have more computing power and more resources of RAM. The most substantial issue during the implementation was the shortage of sufficient RAM in used Arduino modules. For this reason, in further work we anticipate to use an additional EEPROM and/or a SDRAM memory.

In order to shorten the delay of preparing and handling of the data transfer, in the future work, we plan to focus on analyzing the risk of continuous storage of keys in the RAM. Another direction of work is to enable direct and secured communication between S nodes (not only from the S node to the M node) just like it is shown in [24] and to improve the integration of the developed solution with the capabilities of the XBee transmission system (just like it is described in [14]) aimed at reducing the time for the data transfer.

References

1. Furtak, J., Chudzikiewicz J.: Securing transmissions between nodes of WSN using TPM. In: 2015 Federated Conf. on Computer Science and Information Systems, pp. 1059–1068, PTI, Łódź (2015), IEEE Press, New York (2015), doi:10.15439/2015F144
2. Sohraby K., Minoli D., Znati T.: Wireless Sensor Networks Technology, Protocols, and Applications. Wiley, New Jersey (2007), doi:10.1002/047011276X
3. Faludi R.: Building Wireless Sensor Networks. O'Reilly Media (2010)
4. Stojcev, M.K., Kosanovic, M.R., Golubovic, L.R.: Power management and energy harvesting techniques for wireless sensor nodes. In: 9th Int. Conf. on Telecommunications in Modern Satellite, Cable and Broadcasting Services, pp. 65–72, IEEE Computer Society Press (2009), doi:10.1109/TELSKS.2009.5339410
5. Wang, Y., Attebury, G., Ramamurthy, B.: A survey of security issues in Wireless sensor networks. IEEE Commun. Surv. Tutor. **8**(2), 2–23 (2006). doi:10.1109/COMST.2006.315852
6. Sen, J.: A Survey on Wireless Sensor Network Security. Int. J. Commun. Netw. Inf. Secur. **1** (2), 59–82 (2009)
7. Boyle, D.: Securing wireless sensor networks: security architectures. J. Netw. **3**(1), 65–77 (2008)
8. Perrig, A., et al.: SPINS: security protocols for sensor networks. Wirel. Netw. **8**(5), 521–534 (2002). doi:10.1023/A:1016598314198
9. Al-Dhelaan, A.: Pairwise key establishment scheme for hypercube-based wireless sensor networks. Recent Researches in Computer Science

10. Mohd, Y., Hashim, H., Dani Baba, M.: Identity-based trusted authentication in wireless sensor network. Int. J. Comput. Sci. **9**(3), No 2, 230–239 (2012)
11. Hu L., Evans D.: Secure aggregation for wireless networks. In: Workshop on Security and Assurance in Ad Hoc Networks (2003)
12. Przydatek, B., Song, D., Perrig, A.: SIA: secure information aggregation in sensor networks, SenSys'03: In: 1st International Conference Embedded Networked Sensor Systems, pp. 255–65,. ACM Press, New York (2003). doi:10.1145/958491.958521
13. Hennebert, C., Dos Santos, J.: Security protocols and privacy issues into 6LoWPAN stack: a synthesis. IEEE Int. Things J. **1**(5), (2014). doi:10.1109/JIOT.2014.2359538
14. Furtak, J., Pałys, T., Chudzikiewicz, J.: How to use the TPM in the method of secure data exchange using Flash RAM media. In: 2013 Federated Conference on Computer Science and Information Systems, pp. 831–838, PTI, Warsaw (2013), IEEE Computer Society Press, Los Alamitos (2013)
15. Hu, W., Corke, P., Chan Shih, W., Overs, L.: SecFleck: a public key technology platform for wireless sensor networks. In: U. Roedig, J.S. Cormac (eds.) Wireless Sensor Networks, LNCS, vol. 5432, pp 296–311. Springer, Heidelberg (2009). doi:10.1007/978-3-642-00224-3_19
16. Furtak, J., Chudzikiewicz, J.: The concept of authentication in WSNs using TPM. In: M. Ganzha, L., Maciaszek, M. Paprzycki (eds.) Position Papers of the Federated Conference on Computer Science and Information Systems, pp. 183–190. PTI, Warszawa (2014). doi:10.15439/2014F176
17. Chudzikiewicz, J., Furtak, J., Zieliński, Z.: Secure protocol for wireless communication within Internet of Military Things. In: 2nd IEEE World Forum on Internet of Things (2015)
18. TPM Main Part 1 Design Principles. Specification Version 1.2. Revision 116. TCG Published (2011)
19. TCG Software Stack (TSS) Specification Version 1.2 Part1: Commands and Structures, http://www.trustedcomputinggroup.org/files/resource_files/6479CD77-1D09-3519-AD89EAD1BC8C97F0/TSS_1_2_Errata_A-final.pdf
20. Kinney, S.: Trusted platform module Basics: Using TPM in Embedded Systems. Elsevier Inc., Amsterdam (2006)
21. Module, Atmel Trusted Paltform: AT97SC3204/ AT97SC3205 Security Policy FIPS 140-2, Level 1. Atmel Corporation, Colorado, Springs (2014)
22. Security Requirements For Cryptographic Modules: Federal Information Processing Standard (FIPS 140-2), National Institute of Standard and Technology, Gaithersburg (2002). Retrieved 2013-05-18
23. Panait C., Dragomir D.: Measuring the performance and energy consumption of AES in wireless sensor networks. In: 2015 Federated Conf. on Computer Science and Information Systems, pp. 1261–1266, PTI, Łódź (2015), IEEE Press, New York (2015). doi:10.15439/2015F322
24. Elgenaidi, W., Newe, T.: Trust security mechanism for marine wireless sensor networks. In: 2015 Federated Conferences on Computer Science and Information Systems, pp. 1203–1208. PTI, Łódź (2015), IEEE Press, New York (2015). doi:10.15439/2015F169

Constraint Analysis for Security Policy Partitioning Over Tactical Service Oriented Architectures

Vasileios Gkioulos and Stephen D. Wolthusen

Abstract Tactical networks are typically of an ad-hoc nature operating in highly restricted environments and constrained resources. The frequent presence of communication disruptions and network partitioning must also be expected and managed, while core functionalities must be maintained, providing asynchronous invocation and access to services in a distributed manner. Supporting the required functionalities of the contemporary tactical environment, requires the dynamic evaluation of security policies, incorporating semantic knowledge from various network layers, together with facts and rules that are defined axiomatically a priori. However, the required basis for such policy decisions can be excessively extended and dynamic. Thus, it is desirable to locally minimize the scope of the policy maximizing efficiency. In this paper, we therefore analyze criteria and optimization goals for the a priori distribution and partitioning of security policies, ensuring the continuous support of the required capabilities, given the operational tasks of each deployed actor.

Keywords Ad Hoc network · Distribution · Security · Security policies · Tactical network · Partitioning

1 Introduction

Tactical networks refer to mobile networks, with characteristics similar to Ad-Hoc and mesh structures. They are typically adjusted and deployed to serve the specifics of a particular operation, with characteristics known partially in advance.

V. Gkioulos (✉) · S.D. Wolthusen
Norwegian Information Security Laboratory, Norwegian University of Science and Technology, Trondheim, Norway
e-mail: vasileios.gkioulos@ntnu.no

S.D. Wolthusen
e-mail: stephen.wolthusen@ntnu.no

S.D. Wolthusen
School of Mathematics and Information Security, Royal Holloway, University of London, London, UK

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_9

Consequently, the study, evaluation and realization of globally suitable security mechanisms, must be able to dynamically adapt to the versatile and diverse nature of tactical operations. The tactical environment is continuously studied, both in terms of operational analysis and technical evaluation [1–5], allowing the extraction of valuable information regarding their nature, characteristics and requirements.

The deployed assets for a specific operation should be expected to operate over distinct platforms, with diverse capabilities and requirements, including the ability to operate in coalition environments. Additionally, due to resource limitations and the dynamically evolving topologies, no safe assumptions can be made regarding continuous connectivity, since a tactical network may degrade to the point of partitioning. For the same reasons, communication failures, uncertain service delivery and extensive delays must be expected and properly addressed. Within this environment tactical networks must be able to provide reliable and secure service delivery and communication. Hence, the realized security mechanisms have to be distributed across the deployed assets, since no centralized security dedicated entity can be assumed, due to inability of reassuring a continuously available link towards it.

In addition to the aforementioned constraints, the introduction and increasing requirement of supporting Network Enabled Operations (NEO) and Network Centric Warfare (NCW), formulated a new set of requisite features regarding the functionalities of contemporary tactical networks [6–8]. Thus, mechanisms based on the Service Oriented Architecture (SOA) paradigm emerged as the most suitable mediators for the realization of these requirements, within the deployed C4I (Command, Control, Communication, Computers and Intelligence) and C4ISR (Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance) systems [9–15].

Securing tactical SOA requires not only the accomplishment of general information protection goals (such as confidentiality, availability, authenticity and control) but also the dynamic protection of communication, data at rest and processing, within the aforementioned restrictions imposed by their nature. The realization of suitable security mechanisms requires the conceptualization of the multitudinous semantic attributes available across the network. Such elements rise among others from services, terminals, information, communication links and subjects, alongside their relations and interactions.

Well known mechanisms (Such as WS-Security, Ponder [16], SAML [17], XACML [18], RT [19], cassandra [20], Peer-Trust [21], Tulip [22], ROWLBAC [23], REI [24], KAOS [25], Kolter et al. [26]) have been extensively studied and found to be unsuitable for the contemporary tactical environment for a variety of reasons. Some face limitations in capturing and expressing the required semantics, others are relatively heavyweight regarding their computational and communication requirements, or lack the ability of decentralized operation. Furthermore, some are not rigorous and flexible enough in expressing and reasoning over security policies, face scalability limitations or a combination of these reasons. These studies (Including but not limited to [23, 27–34]) promoted the use of ontologies for the definition of general purpose security policies, due to their expressive power and ability to overcome the aforementioned constraints.

For the same reasons in our previous study [35] we proposed a framework for the realization of an ontologically defined security infrastructure, with the use of Web Ontology Language (OWL), suitably adjusted to the constraints and high level functional requirements of tactical SOA. Yet, although ontologies can provide the required extended scope over the existing semantic attributes, the aforementioned inability to rely on a centralized security dedicated entity requires the distribution of the defined mechanisms across the deployed tactical nodes. However, due the functional limitations of tactical nodes (e.g. computational capacity, storage capacity, bandwidth availability), mere replication of those mechanisms across the network is inefficient and commonly infeasible.

In this paper we present our findings regarding the partitioning and distribution of ontologically defined security policies, suitably adjusted to the specifics of tactical SOA, aiming to maximize efficiency by minimizing the local scope of the policy. We approach this topic by identifying the criteria rising from the nature of tactical SOA, seeking a reliable limitation to a problem similar in nature to a 0-1 multiple knapsack problem, therefore subject to existing mechanisms of discrete optimization. Furthermore, we identify suitable elements in order to minimize the complexity by reducing the number of instances, maintaining the complete set of functionalities supported by the defined security policies.

2 Ontologically Defined Security Policies for Tactical SOA

An ontologically defined security policy dedicated to the specifics of tactical SOA must be able to provide the dynamic protection of communication, data at rest and processing, alongside the general information protection goals. Such a mechanism requires the conceptualization of the assorted semantic attributes, within a robust yet flexible mapping between the involved elements. These elements comprise of the defined *Domains* (Including but not limited to planning, protection, diligence, detection and response), the required *Capabilities* (Similar to NATO Architecture Framework/NATO Capability View (NAF/NCV) [36], including but not limited to core, application, communication and inter-domain), the available *Actions* and a set of governing *Rules* for each action, each of which incorporates a varying set of the involved *Conditions* (Which correspond to the aforementioned dynamic and static semantics). An outline of the security policy structure, including the overlaying relations, is presented at Fig. 1.

These elements are defined as OWL classes, which are populated according to the requirements of each tactical operation. The Security_Core is the anchor of the policy structure similar to owl: Thing of ontologies, incorporating all the other elements as subclasses. Furthermore, the Security_Core is the gateway towards the TSI_common (Tactical Service Infrastructure common core ontologies) and additional ontologies that are required to be linked with the security infrastructure. Thus, through the Security_Core the security policy can monitor the functionality of the enabled capabilities, within each tactical domain. This is achieved by the on-line

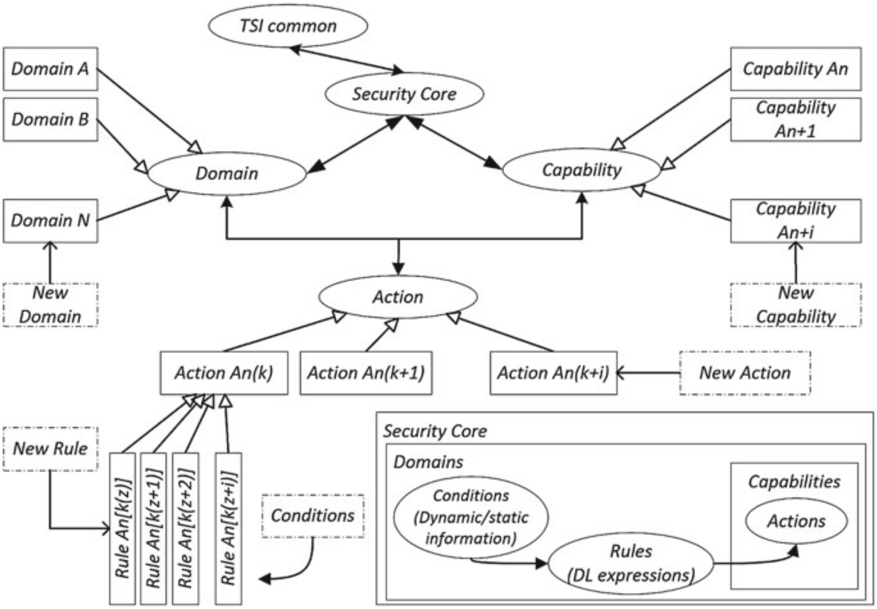


Fig. 1 Outline of security policy structure

evaluation of the environmental conditions, through the set of governing rules established for each action.

This framework permits the multi-domain and cross-layer implementation of security policies. Making use of the expressive power of description logic, complex relations can be established between the defined elements. Thus, actions within a specific capability can be linked to trigger the conditions evaluation of a rule established over a different domain. Additionally, conditions collected from various layers can affect decisions on other layers. Namely, a condition within the physical layer can affect a decision regarding the application layer.

The conceptualization of the policy framework is achieved by the use of unary and binary predicates, which are utilised to define the various network entities (data, services, users, terminals) and the relationships among them. Thus, a complete representation of the network can be achieved by defining the distinct constituting elements and their relations, as part of the tactical terminology. The tactical terminology is constructed within the T-Box with unique and acyclic concept definition, while the A-Box is used for instance identification with the use of concept and role assertions.

A detailed procedure for the ontological definition of security policies dedicated to tactical SOA was described earlier [35].

Table 1 Governing parameters for the distribution of security policies

Security policy distribution		
Ontology	Tactical nodes	Dynamism
1-Syntactic complexity	3-Operational specialization	6-Dynamic attributes
2-Structural complexity	4-Functional specialization	7-Dynamic policy evaluation
	5-Operating features	8-Tactical decision cycle

3 Constraint Analysis for the Distribution of Security Policies

Limiting the local scope of the security mechanisms in each tactical node, requires the identification of the parameters enabling the partitioning and distribution of security policies, within the context of tactical SOA. In the following sections, we present our findings regarding the identified parameters of critical impact, as they are presented in Table 1.

Our study over the functional characteristics of tactical SOA and the operation of ontologically defined security policies, promoted three main categories of governing parameters, regarding the attainment of the required horizontal and vertical security policy distribution. The first category refers to the evaluation of the policy, constructed based on the framework described in Fig. 1, regarding its overall and local complexity. The second category refers to the evaluation and categorization of the deployed tactical nodes, based on their expected functional and operational specialization, alongside their presumably known operating features. The last category refers to the sufficient integration of dynamism, emerging from the aforementioned characteristics of the tactical environment.

3.1 Complexity Inducing Components of Tactical Ontological Constructs

As highlighted earlier, the definition of the ontological security policy is unique for each tactical operation, constructed over an overlaying common framework (Fig. 1).

Regarding the syntactic complexity, OWL is provided in three increasingly expressive subsets that can be used for the definition of suitable security policies, namely OWL-Lite (Exp-time complete complexity), OWL-DL (NExp-time complete complexity) and OWL-Full (Undecidability). OWL-Lite supports simple constraint features and basic classification hierarchies. OWL-DL supports increased expressiveness, maintaining guaranteed computational completeness. Finally, OWL-Full provides maximum expressiveness and syntactic capabilities similar to RDF, yet reasoning is not reassured. A summary of the available constructs within OWL-Lite and OWL-DL is presented in Table 2 [37–39]. Furthermore, OWL 2 provides a

Table 2 Summary of available constructs within OWL-Lite and OWL-DL

OWL-Lite	
Category	Constructs
Constructors	Class, subClassOf, Property, subPropertyOf, domain, Individual
Restrictions	Restriction, allValuesFrom, someValuesFrom, intersectionOf
Equality	EquivalentClass, equivalentProperty, sameAs, differentFrom
Cardinality (0 or 1)	MinCardinality, maxCardinality
Properties	ObjectProperty, inverseOf, Datatype, Transitive, Symmetric, Functional, InverseFunctional
OWL-DL (In addition to the aforementioned)	
Values	HasValue
Cardinality (No limitation)	MinCardinality, maxCardinality
Class axioms	DisjointWith, equivalentClass, complementOf, subClassOf, unionOf, intersectionOf

wide set of subset profiles, supporting assorted accommodation between expressive power and reasoning efficiency. For instance, OWL 2 QL (NLogSpace complete complexity) is dedicated to efficiently supporting extensive instance data and database queries, OWL 2 RL (NP-time complete complexity) is optimized for scalable reasoning without fully utilizing the available expressive power, while OWL 2 EL (P-Time complete complexity) is suitable for large scale definition of properties and classes.

Regarding the structural complexity of the defined security policy, a variety of metrics with significant impact have been identified through our study. Their additive complexity overhead must be contemplated during the initial construction of the security policy, while they can be classified as:

1. **Vocabulary size**: The amount of the defined classes, individuals and properties.
2. **Impurity**: The deviation of the ontological structure from a pure tree form, as a result of the defined rdfs: subClassOf axioms.
3. **Mean inheritance**: The mean overall distance between the defined ancestor classes to the corresponding root classes.
4. **Connectivity**: A measurement of the connection density within the security policy, defined as the average number of connections for each of the defined elements (Classes and individuals).

Additionally, estimating the significance of individual classes over the overall functionality of the security policy, is pivotal for the identification of crucial distribution links within the policy structure. Such an estimation is possible with the use of the following metrics, for each of the defined classes.

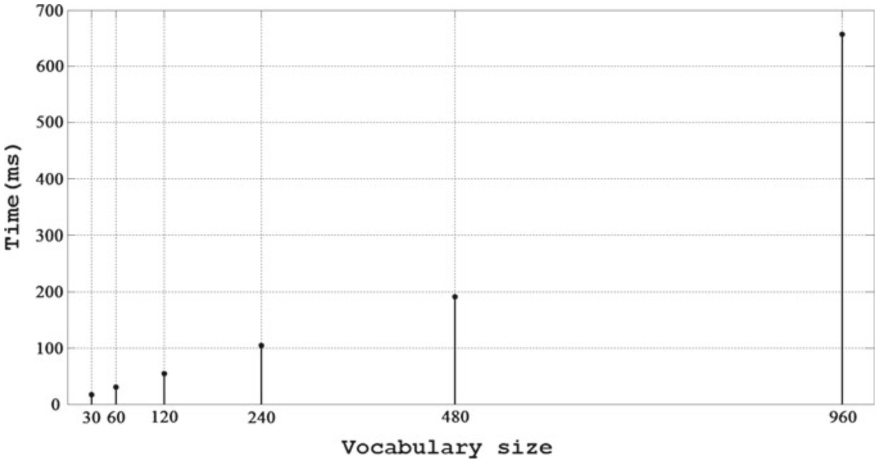


Fig. 2 Reasoning time escalation in relation to vocabulary size

1. **Direct inheritance**: The number of direct ancestors for each defined class. Meaning the number of subclasses defined based on a specific class and affected by changes within it.
2. **Inheritance exponentiation**: The depth of the most distant ancestor of a given class. It can be used as a measure of information inheritance within classes that belong to the same policy branch.
3. **Individual connectivity**: A connection density measure, referring to a specific class, calculated as the sum of the defined relations from and towards this class.

A representation of how these parameters affect the complexity of the security policy and the time required for reasoning over it, is provided in Fig. 2. In this set from our executed simulations, the Pellet reasoner is used over a basic ontological construct, structured using the ALC(D) fragment, in order to isolate and measure the impact of the value of the Vocabulary_size parameter. Furthermore, Fig. 3 provides an illustration of the global complexity estimation, based on the aforementioned combination of the propagating syntactic and local structural complexities.

3.2 Classification and Management of Tactical Nodes

Tactical nodes refer to a plethora of mobile platforms, with restricted operational characteristics and distinct requirements. Achieving a viable security policy distribution, requires the identification and incorporation of their influential attributes, for which we can attain a priori awareness. Our study over the characteristics of tactical nodes and the nature of tactical operations promoted three elements, of significant impact, as presented in Table 1.

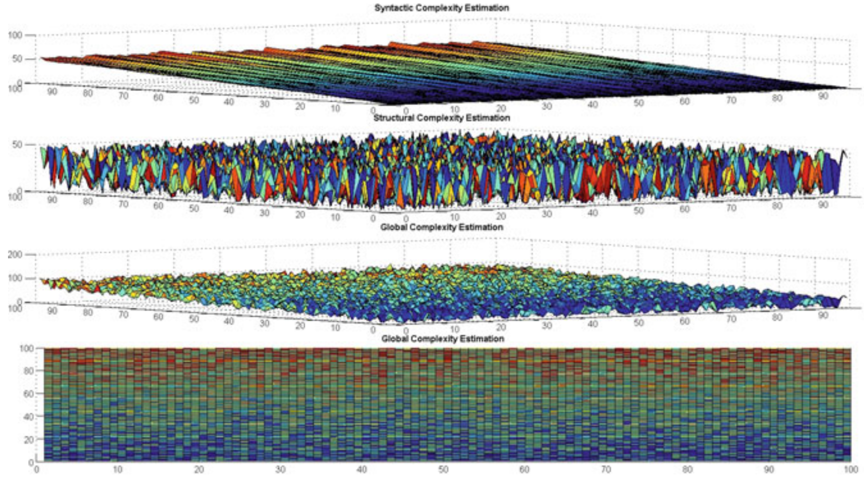


Fig. 3 Complexity estimation of tactical ontological constructs

The first two elements represent the operational and functional specialization of tactical nodes, rising through the initial operational and contingency planning of a tactical operation. The operational specialization refers to the identification of distinct operational groups among the entirety of the deployed assets, based on their particular strategic objectives. Additionally, functional node specialization, occurs due to the distinct roles of each node within the initial categorization into operational groups (e.g. Assuming a tactical team, the hand-held device of a medic, has distinct service/security requirements from the hand-held device of the team leader or a rifleman).

Hence, the defined operational and functional node specializations can provide an initial classification of nodes, in discrete groups with distinct yet entangled security requirements. This classification can form the basis for the horizontal (In terms of Domain/Capability groups) or vertical (In terms of Action/Rule groups), distribution of security policies, incorporating the operational perspective. A representation of the aforementioned procedure is presented in Fig. 4, based on our executed simulations. In this scenario, ten tactical nodes are organised in two operational groups (OG1-square, OG2-circle), while three functional groups (FG1-green(—), FG2-red(|), FG3-blue(\)) are globally defined.

An additional element that can significantly affect the distribution of security policies, within tactical SOA, is the presumably known operating features of tactical nodes. Tactical nodes refer to a variety of platforms, which may differ in various terms affecting their performance (Grouped afterwards as Computational Capacity). These elements can be classified as:

1. Computational power
2. Environmental limitations
3. Physical limitations
4. Resolution/accuracy limitations

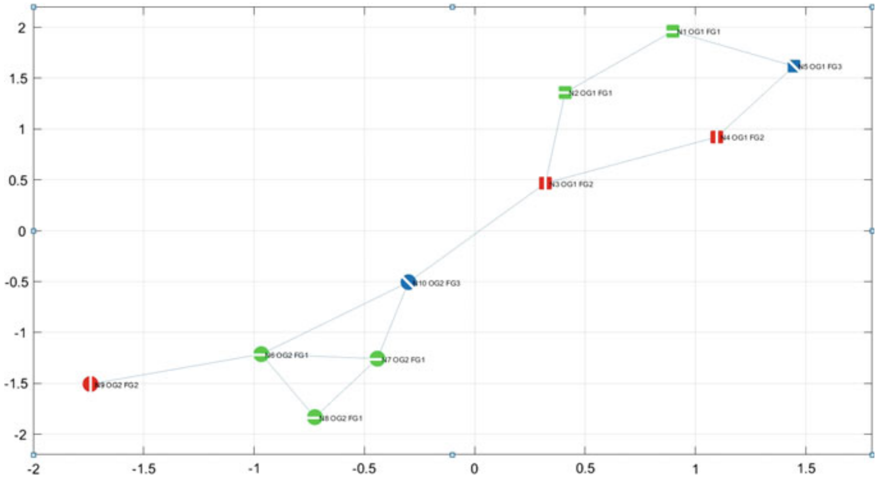


Fig. 4 Node classification based on operational and functional specialization

5. Input/output limitations
6. Range/coverage limitations
7. Network interconnection limitations

The knowledge of these parameters and their incorporation within the policy distribution decisions, can be used to enhance the network performance, in terms that include communication latency, service delivery/discovery and autonomy in case of partitioning, since they are correlated with the elements presented at Sect. 3.1.

3.3 *Incorporation of Dynamism*

The aforementioned characteristics of the tactical ecosystem, describe a highly dynamic and continuously evolving environment. Thus, the notion of dynamism has to be embodied, not only within the definition of the security policy, but also through the distribution mechanisms. For this reason, the realised security components must incorporate the available dynamic attributes across the network elements/domains, but also allow for the dynamic security policy evaluation, as presented at Sect. 2.

For the purpose of this study, achieving the efficient security policy distribution, also relies on the incorporation of a suitable tactical decision cycle. John Boyd's OODA (Observe, Orient, Decide, Act), is a decision cycle developed and used by military strategists, primarily within the strategic domain and the first two stages (Preparation, Execution) of combat operations, with additional applications to the third stage (Debrief/Evaluation). Evaluating the various suggested iterations of the OODA loop [40], the NCW targeted OODA model, proposed by Smith [41], emerged as the most suitable solution for tactical SOA, despite its complexity. Our decision was promoted by the fact that this model can coincide with suitably adjusted ontologically structured security policies, into the representation of complex and dynamic systems, providing in addition an enhanced level of granularity.

Similarly to the implementations within the strategic domain, the distinction between the involved processes (Observe, Orient, Decide, Act) and further segmentation to the defined domains (Physical, Information and Cognitive in Smith's model), can be eminently beneficial towards the technical implementation of a suitable distribution mechanism, within the tactical domain. Thus, the execution of the distinct processes of the decision cycle, can be delegated and distributed within the nodes of each operational group, allowing them to cooperatively reach the attainment of each objective, while dispensing the computational and overall cost. Additionally, the distribution of the involved processes, dispenses the required resources and time for the achievement of the optimality point, within the *Time Cost of Information* and *Decision Confidence/Quality* function, as described by Harrison [42].

4 Accommodation of the Defined Constraints for Security Policy Distribution

Having defined the overall security architecture and the critical parameters, for the distribution of security policies over tactical SOA, it is necessary to reconstruct the framework presented in Fig. 1, in accordance to the aforementioned criteria. This will allow the required minimization of the local policy scope in each tactical node, maintaining all the requisite functionalities. Additionally, this procedure will provide a transformation into a problem similar in nature to a 0-1 multiple knapsack problem, therefore subject to existing and widely studied optimization mechanisms. Furthermore, the incorporation of the identified elements, prior to the implementation of these mechanisms, will significantly increase the computational efficiency, due to the induced minimization of the number of instances.

Aiming to continuously support the required functionalities, within the defined security mechanisms:

1. Capabilities may span across various domains.
2. Actions may span across various capabilities.
3. A specific action within the context of different capabilities or domains, may be governed by a distinct set of rules.

Thus, a three dimensional space is required, in order to represent all the possible combinations of domains, capabilities and actions. The multitude of these ordered triplets constitutes the overall security policy of the tactical network, as presented in Fig. 5, while every individual action can be represented by a vector:

$$Action : A'_m = (D\hat{\imath} + C\hat{\jmath} + A\hat{k}), \; where \; \hat{\imath}, \hat{\jmath}, \hat{k} \; are \; unit \; vectors. \tag{1}$$

as presented in Fig. 6

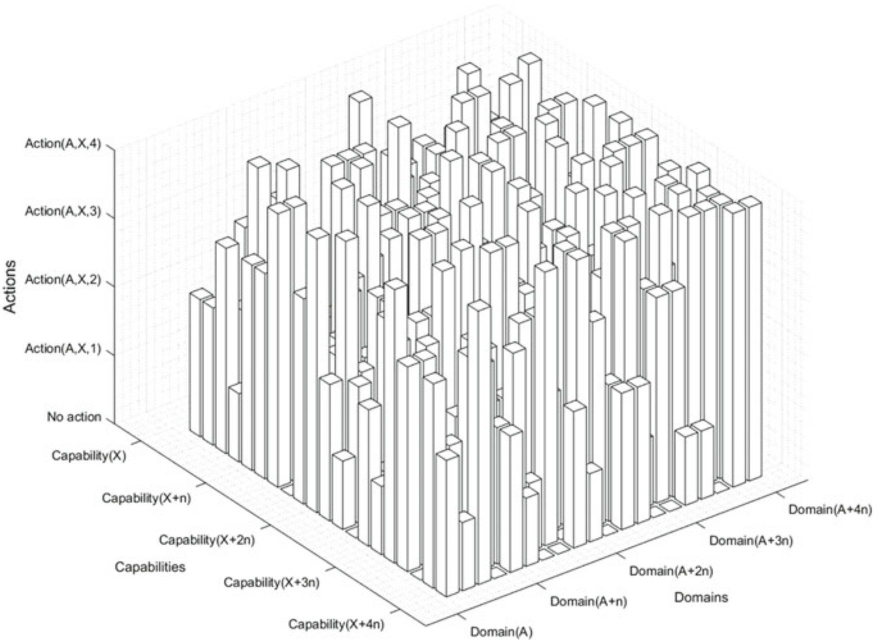


Fig. 5 Visualisation of a simplified security policy

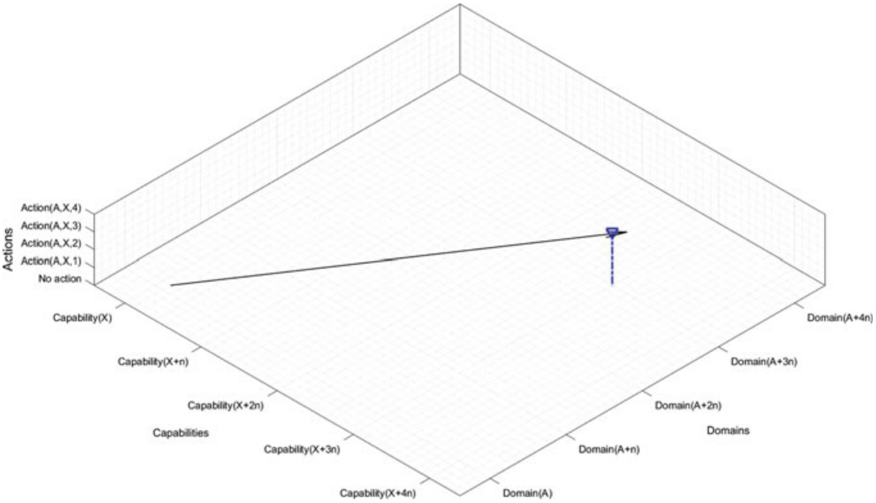


Fig. 6 Visualisation of a distinct action within the security policy

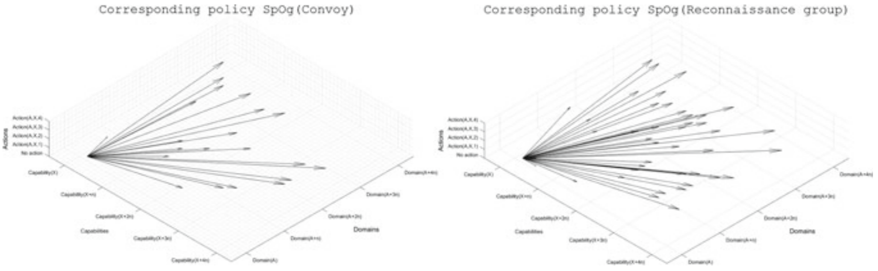


Fig. 7 Specimen security policy vector sets for convoy and reconnaissance operational groups

Due to the aforementioned constraints, mere replication of the entire security policy across all the deployed nodes is not sufficient. The incorporation of node operational specialization (Third identified element—Table 1), can provide an initial filtering, towards the minimization of the distributed policy branches. Thus, the specific operational contexts of the various deployed groups of nodes, correspond to a distinct set of basic vectors (Linearly independent), in the form:

$$Security\ policy\ :\ SpOg_{(x)} = \{A'_m, A'_{m+1},\ \dots\ , A'_{m+n}\} \tag{2}$$

This mapping is based on the required/estimated actions of each operational group, within each tactical operation, while it can be constructed a priori and automatically recalled when needed. For instance, a convoy operation may incorporate various operational groups including but not limited to the convoy, multiple protection groups and a medical evacuation group. The structure of the corresponding security policies, for each operational group, has a form similar to those presented in Fig. 7.

Yet, policy replication within an operational group is not the optimal solution, due to the node functional specialization (Fourth identified element—Table 1). Thus, the distinction between the functional groups of nodes across each given operational group, allows for further partitioning of the security policy as:

$$SpOg_{(x)} = SpFg_{(y)}\ \cup\ SpFg_{(y+1)}\ \cup\ \dots\ \cup\ SpFg_{(y+n)} \tag{3}$$

Hence, the security policy of a given operational group is defined as the union of the security policies of the functional groups that constitute it. This allows for the defined subsets ($SpFg_{(y)}$), to collectively compose or address distinct dimensions of the given $SpOg_{(x)}$. Yet, a given vector (Action: $A'_m = (D\hat{i} + C\hat{j} + A\hat{k})$) can span various subsets ($SpFg_{(y)}$) or be unique to one of them. A calculation of the sets intersections (e.g. $SpFg_{(y)} \cap SpFg_{(y+1)}$) and the sets differences (e.g. $SpFg_{(y)}/SpFg_{(y+1)}$), can provide a direct mapping between each action vector and the functional groups, across which it can be distributed, as:

$$\begin{array}{ll} SpFg_{(y)} = \{A'_1, A'_2, A'_3\} & \\ SpFg_{(y+1)} = \{A'_1, A'_3\} & \succ \\ SpFg_{(y+2)} = \{A'_2, A'_3, A'_4\} & \end{array} \qquad \begin{array}{l} A'_1 : Fg_{(y)}, Fg_{(y+1)} \\ A'_2 : Fg_{(y)}, Fg_{(y+2)} \\ A'_3 : Fg_{(y)}, Fg_{(y+1)}, Fg_{(y+2)} \\ A'_4 : Fg_{(y+2)} \end{array}$$

As presented in the defined security policy framework (Fig. 1), each vector $A'_m = (D\hat{i} + C\hat{j} + A\hat{k})$ corresponds to a set of governing rules, distinct for each individual action, enabling the dynamic adaptation of the security policy to alterations of the environmental conditions:

$$A'_m = \{R_{(z)}, R_{(z+1)}, \ \dots \ , R_{(z+n)}\} \tag{4}$$

Each rule is constructed making use of the expressive power of description logic, in order to incorporate the available static and dynamic attributes (Sixth identified element—Table 1) across the network, into the defined security policy decisions. Furthermore, as presented at Sect. 3.1, each rule caries an inherited complexity based on the values of the presented metrics, as a function of its syntactic and structural complexities (First and second identified elements—Table 1). Thus:

$$Vector\ complexity : CA'_m = \sum_{z=1}^{n} CR_{(z)} \tag{5}$$

Consequently, based on the operational features of the tactical nodes constituting each functional group (Fifth identified element—Table 1), suitable metrics incorporating their computational capacity (e.g. $CCFg_{(y)}$) can be constructed. Hence, given the aforementioned scenario, it is possible to construct a corresponding set of equations among the defined CA'_m and $CCFg_{(y)}$, as:

$$\begin{aligned} CA'_1 &= a * CCFg_{(y)} + b * CCFg_{(y+1)} \\ CA'_2 &= c * CCFg_{(y)} + d * CCFg_{(y+2)} \\ CA'_3 &= e * CCFg_{(y)} + f * CCFg_{(y+1)} + g * CCFg_{(y+2)} \\ CA'_4 &= h * CCFg_{(y+2)} \\ a + c + e &= 1 \\ b + f &= 1 \\ d + g + h &= 1 \end{aligned} \tag{6}$$

If the evaluation of the occurring equations is not feasible or a simplification of the process is required, assumptions can be made regarding the values of the variables, given the incorporation of the two additional identified elements of our study, namely:

1. Dynamic policy evaluation (Seventh identified element—Table 1): Meaning that the most suitable of the *available* rules, is dynamically selected to govern an action.
2. Decision cycle (Eighth identified element—Table 1): Meaning that (i) Gathering/storing the required rule inputs, (ii) Selecting the most suitable rule, (iii) Evaluating the selected rule, (iv) Enforcing the rule outcome, can be further distributed among the nodes constituting each functional group.

Thus, allowing for some additional flexibility regarding the exact values.

The utilization of the identified elements, as presented in this section, significantly limits the scale of the security policy distribution requirement, by identifying the maximum set of nodes responsible for a given set of actions (Equivalently: Minimizing the set of actions each node is responsible for). Having introduced the notions of CA'_m and $CCFg_{(y)}$, this has been limited to a problem similar in nature to a 0–1 knapsack problem in the following form.

Given for an action vector $A'_m = \{R_{(1)}, R_{(2)}, ..., R_{(n)}\}$ a finite set of rules, defined so $CR_{(1)} \leq CR_{(2)} \leq, ..., \leq CR_{(n)}$, and $SpFg_{(y)} = \{SpFg_{(1)}, SpFg_{(2)}, ..., SpFg_{(k)}\}$ a finite set of functional groups of tactical nodes with fixed capacities $CCFg_{(y)} = \{CCFg_{(1)}, CCFg_{(2)}, ..., CCFg_{(k)}\}$ (Calculated earlier as a percentage of their overall CC, dedicated to this action) and fixed 'k'. Assign each element of A'_m across the elements of $SpFg_{(y)}$ so:

1. The capacity of no element of $SpFg_{(y)}$ is exceeded.
2. No element of A'_m is duplicated within any given element of $SpFg_{(y)}$.
3. Duplicates of the elements of A'_m with minimum complexity, are allowed across the elements of $SpFg_{(y)}$, to increase redundancy.

Thus, given that:

1. $pR_{(j)}$ = Profit form $R_{(j)}$ (Requirement for a specific subset of rules).
2. $CR_{(j)}$ = Complexity of $R_{(j)}$.
3. $CCFg_{(i)}$ = The calculated percentage of each CC dedicated to this action.

Then maximize:

$$D = \sum_{i=1}^{k} \sum_{j=1}^{n} pR_{(j)} * X_{ij} \tag{7}$$

Subject to:

$$\sum_{j=1}^{n} CR_{(j)} * X_{ij} \leq CCFg_{(i)}, \quad i = [1, ..., k] \tag{8}$$

$$\sum_{j=1}^{n} X_{ij} = 1, \quad i = [1, ..., k] \tag{9}$$

$$X_{ij} = 1 \; or \; 0, \quad i = [1, ..., k], j = [1, ..., n] \tag{10}$$

where:

$$X_{ij} = \begin{cases} 1 \text{ if } R_{(j)} \text{ is selected for } Fg_{(i)}, \\ 0 \text{ if not} \end{cases}$$

A variety of exact and heuristic algorithms has been developed for the attainment of optimal/near optimal solutions for this type of problems [43–54]. The average solution time of these algorithms is directly correlated to the number of instances [55, 56], which with the incorporation of the defined parameters, has been limited to a minimum set of rules for each node, maintaining at the same time support of all the required functionalities within a tactical operation.

It must also be stated that the described procedure is executed at the mission preparation stage, facing no computational, time, communication or other type of limitations. In this manner, we can achieve a mapping between the required and the available computational power achieving optimal policy partitioning and distribution, incorporating all the corresponding elements of significant impact.

5 Conclusions

Through this article, the findings of our study regarding the parameters governing the partitioning and distribution of security policies within tactical SOA, have been presented. Evaluating the characteristics of tactical networks and utilized actors, the involved elements of critical impact, have been identified and analysed. Furthermore, a suitable mechanism has been suggested, accommodating the identified parameters, for the optimum partitioning and distribution of security policies within the mission preparation stage.

Our future plans include the further refinement and evaluation of the proposed mechanism for the mission preparation stage and its extension within the mission execution stage, in the presence of additional constraints, such as connectivity and bandwidth availability. More precisely the utilisation of hierarchical structures within the defined rule sets, governing the individual actions, and the constrained optimization for online distribution of both security policies and governing conditions. Furthermore, we intent to identify suitable mechanisms for the reconciliation of security policies, adjusted to the dynamics of tactical SOA.

Acknowledgements The results described in this work were obtained as part of the European Defence Agency project TACTICS (Tactical Service Oriented Architecture). The TACTICS project is jointly undertaken by Patria (FI), Thales Communications & Security (FR), Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie FKIE (DE), Thales Deutschland (DE), Finmeccanica (IT), Thales Italia (IT), Norwegian University of Science and Technology (NO), ITTI (PL), Military Communication Institute (PL), and their partners, supported by the respective national Ministries of Defence under EDA Contract No. B 0980.

References

1. Horne, G., Leonardi, M.: Maneuver warfare science 2001 (2001)
2. Bar-Noy, A., Cirincione, G. Govindan, R. Krishnamurthy, S., LaPorta, T., Mohapatra, P., Neely, M., Yener, A.: Quality-of-information aware networking for tactical military networks. In: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 2–7, Mar 2011
3. Burbank, J., Chimento, P., Haberman, B., Kasch, W.: Key challenges of military tactical networking and the elusive promise of manet technology. IEEE Commun. Mag. **44**, 39–45 (2006)
4. Elmasry, G.: A comparative review of commercial versus tactical wireless networks. IEEE Commun. Mag. **48**, 54–59 (2010)
5. Shi, V.: Evaluating the performability of tactical communications networks. IEEE Trans. Veh. Technol. **53**, 253–260 (2004)
6. Moffat, J.: Adapting Modeling & Simulation for Network Enabled Operations (2011)
7. Alberts, D.S., Hayes, R.E.: Power to the Edge (2003)
8. Smith, E.A.: Complexity, Networking, and Effects-Based Approaches to Operations (2006)
9. Lund, K., Eggen, A., Hadzic, D., Hafsoe, T., Johnsen, F.: Using web services to realize service oriented architecture in military communication networks. IEEE Commun. Mag. **45**, 47–53 (2007)
10. Johnsen, F., Bloebaum, T., Schenkels, L., Fiske, R., Van Selm, M., de Sortis, V., van der Zanden, A., Sliwa, J., Caban, P.: Soa over disadvantaged grids experiment and demonstrator. In: Communications and Information Systems Conference (MCC), 2012 Military, pp. 1–8, Oct 2012
11. Suri, N.: Dynamic service-oriented architectures for tactical edge networks. In: Proceedings of the 4th Workshop on Emerging Web Services Technology, WEWST ’09, pp. 3–10. ACM, New York, NY, USA (2009)
12. IST-090 Task Group, Service oriented architecture (SOA) challenges for real time and disadvantaged grid (IST-090). https://www.cso.nato.int/Activity_Meta.asp?ACT=1830, Apr 2014
13. IST-118 Task Group, SOA recommendations for disadvantaged grids in the tactical domain (IST-118). https://www.cso.nato.int/ACTIVITY_META.asp?ACT=2293
14. Maule, R., Lewis, W.: Security for distributed soa at the tactical edge. In: Military Communications Conference, 2010—MILCOM 2010, pp. 13–18, Oct 2010
15. Mayott, G., Self, M., Miller, G.J., McDonnell, J.S.: Soa approach to battle command: simulation interoperability (2010)
16. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The ponder policy specification language. In: Sloman, M., Lupu, E., Lobo, J. (eds.) Policies for Distributed Systems and Networks, Lecture Notes in Computer Science, vol. 1995, pp. 18–38. Springer, Berlin (2001)
17. OASIS, Oasis security services (saml) tc
18. Ramli, C.D.P.K., Nielson, H.R., Nielson, F.: The logic of XACML. Sci. Comput. Prog. **83**, 80–105 (2014)
19. Li, N., Mitchell, J., Winsborough, W.: Design of a role-based trust-management framework. In: Proceedings. 2002 IEEE Symposium on Security and Privacy, 2002, pp. 114–130 (2002)
20. Becker, M., Sewell, P.: Cassandra: distributed access control policies with tunable expressiveness. In: Proceedings. Fifth IEEE International Workshop on Policies for Distributed Systems and Networks, 2004. POLICY 2004, pp. 159–168, June 2004
21. Nejdl, W., Olmedilla, D., Winslett, M.: Peertrust: Automated trust negotiation for peers on the semantic web. In: Jonker, W., Petkovi, M. (eds.) Secure Data Management, Lecture Notes in Computer Science, vol. 3178 pp. 118–132. Springer, Berlin (2004)
22. Czenko, M., Doumen, J., Etalle, S.: Trust management in p2p systems using standard tulip. In: Karabulut, Y., Mitchell, J., Herrmann, P., Jensen, C. (eds.) Trust Management II, FIP The International Federation for Information Processing, vol. 263 pp. 1–16, Springer, US (2008)
23. Finin, T., Joshi, A., Kagal, L., Niu, J., Sandhu, R., Winsborough, W.H., Thuraisingham, B.: ROWLBAC—representing role based access control in OWL. In: Proceedings of the 13th

Symposium on Access control Models and Technologies, ACM Press, Estes Park, Colorado, USA, June 2008

24. Kagal, L., Finin, T., Paolucci, M., Srinivasan, N., Sycara, K., Denker, G.: Authorization and privacy for semantic web services. IEEE Intell. Syst. **19**, 50–56 (2004)
25. Uszok, A., Bradshaw, J.M., Johnson, M., Jeffers, R., Tate, A., Dalton, J., Aitken, S.: Kaos policy management for semantic web services. IEEE Intell. Syst. **19**, 32–41 (2004)
26. Kolter, J., Schillinger, R., Pernul, G.: Building a distributed semantic-aware security architecture. In: Venter, H., Eloff, M., Labuschagne, L., Eloff, J., von Solms, R. (eds.) New Approaches for Security, Privacy and Trust in Complex Environments, IFIP International Federation for Information Processing, vol. 232 pp. 397–408. Springer, US (2007)
27. Ferrini, R., Bertino, E.: Supporting RBAC with XACML + OWL. In: Carminati, B., Joshi, J. (eds.) Proceedings of the 14th ACM Symposium on Access Control Models and Technologies (SACMAT '09), pp. 145–154. ACM Press, Stresa, Italy, June 2009
28. Ben Brahim, M., Chaari, T., Ben Jemaa, M., Jmaiel, M.: Semantic matching of ws-securitypolicy assertions. In: Pallis, G., Jmaiel, M., Charfi, A., Graupner, S., Karabulut, Y., Guinea, S., Rosenberg, F., Sheng, Q., Pautasso, C., Ben Mokhtar, S. (eds.) Service-Oriented Computing-ICSOC 2011 Workshops, Lecture Notes in Computer Science, vol. 7221, pp. 114–130. Springer, Berlin (2012)
29. Helil, N., Rahman, K.: Extending xacml profile for rbac with semantic concepts. In: 2010 International Conference on Computer Application and System Modeling (ICCASM), vol. 10, pp. V10–69–V10–74, Oct 2010
30. Blanco, C., Lasheras, J., Valencia-Garcia, R., Fernandez-Medina, E., Toval, A., Piattini, M.: A systematic review and comparison of security ontologies. In: 3rd International Conference on Availability, Reliability and Security, 2008. ARES 08, pp. 813–820, Mar 2008
31. Souag, A., Salinesi, C., Comyn-Wattiau, I.: Ontologies for security requirements: a literature survey and classification. In: Bajec, M., Eder, J. (eds.) Advanced Information Systems Engineering Workshops, Lecture Notes in Business Information Processing, vol. 112 pp. 61–69. Springer, Berlin (2012)
32. Nguyen, V.: Ontologies and information systems: a literature survey 6 (2011)
33. Kolovski, V., Parsia, B., Katz, Y., Hendler, J.: Representing web service policies in owl-dl. In: International Semantic Web Conference (ISWC), pp. 6–10 (2005)
34. Trivellato, D., Zannone, N., Glaundrup, M., Skowronek, J., Etalle, P.S.: A semantic security framework for systems of systems. Int. J. Coop. Inf. Syst. **22**, 1–35 (2013)
35. Gkioulos, V., Wolthusen, S.D.: Enabling dynamic security policy evaluation for service-oriented architectures in tactical networks. In: Accepted for presentation at Norwegian Information Security Conference 2015 (NISK-2015) (2015)
36. Nato Architecture Framework, NATO Capability View, NAF v3 NCV-2, June 2013
37. W3C Recommendation, OWL2-OVERVIEW OWL 2 Web Ontology Language Document Overview, 2nd Edn. http://www.w3.org/TR/2012/REC-owl2-overview-20121211/. Dec 2012
38. Schneider, P.F.P., Hayes, P., Horrocks, I.: OWL-SEMANTICS OWL Web Ontology Language Semantics and Abstract Syntax. http://www.w3.org/TR/2004/REC-owl-semantics-20040210/. Feb 2004
39. Motik, B., Grau, B.C., Horrocks, I., Wu, Z., Fokoue, A.: OWL2-PROFILES OWL 2 Web Ontology Language Profiles, 2nd edn. http://www.w3.org/TR/2012/REC-owl2-profiles-20121211/. Dec 2012
40. Breton, R., Rousseau, R.: The future of c2 the c-ooda: a cognitive version of the ooda loop to represent c2 activities. Topic: C2 process modelling
41. Smith, E.A.: Effects Based Operations. Crisis, and War. Center for Advanced Concepts and Technology, Applying Network Centric Warfare in Peace (2002)
42. Harrison, F.: The Managerial Decision-Making Process-5th edn. South-Western College Pub (1998)
43. Veni, K.K., Balachandar, S.R.: Int. J. Math. Comput. Phys. Electr. Comput. Eng. **4**(7), 1044–1048 (2010)

44. Balas, E., Glover, F., Zionts, S.: An additive algorithm for solving linear programs with zero-one variables. Oper. Res. **13**(4), 517–549 (1965)
45. Frville, A.: The multidimensional 01 knapsack problem: an overview. Eur. J. Oper. Res. **155**(1), 1–21 (2004)
46. Ohkura, K., Igarashi, T., Ueda, K., Okauchi, S., Matsunaga, H.: A genetic algorithm approach to large scale combinatorial optimization problems in the advertising industry. In: Proceedings. 2001 8th IEEE International Conference on Emerging Technologies and Factory Automation, 2001, vol. 2, pp. 351–357, Oct 2001
47. Chu, P., Beasley, J.: A genetic algorithm for the multidimensional knapsack problem. J Heuristics **4**(1), 63–86 (1998)
48. Balas, E., Martin, C.H.: Pivot and complement heuristic for 0–1 programming. Manag. Sci. **26**(1), 86–96 (1980)
49. Gavish, B., Pirkul, H.: Efficient algorithms for solving multiconstraint zero-one knapsack problems to optimality. Math. Program. **31**(1), 78–105 (1985)
50. Gilmore, P.C., Gomory, R.E.: The theory and computation of knapsack functions. Oper. Res. **14**(6), 1045–1074 (1966)
51. Osorio, M., Glover, F., Hammer, P.: Cutting and surrogate constraint analysis for improved multidimensional knapsack solutions. Ann. Oper. Res. **117**(1–4), 71–93 (2002)
52. Magazine, M., Oguz, O.: A heuristic algorithm for the multidimensional zero-one knapsack problem. Eur. J. Oper. Res. **16**(3), 319–326 (1984)
53. Volgenant, A., Zoon, J.A.: An improved heuristic for multidimensional 0–1 knapsack problems. J. Oper. Res. Soc. **41**(10), 963–970 (1990)
54. Weingartner, H.M., Ness, D.N.: Methods for the solution of the multidimensional 0/1 knapsack problem. Oper. Res. **15**(1), 83–103 (1967)
55. Caccetta, L., Kulanoot, A.: Computational aspects of hard knapsack problems. In: Proceedings of the Third World Congress of Nonlinear Analysts Nonlinear Analysis: Theory, Methods & Applications, vol. 47, no. 8, pp. 5547–5558 (2001)
56. Pisinger, D.: Where are the hard knapsack problems? Comput. Oper. Res. **32**(9), 2271–2284 (2005)

Practical Extensions of Trust Management Credentials

Anna Felkner and Adam Kozakiewicz

Abstract Trust management is a unified approach to access control in open distributed systems, where decisions connected with access control are based on policy statements made by many principals. The family of Role-based Trust management languages (RT) is an effective means for representing security policies, credentials and relationship in distributed, decentralized, large scale access control systems. It delivers a set of role assignment credentials and is used in systems where the identities of users are not the most important form of identification. A credential gives information about the privileges of users and the security policies issued by (usually more than one) trusted authorities. The main purpose of this article is to show how some credential extensions can make a trust management system more useful in practice. It shows how security systems can be made more realistic by maintaining the procedure or including timing information.

1 Introduction

This paper is an expanded and updated version of the paper “More Practical Application of Trust Management Credentials” [6], presented at the 3rd International Conference on Innovative Network Systems and Applications within multi-conference 2015 Federated Conference on Computer Science and Information Systems.

Most access control systems, based on traditional access control models (such as mandatory (MAC), discretionary (DAC) and Role Based Access Control (RBAC)), are in essence identity based. Access control decisions are based entirely on the role or identity of the requesting party and can only be made if the petitioner knowns the owner of the resource. The decisions are based on the relation of the only two entities involved—the owner of the protected resource, who is responsible for granting access, and the petitioner, who requires access. Unfortunately, such a simple, bilat-

A. Felkner · A. Kozakiewicz (✉)
Research and Academic Computer Network, Kolska 12, 01-045 Warsaw, Poland
e-mail: adam.kozakiewicz@nask.pl

A. Felkner
e-mail: anna.felkner@gmail.com

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent
Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_10

eral approach means that possibility of applying these solutions in modern systems is very limited.

In closed, centralized environments this approach works properly. As identity of the users of the system is established in advance, basing access decisions on it is a natural and easy to implement choice. Unfortunately it does not scale well as the system becomes highly distributed over a large network. In such open decentralized systems the set of users is rather large and usually may change dynamically. That introduces entirely new challenges, like the problem of propagation of change information. It can be partially solved by a central database of user's identities, but it creates a single point of failure, where temporary lack of access to the database makes some authorization decisions impossible to take. In absence of such a central repository of identification data it is no longer possible to assume that the identities of the proponent and the resource owner are reciprocally recognized. More complex solutions are needed. One of them is called trust management.

Consider a simple example where one of the aspects of bookstore policy says that anyone who is a student and has a bookstore loyalty card, is eligible for a discount. An electronic bookstore system must enable a bookstore to state the rule that proving the status of a student and ownership of a bookstore loyalty card results in a discount (*policy*). It must enable a client to prove that he/she has *credentials* stating that he/she is indeed a student and has a bookstore loyalty card and it must enable to specify what persons or institutions may issue such credentials (*trust relationship*). The access rights are determined based on the sufficiently documented information about the user's rights, assigned by third party, not on user identity.

The rest of this paper is organized as follows. Trust management concept is shown in Sect. 2. Section 3 presents inference system over RT^T credentials. Section 4 describes a few extensions of RT^T language (time validity and determination of the order). Final remarks are given in the Conclusions.

2 Trust Management

The first definition of *trust management* was introduced by Blaze et al. [1] in 1996. It was defined as a unified approach to specify and interpret security policies, credentials and trust relationships. The privileges of entities in such a system are based not on their identity, but on their attributes. There can be a lot of principals, who have the right to issue credentials which are then used to demonstrate the entity's attributes. A *credential* is defined as an attestation of competence, authority or qualification of an individual issued by a third party. Credentials include such information as privileges of a given user and security policies issued by some trusted authorities. Real life examples of credentials are academic diplomas, identification documents, certificates, driver's licenses, membership cards—all kinds of things giving some privileges.

2.1 The Family of Role-Based Trust Management Languages

Role-based Trust management family languages are used for representing security policies, trust relationships and credentials in distributed, decentralized access control systems. Depending on the language, several types of role assignment credentials are provided in *RT* languages.

One of the main advantages of the trust management approach is the ability to use *delegation*. A principal's authority over a resource may be transferred in a limited way to other principals by means of a credential.

This work extends the possibility of practical usage of trust management languages. It shows how we can make RT family languages more useful in practice by introducing the order of entities which can appear in the execution context and by including time validity constraints.

Most fundamental credentials were introduced in $\mathbf{RT_0}$ [8]. It forms the core part of the family, providing basic abilities—localization of authority for roles, delegation of that authority, role hierarchies and role intersections. These features are available in all *RT* languages, which extend this set with additional features. Parametrized roles were introduced in $\mathbf{RT_1}$. This language enables representation of relationships between entities. $\mathbf{RT_2}$ extends RT_1 with the notion of logical objects—it enables simple assignment of access rights for entire groups of logically related objects (resources). Both extensions presented so far do not change the expressive power of the language. They allow much more concise notation, but the same policy can in fact be expressed in RT_0. The next language, and the first actually adding new capabilities not present in other members of the family is the $\mathbf{RT^T}$ language, the main focus of this paper. It includes the ability to express agreement of multiple principals, even from disjoint sets, via manifold roles and separation of duties or threshold policies, via role-product operators.

Role-based Trust management family of languages uses a set of basic elements, such as entities, roles, role names and credentials. *Entities* are principals controlling access to resources by defining roles and issuing credentials as well as petitioners willing to access resources. The entity can be a person or application. *Roles* represent sets of entities for which the access control policies grant particular permissions. *Role names* represent permissions and can be issued to entities or groups of them by other entities. *Credentials* define roles by appointing a new member of the role or by delegating authority to members of other roles.

RT^T contains six different types of credentials, the first four in common with the rest of the languages from the *RT* family:

$A.r \leftarrow B$ — entity B is a member of role $A.r$

$A.r \leftarrow B.s$ — role $A.r$ includes all members of role $B.s$. This type of credential also defines role hierarchies.

$A.r \leftarrow B.s.t$ — role $A.r$ includes role $C.t$ for each C, which is a member of role $B.s$.

$A.r \leftarrow B.s \cap C.t$ — role $A.r$ includes all members of both roles $B.s$ and $C.t$. This is a partial delegation from A to B and C.

$A.r \leftarrow B.s \odot C.t$ — role $A.r$ can be satisfied by a union set containing one member of each of the two roles ($B.s$ and $C.t$). A single entity being a member of both roles suffices.

$A.r \leftarrow B.s \otimes C.t$ — role $A.r$ includes two *different* entities, one of which is a member of role $B.s$ and one a member of role $C.t$.

The models used in practice can be very complex, but this paper uses some simplified examples. We focus on RT^T credentials and our intention is to illustrate the basic notions and notation, not the full expressive power of the language.

Example 1 (*Example of RT^T—subject*) Suppose that a university will activate the subject if at least two out of four students and one PhD student apply. Using RT_0 credentials, we have to list all the students (four in this simple case) and choose two of them and the policy has to be changed whenever the list of students changes. RT^T allows us to use only one credential to express this rule. Further, we need to have one PhD student, who also can (but does not have to) be a regular student. This requires only one more RT^T credential. The entire policy can be expressed as follows:

$$F.students \leftarrow F.student \otimes F.student \tag{1}$$

$$F.activeSubject \leftarrow F.students \odot F.phdStudent \tag{2}$$

Now, when we add the following credentials:

$$F.student \leftarrow \{Alicia\} \tag{3}$$

$$F.student \leftarrow \{Carol\} \tag{4}$$

$$F.student \leftarrow \{Marry\} \tag{5}$$

$$F.student \leftarrow \{Greg\} \tag{6}$$

$$F.phdStudent \leftarrow \{Greg\} \tag{7}$$

$$F.phdStudent \leftarrow \{Marc\} \tag{8}$$

we can conclude that, according to the policy, any pair of students from the set $\{Alicia, Carol, Marry, Greg\}$ is sufficient to meet the role $F.students$, and to activate the subject it is required that *Marc* must attend unless one of the students is $Greg$.

Example 2 (*Example of RT^T—signature*) Suppose that we have a company, named Comp. Its security policy states that we need to collect five signatures to accept some transaction. In our example it could be the signatures of the petitioner, company accountant, his official superior, the manager of financial department of the company and the director of a company. This policy can be represented using the following credential:

$$Comp.signature \leftarrow Comp.petitioner \odot Comp.accountant$$

$$\odot\ Comp.superior \odot Comp.fdManager \odot Comp.director \tag{9}$$

Now suppose that we have people, who play those roles, so the following credentials have been added to our security policy:

$$Comp.petitioner \leftarrow \{Alex\} \tag{10}$$

$$Comp.accountant \leftarrow \{Alex\} \tag{11}$$

$$Comp.accountant \leftarrow \{Betty\} \tag{12}$$

$$Comp.accountant \leftarrow \{John\} \tag{13}$$

$$Comp.superior \leftarrow \{David\} \tag{14}$$

$$Comp.superior \leftarrow \{Jacob\} \tag{15}$$

$$Comp.fdManager \leftarrow \{Alex\} \tag{16}$$

$$Comp.director \leftarrow \{David\} \tag{17}$$

As we can see, to complete the set of signatures we need just two people: *Alex*, who can play a role of *petitioner*, *accountant*, *fdManager* and *David* who plays the role of *superior* and *director*, but as may be required, groups of people {*Alex*, *Betty*, *David*}, {*Alex*, *Betty*, *Jacob*, *David*}, {*Alex*, *John*, *David*}, and {*Alex*, *John*, *Jacob*, *David*} can also play a manifold role, and cooperatively complete the set of signatures.

3 Inference System over RT^T Credentials

RT^T credentials define roles, which in turn are used to define permissions. The set of member entities for a role is defined by a set $\mathscr{P}$ of RT^T credentials. This set can be calculated using an inference system, which defines an operational semantics of RT^T language. The system consists of a set of inference rules used to infer credentials from existing ones and an initial set of formulae considered true.

Let $\mathscr{P}$ be a set of RT^T credentials. The inference rules can be applied to create new credentials, derived from credentials of the set $\mathscr{P}$. A derived credential c will be denoted using a formula $\mathscr{P} \succ c$, meaning that credential c can be derived from a set of credentials $\mathscr{P}$.

The initial set of formulae of an inference system over a set $\mathscr{P}$ of RT^T credentials are all the formulae: $c \in \mathscr{P}$ for each credential c in $\mathscr{P}$. The inference rules of the system are the following:

$$\frac{c \in \mathscr{P}}{\mathscr{P} \succ c} \tag{$\mathbf{W_1}$}$$

$$\frac{\mathscr{P} \succ A.r \leftarrow B.s \quad \mathscr{P} \succ B.s \leftarrow X}{\mathscr{P} \succ A.r \leftarrow X} \tag{$\mathbf{W_2}$}$$

$$\frac{\mathscr{P} \succ A.r \leftarrow B.s.t \quad \mathscr{P} \succ B.s \leftarrow C \quad \mathscr{P} \succ C.t \leftarrow X}{\mathscr{P} \succ A.r \leftarrow X} \tag{$\mathbf{W_3}$}$$

$$\frac{\mathscr{P} \succ A.r \leftarrow B.s \cap C.t \quad \mathscr{P} \succ B.s \leftarrow X \quad \mathscr{P} \succ C.t \leftarrow X}{\mathscr{P} \succ A.r \leftarrow X} \tag{$\mathbf{W_4}$}$$

$$\frac{\mathscr{P} \succ A.r \leftarrow B.s \odot C.t \quad \mathscr{P} \succ B.s \leftarrow X \quad \mathscr{P} \succ C.t \leftarrow Y}{\mathscr{P} \succ A.r \leftarrow X \cup Y} \tag{$\mathbf{W_5}$}$$

$$\frac{\mathscr{P} \succ A.r \leftarrow B.s \otimes C.t \quad \mathscr{P} \succ B.s \leftarrow X \quad \mathscr{P} \succ C.t \leftarrow Y \quad X \cap Y = \phi}{\mathscr{P} \succ A.r \leftarrow X \cup Y} \tag{$\mathbf{W_6}$}$$

Many different inference systems of a given language could be defined. Two properties of the system are required for it to be useful in practice—it must be sound and complete. Soundness guarantees that any formula derived by the system must be valid with respect to the semantics of the language, while completeness ensures that any valid formula is derivable.

The system presented above is sound and complete. Proofs of both properties can be found in [3], quaranteeing that the inference system is a valid alternative way of presenting the semantics of RT^T.

Example 3 (*Inference system for Example* 2) We will now use an inference system to derive the set of entities that can cooperate to accept that transaction, i.e. the signatures of the petitioner, company accountant, his official superior, the manager of financial department of the company and the director of a company, using a limited set of credentials for brevity:

Using credentials (9)–(17) according to the rule (W_1) we can infer:

$$\frac{\begin{array}{c}\mathit{Comp.signature} \leftarrow \mathit{Comp.petitioner} \odot \mathit{Comp.accountant} \odot \mathit{Comp.superior}\\ \odot\, \mathit{Comp.fdManager} \odot \mathit{Comp.director} \in \mathscr{P}\end{array}}{\begin{array}{c}\mathscr{P} \succ \mathit{Comp.signature} \leftarrow \mathit{Comp.petitioner} \odot \mathit{Comp.accountant} \odot \mathit{Comp.superior}\\ \odot\, \mathit{Comp.fdManager} \odot \mathit{Comp.director}\end{array}}$$

$$\frac{\mathit{Comp.petitioner} \leftarrow \{\mathit{Alex}\} \in \mathscr{P}}{\mathscr{P} \succ \mathit{Comp.petitioner} \leftarrow \{\mathit{Alex}\}}$$

$$\frac{\mathit{Comp.accountant} \leftarrow \{\mathit{Alex}\} \in \mathscr{P}}{\mathscr{P} \succ \mathit{Comp.accountant} \leftarrow \{\mathit{Alex}\}}$$

$$\frac{\mathit{Comp.accountant} \leftarrow \{\mathit{Betty}\} \in \mathscr{P}}{\mathscr{P} \succ \mathit{Comp.accountant} \leftarrow \{\mathit{Betty}\}}$$

$$\frac{\mathit{Comp.accountant} \leftarrow \{\mathit{John}\} \in \mathscr{P}}{\mathscr{P} \succ \mathit{Comp.accountant} \leftarrow \{\mathit{John}\}}$$

$$\frac{\mathit{Comp.superior} \leftarrow \{\mathit{David}\} \in \mathscr{P}}{\mathscr{P} \succ \mathit{Comp.superior} \leftarrow \{\mathit{David}\}}$$

$$\frac{\mathit{Comp.superior} \leftarrow \{\mathit{Jacob}\} \in \mathscr{P}}{\mathscr{P} \succ \mathit{Comp.superior} \leftarrow \{\mathit{Jacob}\}}$$

$$\frac{\mathit{Comp.fdManager} \leftarrow \{\mathit{Alex}\} \in \mathscr{P}}{\mathscr{P} \succ \mathit{Comp.fdManager} \leftarrow \{\mathit{Alex}\}}$$

$$\frac{\mathit{Comp.director} \leftarrow \{\mathit{David}\} \in \mathscr{P}}{\mathscr{P} \succ \mathit{Comp.director} \leftarrow \{\mathit{David}\}}$$

Then, using credentials (9), (10), (11), (14), (16) and (17) and rule (W_5) we infer:

$$\frac{\begin{array}{c}\mathscr{P} \succ \mathit{Comp.signature} \leftarrow \mathit{Comp.petitioner} \odot \mathit{Comp.accountant} \odot \mathit{Comp.superior}\\ \odot\, \mathit{Comp.fdManager} \odot \mathit{Comp.director}\\ \mathscr{P} \succ \mathit{Comp.petitioner} \leftarrow \{\mathit{Alex}\} \qquad \mathscr{P} \succ \mathit{Comp.accountant} \leftarrow \{\mathit{Alex}\}\\ \mathscr{P} \succ \mathit{Comp.superior} \leftarrow \{\mathit{David}\} \qquad \mathscr{P} \succ \mathit{Comp.fdManager} \leftarrow \{\mathit{Alex}\}\\ \mathscr{P} \succ \mathit{Comp.director} \leftarrow \{\mathit{David}\}\end{array}}{\mathscr{P} \succ \mathbf{Comp.signature} \leftarrow \{\mathbf{Alex}, \mathbf{David}\}}$$

showing that the set of entities $\{\mathit{Alex}, \mathit{David}\}$ is sufficient to complete the set of signatures.

Or, using credentials (9), (10), (12), (14), (16) and (17) and rule (W_5) we infer:

$$\frac{\begin{array}{c}\mathscr{P} \succ Comp.signature \leftarrow Comp.petitioner \odot Comp.accountant \odot Comp.superior \\ \odot\, Comp.fdManager \odot Comp.director \\ \mathscr{P} \succ Comp.petitioner \leftarrow \{Alex\} \qquad \mathscr{P} \succ Comp.accountant \leftarrow \{Betty\} \\ \mathscr{P} \succ Comp.superior \leftarrow \{David\} \qquad \mathscr{P} \succ Comp.fdManager \leftarrow \{Alex\} \\ \mathscr{P} \succ Comp.director \leftarrow \{David\}\end{array}}{\mathscr{P} \succ \mathbf{Comp.signature} \leftarrow \{\mathbf{Alex}, \mathbf{Betty}, \mathbf{David}\}}$$

showing that here we need more people to complete the set of signatures, i.e. $\{Alex, Betty, David\}$.
Or, using credentials (9), (10), (13), (15), (16) and (17) and rule (W_5) we infer:

$$\frac{\begin{array}{c}\mathscr{P} \succ Comp.signature \leftarrow Comp.petitioner \odot Comp.accountant \odot Comp.superior \\ \odot\, Comp.fdManager \odot Comp.director \\ \mathscr{P} \succ Comp.petitioner \leftarrow \{Alex\} \qquad \mathscr{P} \succ Comp.accountant \leftarrow \{John\} \\ \mathscr{P} \succ Comp.superior \leftarrow \{Jacob\} \qquad \mathscr{P} \succ Comp.fdManager \leftarrow \{Alex\} \\ \mathscr{P} \succ Comp.director \leftarrow \{David\}\end{array}}{\mathscr{P} \succ \mathbf{Comp.signature} \leftarrow \{\mathbf{Alex}, \mathbf{John}, \mathbf{Jacob}, \mathbf{David}\}}$$

showing that here we need four people to complete the set of signatures, i.e. $\{Alex, John, Jacob, David\}$.

Depending on which credentials we have at the moment (because not all the credentials are always available), we can determine the sets of people who can cooperatively sign the document.

4 Credential Extensions

The trust management languages are a powerful tool but they are limited in several ways. As many potential users comment, the languages lack tools to specify a procedure—enforced order of activation for manifold roles. Also, the credentials cannot easily be revoked. The main goal of this section and the paper as a whole is to present two extensions of RT^T language which mitigate these problems, making this language more useful in practice.

4.1 Determination of Order

The first powerful feature which would be useful to model more realistic security policies is the ability to determine the order in which a member of a role or an entity (entities) can appear.

When we want to maintain a procedure, we need to add two new types of credentials at the syntax level. These are:

$A.r \leftarrow B.s\underset{\rightarrow}{\odot}C.t$ — role $A.r$ is satisfied by a union set of one member of role $B.s$ and one member of role $C.t$ in this exact order or by one entity satisfying the intersection role $B.s \cap C.t$

$A.r \leftarrow B.s\underset{\rightarrow}{\otimes}C.t$ — role $A.r$ is satisfied by a set of two different entities: one member of role $B.s$ and one member of role $C.t$ in this order.

In our Example 1 we may want to have such situation:

$F.activeSubject \leftarrow F.students\underset{\rightarrow}{\odot}F.phdStudent$

which means that the order of appearance of people in roles is important. First we need to have two students and just after that one PhD student.

That extension can be very useful in a large variety of situations. For example, in a situation when one person is a member of a few roles, it can be useful to have some restrictions connected with appearing of roles and people in particular roles during the execution context when the credential is used. This is particularly important if the cost of activation of roles differs. For example a high-level manager is usually burdened with many responsibilities and should not be involved unless the other members of a manifold role are available. Alternatively, if an activity involves people from different sites it might make sense to activate all local roles before activating remote ones (for example the central management may reside in a different time zone, introducing long delays).

Example 4 (*Enforced order of signatures*) Suppose that we have a situation in which we need to collect the signatures of people who are necessary to accept some transaction, we can imagine at least a few scenarios for our security policy. Suppose that we need a signature of the petitioner, company accountant, his official superior, the manager of financial department and the director of a company, in such order.

Now we can use the data from *Example* 2 and analyze three different scenarios:

1. The order is strictly obeyed and it is important that the company accountant can only give his signature after having received the signature of a petitioner, and accountant's superior can give his signature after having received the signature of an accountant, even if it is one person. This means that in the first step *Alex* can sign the document as a *petitioner*, in the second step as an *accountant*, after that *David* can give his signature as *Alex′s* official superior. In the next step *Alex* can sign the document as a financial department manager, and finally *David* can sign the document as the director of the company. Table 1 presents the signature order in our first scenario.

   Some situations require keeping a strict order of signatures, but in most implementations it can be a bit inefficient. That is why we propose two other scenarios.
2. We can allow signing the document by one person who plays a few roles at once if the roles appear in credentials successively without any role in between. Table 2 shows how it can look in our second scenario (to make our example easier, we can use just credentials (10), (11), (14), (16), (17)).

Table 1 Signature order in the first scenario

Step	Petitioner	Accountant	Superior	fdManager	Director
1	*Alex*	ϕ	ϕ	ϕ	ϕ
2	*Alex*	*Alex*, *Betty*, *John*	ϕ	ϕ	ϕ
3	*Alex*	*Alex*, *Betty*, *John*	*David*, *Jacob*	ϕ	ϕ
4	*Alex*	*Alex*, *Betty*, *John*	*David*, *Jacob*	*Alex*	ϕ
5	*Alex*	*Alex*, *Betty*, *John*	*David*, *Jacob*	*Alex*	*David*

Table 2 Signature order in the second scenario

Step	Petitioner	Accountant	Superior	fdManager	Director
1	*Alex*	*Alex*	ϕ	ϕ	ϕ
2	*Alex*	*Alex*	*David*	ϕ	ϕ
3	*Alex*	*Alex*	*David*	*Alex*	ϕ
4	*Alex*	*Alex*	*David*	*Alex*	*David*

Table 3 Signature order in the third scenario

Step	Petitioner	Accountant	Superior	fdManager	Director
1	*Alex*	*Alex*	ϕ	*Alex*	ϕ
2	*Alex*	*Alex*	*David*	*Alex*	*David*

As you can see, in such a simple example, we have one step less than in the previous scenario. It shows how such change can be useful in real large systems.

3. In our third scenario we allow that one person, who plays more than one role, can give all the signatures at once. In this case some of the signatures are conditional, that is they are only valid if all previous signatures in the chain are given. This represents a situation where the order of agreements is important, but it is not necessary for one entity to restate its agreement as long as it is initially aware of the full set of roles in which it will be involved in the process. It can be useful in an automatic implementation. Table 3 shows how it looks in our example.

   That situation means that *Alex* accepts his signature as a financial department manager only if *David* signs the document as his official superior and *David* accepts his signature as a *director* if *Alex* signs the document as a financial department manager.

For the last, most relaxed scenario, if we want to mandate that the entity can only appear in particular roles during the execution context exactly when the credential is used, we can put a new type of role denoted by underlined identifiers (e.g. $\underline{r}$, $\underline{s}$, $\underline{t}$). In such situation, when we change the credential:

Table 4 Signature order in the third "underlined" scenario

Step	Petitioner	Accountant	Superior	fdManager	Director
1	*Alex*	*Alex*	ϕ	*Alex*	ϕ
2	*Alex*	*Alex*	*David*	*Alex*	ϕ
3	*Alex*	*Alex*	*David*	*Alex*	*David*

$$\mathit{Comp.signature} \leftarrow \mathit{Comp.petitioner} \overset{\odot}{\rightarrow} \mathit{Comp.accountant} \overset{\odot}{\rightarrow} \mathit{Comp.superior} \overset{\odot}{\rightarrow} \mathit{Comp.fdManager} \overset{\odot}{\rightarrow} \mathit{Comp.director}$$

into:

$$\mathit{Comp.signature} \leftarrow \mathit{Comp.petitioner} \overset{\odot}{\rightarrow} \mathit{Comp.accountant} \overset{\odot}{\rightarrow} \mathit{Comp.superior} \overset{\odot}{\rightarrow} \mathit{Comp.fdManager} \overset{\odot}{\rightarrow} \mathit{Comp.}\underline{\mathit{director}}$$

the final signature can only be given when all others are present.

Table 4 shows how it will look in our third scenario, meaning that *David* has to wait with his signature as a company *director* until the time *Alex* approves his signature as *fdManager*.

All the semantics previously defined for RT^T_+, set-theoretic (which maps roles to a set of entity names), operational semantics (where credentials can be derived from the initial set of credentials using a set of inference rules [5]), and logic-programming (where credentials are translated into a logic program [4]), are still valid, meaning that the proofs of soundness and completeness of those semantics are also valid.

4.2 Time Validity in RT^T

Real security policies usually involve some kind of time restrictions. Allowing the credentials to have limited time validity can make the RT-family languages more useful in practice. While time restrictions for RT_0 have been proposed before [7], they were not so far generalized to the RT^T language. Most permissions are in fact given for fixed periods of time, permanent permissions are less common. Time dependent credentials take the form: c **in** v, meaning "the credential c is available during the time v". Finite sets of time dependent credentials are denoted by $\mathscr{CP}$ and the new language is called RT^T_+. c is used to denote "c **in** $(-\infty, +\infty)$" to make notation lighter.

Most trust management languages are monotonic: adding a new assertion to a query can never result in canceling an action, which was accepted before [2]. Therefore, each policy statement or credential added to the system may only increase the capabilities and privileges granted to others, making revocation of rights impossible. In centralized solutions nonmonotonicity only makes caching of inferred credentials more difficult, but in a distributed environment it is crucial—connectivity problems or failures of some nodes might block the propagation of the assertion revoking a credential, potentially leading to wrong decisions, granting access on the basis of revoked credentials. Introduction of time constraints does not invalidate the monotonicity of the system, but achieves some of the utility of negation. For example, while the credentials still cannot be revoked, they can be issued automatically and periodically for short periods. Cancelling this automatic reissuing for a certain principal does not in fact invalidate any existing credentials but has the same effect, although with a delay (no longer than the length of the validity period of the periodic credentials).

Time validity can be denoted as follows:
$[\tau_1, \tau_2]; [\tau_1, \tau_2); (\tau_1, \tau_2]; (\tau_1, \tau_2); (-\infty, \tau]; (-\infty, \tau);$
$[\tau, +\infty); (\tau, +\infty); (-\infty,\ +\infty); v_1 \cup v_2; v_1 \cap v_2; v_1 \backslash v_2$
and v_1, v_2 of any form in this list, with τ ranging over time constants.

Example 5 (*Time validity for Example* 1) In our scenario, it is quite natural to assume that *Alicia*, *Carol*, *Marry* and *Greg* are students only for some fixed period of time. The same with *Greg* and *Marc* as PhD students. Thus, credentials (3)–(8) should be generalized to:

$$F.student \leftarrow \{Alicia\}\ \mathbf{in}\ v_1 \tag{18}$$

$$F.student \leftarrow \{Carol\}\ \mathbf{in}\ v_2 \tag{19}$$

$$F.student \leftarrow \{Marry\}\ \mathbf{in}\ v_3 \tag{20}$$

$$F.student \leftarrow \{Greg\}\ \mathbf{in}\ v_4 \tag{21}$$

$$F.phdStudent \leftarrow \{Greg\}\ \mathbf{in}\ v_5 \tag{22}$$

$$F.phdStudent \leftarrow \{Marc\}\ \mathbf{in}\ v_6 \tag{23}$$

stating that (3)–(8) are only available during v_1, v_2, v_3, v_4, v_5, and during v_6, respectively. On the other hand, credentials (1) and (2) are always valid, as they express some time-independent facts. Now, by using (1), (2) and (18)–(23), we want to be able to derive that for example the set $\{Alicia, Carol, Greg\}$ can cooperatively activate the subject during all of the period: $v_1 \cap v_2 \cap v_5$ or $\{Carol, Greg\}$ during the time $v_2 \cap v_4 \cap v_5$ or $\{Alicia, Marry, Marc\}$ during the time intersection $v_1 \cap v_3 \cap v_6$. Another set of people can cooperatively activate the subject (depending of the time).

Example 6 (*Time validity for Example* 2) In our signature scenario, it is quite natural to assume that *Alex* is a petitioner only for a fixed period of time. The same with *Alex*, *Betty* and *John* as a company accountants, also *Wiliam* and *Jacob* as a superior, and *Alex* as a financial department manager, as well as *David* as a director. Thus, credentials (10)–(17) should be generalized to:

$$Comp.petitioner \leftarrow \{Alex\} \textbf{ in } v_1 \tag{24}$$

$$Comp.accountant \leftarrow \{Alex\} \textbf{ in } v_2 \tag{25}$$

$$Comp.accountant \leftarrow \{Betty\} \textbf{ in } v_3 \tag{26}$$

$$Comp.accountant \leftarrow \{John\} \textbf{ in } v_4 \tag{27}$$

$$Comp.superior \leftarrow \{David\} \textbf{ in } v_5 \tag{28}$$

$$Comp.superior \leftarrow \{Jacob\} \textbf{ in } v_6 \tag{29}$$

$$Comp.fdManager \leftarrow \{Alex\} \textbf{ in } v_7 \tag{30}$$

$$Comp.director \leftarrow \{David\} \textbf{ in } v_8 \tag{31}$$

stating that (10)–(17) are only available during v_1, v_2, v_3, v_4, v_5, v_6, v_7, and during v_8, respectively. On the other hand, credential (9) can be always valid, as it expresses some time-independent fact. Now, by using (9) and (24)–(31), we want to be able to derive that for example the set $\{Alex, David\}$ can cooperatively sign the document during all of the period: $v_1 \cap v_2 \cap v_5 \cap v_7 \cap v_8$, where *Alex* plays the role of *petitioner*, *company accountant* and *financial department manager* and *David* acts as a *superior* and *director of the company*. But during the time $v_1 \cap v_3 \cap v_6 \cap v_7 \cap v_8$ it has to be the set consisting of $\{Alex, Betty, Jacob, David\}$.

While in our examples the policy defining credentials were assummed to be permanently valid, this is not required. Some policies are naturally time-limited (eg. seasonal sales).

5 Conclusions

In the paper we model the use of trust management systems in decentralized and distributed environments. The modelling framework is RT^T—one of the languages from a family of Role-based Trust management languages. The core, innovative part of the paper is the introduction of time validity constraints and especially maintaining the procedure—modifications aimed at making the RT^T language more realistic. The utility of the proposed extensions is most visible in large-scale distributed systems, where users have only partial view of their execution context. The language achieves the ability to define non-permanent credentials, which are necessary in real applications, but the monotonicity of the system is preserved, meaning that temporary unavailability of some constraints never leads to granting illegal access.

The proposed language has recently been implemented. The experiences with potential users show that the newly introduced extensions add significant value, as most comments regarding its usability in practice are addressed. Therefore the model itself is now considered satisfactory and future work is focused on implementation details.

References

1. Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized trustmanagement. In: Proceedings of the 17th IEEE Symposium on Security andPrivacy, Oakland CA, 1996, pp. 164–173. http://dx.doi.org/10.1109/SECPRI.1996.502679
2. Czenko, M.R.: Nonmonotonic trust management for P2P applications. In: Proceedings of the 1st International Workshop Security and Trust Management STM, et al.: Milan, Italy, vol. 2005 (2005). http://dx.doi.org/10.1016/j.entcs.2005.09.037
3. Felkner, A., Sacha, K.: Deriving RT^T credentials for role-based trust management, e-Informatica. Softw. Eng. J. **4**(1), 9–19 (2010)
4. Felkner, A., Kozakiewicz, A.: RT^T_+-time validity constraints in RT^T language. J. Telecommun. Inf. Technol. **2**, 74–82 (2012)
5. Felkner, A., Kozakiewicz, A.: Time validity in role-based trust management inference system. secure and trust computing, data management, and applications communications. Comput. Inf. Sci. **187**, 7–15 (2011). http://dx.doi.org/10.1007/978-3-642-22365-5_2
6. Felkner, A., Kozakiewicz, A.: More practical application of trust management credentials. In: Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS), 2015, pp. 1137–1146
7. Gorla, D., Hennessy, M., Sassone, V.: Inferring dynamic credentials for role-based trust management. In: Proceedings of the 8th Conference on Principles and Practice of Declarative Programming, ACM, 2006, pp. 213–224. http://dx.doi.org/10.1145/1140335.1140361
8. Li, N., Winsborough, W., Mitchell, J.: Distributed credential chain discovery in trust management. J. Comput. Secur. **1**, 35–86 (2003)

Performance and Energy Consumption Analysis of AES in Wireless Sensor Networks

Dan Dragomir and Cristina Panait

Abstract With WSN deployments increasing in popularity, securing those deployments becomes a necessity. This can be achieved by encrypting inter-node communications and/or using message authentication codes. AES is a well studied symmetric cipher, with no known practical vulnerabilities, that can be used to solve both problems. We provide an optimized implementation of AES, with five modes of operation for encryption (ECB, CBC, CFB, CTR and GCM) and two modes for authentication (CBC-MAC and GCM-MAC), that use the hardware accelerator available on the ATmega128RFA1 microcontroller, and compare it with the best known software implementation. We show that our hardware AES implementation is both faster and more energy efficient than a software implementation. This is not the case for previous sensor nodes and implementations, which show an improved execution speed, but with a higher energy consumption. We also show that our implementation of CTR is faster and more energy efficient than the unsecure fully hardware-supported ECB mode.

1 Introduction

Wireless sensor networks (WSNs) and their applications present an increased risk to a series of attacks, which can affect a network's operation. As a solution to these problems we present a performance and energy consumption analysis of AES-128 on WSN hardware. AES is a well studied block cipher with no known practical vulnerabilities, has a speed comparable with other symmetrical encryption algorithms and is supported on multiple WSN platforms through a hardware acceleration module. This work is an extended version of [12] and improves the previous study in two key areas. Firstly, it further optimizes the hardware-assisted implementation, leading

D. Dragomir (✉) · C. Panait
Faculty of Automatic Control and Computers—University POLITEHNICA of Bucharest,
Splaiul Independentei 313, 060042 Bucharest, Romania
e-mail: dan.dragomir@cs.pub.ro

C. Panait
e-mail: cristina.panait19@gmail.com

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent
Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_11

to better performance and lower energy consumption and secondly, it adds two new algorithms to the analysis, that handle message authentication and verification.

As a general definition, a wireless sensor network is composed of a set of nodes which communicate through a wireless medium in order to perform certain tasks. A couple of examples where WSNs can be deployed, as stated in [6], are: fire extension detection, earthquake detection, environment surveillance for pollution tracking, intelligent building management, access restriction, detection of free spaces in parking lots and so on. WSNs bring a number of advantages in these situations, like enhanced flexibility and mobility, mainly because nodes are generally powered from an on-board battery and are thus self sufficient. This, however, is also their biggest weakness. The lifetime expectancy of a node depends on its usage. The constraints mainly come from the limited energy source, as data processing and transmission can be energy intensive.

The particular characteristics of these types of networks make the direct implementation of conventional security mechanisms difficult. The imposed limitations on minimizing data processing and storage space, and reducing bandwidth need to be addressed. The major constraints for WSNs, as presented in [2, 14, 17], are: energy consumption (which can lead to premature exhaustion of the energy source and to the denial of service), memory limitations (flash, where the application source code is stored, and RAM, where sensed data and intermediary computing results are stored), unreliable communication (the routing protocols used, collisions), latency (which can lead to synchronization issues and algorithms that cannot act correctly) and unattended nodes (an attacker could have physical access to the nodes).

In Sect. 2 we discuss some of the related work. Sections 3 and 4 present the algorithm design and modes of operation and the implementation with two methods, software and hardware. Then, in Sect. 5, we make a comparative analysis of the solutions, based on execution time and energy consumption, and select the encryption and authentication methods suitable for ATmega128RFA1-based platforms, taking also into consideration the provided security. Finally, we present the conclusions of our work.

2 Related Work

The problem of measuring the cost of encryption on wireless sensor node hardware has been addressed previously. In [9] Lee et al. analyze a range of symmetric-key algorithms and message authentication algorithms in the context of WSNs. They use the MicaZ and TelosB sensor nodes and measure the execution time and energy consumption of different algorithms. For AES they provide measurements for a hardware assisted implementation and conclude that it is the cheapest when either time or energy is considered. They do not however study this implementation on different plaintext lengths and instead rely on datasheets to extend to lengths longer than one block. However, this conclusion is not backed by Zhang et al. [19] which compares different AES implementations on the MicaZ nodes. They conclude that hardware assisted encryption is faster, but also consumes more energy due to the external chip which handles the computation in hardware.

Compared to their work, we study only AES-128 which is a well known cipher also adopted by the National Institute of Standards and Technology (NIST) and which has been proposed as a viable alternative [8] to other less studied ciphers in WSN applications. This choice is also supported by the fact that multiple 802.15.4 transceivers offer a hardware accelerator for AES operations. We study the newer Sparrow v3.2 sensor nodes based on the ATmega128RFA1, which integrates the microcontroller with the radio transceiver and hardware encryption module, and show that AES-128 can be efficiently implemented reducing both execution time and energy consumption. We also provide hybrid implementations for modes of operation that are not natively supported by the hardware and show that they can still be efficiently implemented with the available primitives.

In [8] Law et al. conduct a thorough survey of the costs of different block ciphers, when implemented on sensor node hardware. They conclude that Rijndael (AES) is the second most efficient cipher, being surpassed only by Skipjack. However, their analysis is based on older hardware and does not consider any hardware accelerated implementations.

In [4] de Meulenaer et al. study the problem of key exchange and measure the cost of two key agreement protocols: Kerberos and Elliptic Curve Diffie-Hellman. They measure the energy consumption of the two protocols on MicaZ and TelosB sensor nodes and conclude that the listening mode is the principal factor in the energy efficiency of key exchange protocols, with Kerberos being the more efficient protocol. Compared to their work, we concentrate on encryption and authentication algorithms, and more specifically on AES, with key distribution left for future work.

In a previous work [12] we reported the performance and energy consumption of our implementation for 4 modes of operation that use the AES block cipher: ECB, CBC, CFB and CTR. The current work extends that analysis by also studying the cost of data authentication using CBC-MAC and adding a new mode of operation, GCM, that seamlessly supports both encryption and authentication using the same secret key. We also improve on the performance and energy consumption of our previous implementation for some of the modes, by making use of pipelining, in the hardware assisted case.

3 Design

AES is a block cipher encryption algorithm that uses symmetrical keys for encrypting a block of plaintext and decrypting a block of ciphertext [3]. The algorithm uses a series of rounds consisting of one or more of the following operations: byte-level substitution, permutation, arithmetical operations on a finite field and XOR-ing with a given or calculated key [15]. As a general rule, the operations are handled bytewise.

AES receives as input a plaintext of 16 bytes and the encryption key, which has a variable dimension of 16, 24 or 32 bytes. The input text is processed into the output text (ciphertext) by using the given key and applying a number of transformations. Encryption and decryption are similar, except for the fact that decryption needs an extra step—it first runs a full encryption in order to obtain the modified key needed for decrypting data.

In [13], Schneier divides symmetrical encryption algorithms in two basic categories: block ciphers and stream ciphers. A block cipher encrypts a block of plaintext producing a block of encrypted data, whilst a stream cipher can encrypt plaintexts of varying sizes. This makes block ciphers prone to security issues if used to encrypt plaintexts (in a naïve way) longer than the block size, mainly because patterns in the plaintext can appear in the ciphertext.

A more secure way to encrypt data with a block cipher can be achieved by combining the resulting ciphertext blocks using a few basic operations, resulting in what is called a *mode of operation* for that block cipher. It is worth mentioning that the combining operations are not directly securing the data. This is the responsibility of the block cipher. The operations however are used to maintain the security of the cipher when it is operated on plaintexts longer than the block size.

Another use case for a block cipher is to compute a Message Authentication Code (MAC) for a piece of data, in order to provide data authentication. As for the encryption case, a naïve implementation would not provide the necessary guarantees for the generated MAC. Extra precaution must be taken when both encryption and authentication are required as not all combinations of encryption and authentication methods provide the desired properties. For a thorough treatment of the security properties of the different options we refer to Katz and Lindell [7].

3.1 Electronic Code Book (ECB)

The ECB mode of operation receives blocks of plaintext, respectively ciphertext, and a key and produces corresponding blocks of ciphertext, respectively plaintext. One property of this mode of operation is that two blocks of plaintext, encrypted with the same key, will result in two identical blocks of ciphertext. ECB is the most simple mode of operation. However, one major drawback is that it does not hide data patterns, meaning that identical ciphertext blocks imply the existence of identical plaintext blocks.

3.2 Cipher Block Chaining (CBC)

The CBC mode of operation takes as input parameters the plaintext, respectively the ciphertext, the key and an initialization vector (IV). One property of CBC is that two encrypted blocks are identical only if their respective plaintexts have been encrypted using the same key and the same IV. Unlike ECB, CBC has link dependencies, as its basic chaining mechanism makes the ciphertext blocks dependent on previously encrypted data. This, coupled with a randomly chosen IV, ensures that identical plaintext blocks will be encrypted to different ciphertext blocks.

With slight modifications CBC can be transformed to provide authentication instead of encryption, resulting in CBC-MAC. For this, the initialization vector must be fixed to a constant value (usually 0) and only the last encrypted block must be kept, which is also the authentication tag. This scheme will provide a secure MAC, but only for messages of fixed length (agreed ahead of time by the communicating parties). A further modification of prepending the length of the message as the data in the first block of the encryption, can make this mode secure for varying length messages. A disadvantage of CBC-MAC is that when both encryption and authentication are required, separate keys must be used for the two steps in order to maintain security.

3.3 Cipher Feedback (CFB)

The CFB mode of operation is very similar to CBC regarding its input parameters and the operations it performs. The main difference between them lies in the fact that CBC works as a block cipher, while CFB can be used as a stream cipher. Unlike CBC, CFB can encrypt variable-length blocks (which are not restricted to 16 bytes). The properties of this mode of operation are similar with the ones of CBC. One key difference between the two can be observed at the implementation level: CFB uses only the encryption primitive of the underlying block cipher, both for encrypting and for decrypting data.

3.4 Counter (CTR)

The CTR mode of operation [10] also produces a stream cipher. The IV used in CBC and CFB is now associated with the starting value of a counter, which is incremented and used to encrypt each block in turn. In this mode, the output from an earlier block is not used for computing the current block, as the previous two modes of operation. For the described system to work, a generator is needed on each side of the communication. The generators have to remain synchronized in order to produce the same stream of counter values on both sides. A disadvantage of this mode of operation is the possible desynchronization of the communicating entities. This results in the incorrect decryption of all subsequently received data.

A closely related mode of operation is Counter with CBC-MAC (CCM) [18], which provides both encryption and authentication of the plaintext data. In this mode, CBC-MAC is initially run over the message in order to obtain a authentication tag, and then, CTR mode is run on both the plaintext data and the authentication tag to obtain the ciphertext, which now provides both encryption and authentication. A slight variation of CCM, called CCM*, is part of the IEEE 802.15.4 standard [5] for wireless personal area networks.

3.5 Galois/Counter Mode (GCM)

The GCM mode of operation [11] combines the Counter mode with operations on a Galois field in order to produce another mode of operation which provides both encryption and authentication of a piece of data using a single secret key. The key operation is a multiplication in $GF(2^{128})$, defined by the polynomial $x^{128} + x^7 + x^2 + x + 1$, which is used to define a hashing function that generates the authentication tag. The algorithm supports additional authenticated data (AAD), which is data protected against tampering by the authentication tag, but left unencrypted. This additional data is useful in networking protocols, where source and destination information must be left in cleartext for the purpose of routing. The algorithm can be easily converted to an authentication only mode of operation by providing only AAD and no encryption payload.

4 Implementation

A practical example would be a wireless sensor network, which transmits data gathered from three types of sensors: temperature, humidity and luminosity. Because of privacy and integrity concerns all data must be encrypted during transmission and routing information must be authenticated. The working platform for this scenario is based on the Sparrow v3.2 node [16]. Its technical specifications are:

- CPU: ATmega128RFA1, 16MHz
- Memory: 128KB flash, 16KB RAM
- Bandwidth: up to 2Mbps
- Programming: C/C++

The ATmega128RFA1 microcontroller is actually a SoC (System on Chip) which incorporates a radio transceiver compatible with the IEEE 802.15.4 standard [1]. It offers, among other things, a relatively low energy consumption (mostly in sleep states), a FIFO buffer of 128 bytes for receiving and transmitting data, a partial hardware implementation of the MAC layer and support for AES-128.

This microcontroller facilitates secured data transmissions by incorporating a hardware acceleration module which implements the AES algorithm. The module is capable of both encrypting and decrypting data, with most of the functionality implemented directly in hardware. It is compatible with the AES-128 standard (the key is 128 bits long) and supports encryption and decryption for ECB mode, but only encryption for CBC mode. The input to these operations consists of the plaintext/ciphertext block and the encryption key. Note that for decryption, the extra round needed by AES to compute the decryption key is performed automatically. Other modes of operation are not supported by the hardware.

As we already stated in the previous sections, energy consumption is the main issue and challenge for WSNs. In order to obtain the best approach for ensuring confidentiality and integrity with minimal energy consumption, we implemented and compared AES-128, coupled with the ECB, CBC, CFB, CTR and GCM modes of operation. All five modes have both a hardware and a pure software implementation. Since only ECB has a full hardware implementation, for the other modes we used a hybrid approach, combining the hardware part from ECB with software implementations for the remaining operations. We also refer to these hybrids as hardware implementations. Were possible we pipelined the algorithm's execution so that the extra software steps not implemented in hardware were overlapped with the encryption of the next data block, thus achieving better performance than the serial solution employed in [12]. For the pure software implementation we used an optimized version of AES, called TableLookupAES [19].

5 Evaluation

5.1 Experimental Setup

To measure the energy consumption of our implementation, we perform two kinds of measurements: the time required (t) and the current drawn by the node (I) while encrypting/decrypting. Using the formula $E = P \cdot t$, where $P = U \cdot I$ is the power required by the node, we can compute the energy consumed by the algorithm, be it implemented in software, in hardware or using a hybrid approach. We ensure a constant voltage U using a voltage regulator, as explained in the next subsection.

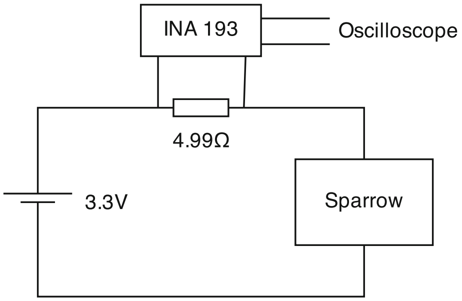
In certain applications, the latency of encrypting/decrypting a given payload might be more important than the energy consumed. For this reason, this section also presents the timing results of the different solutions, independent of the energy measurements. As we later show, the current drawn by the node using both software and hardware security approaches is practically the same. Thus, the time taken is a sufficient metric for relative comparisons between the different solutions.

5.1.1 Current Measurement

For the purpose of measuring the energy consumption of the Sparrow sensor node during our experiments, we built a current sensing circuit based on the INA 193 current shunt monitor.

Figure 1 presents the circuit we designed. Power is provided by a $3.3V$ voltage regulator, which ensures a constant voltage regardless of the current drawn by the circuit. A shunt resistor connected in series with the Sparrow node acts as a cur-

Fig. 1 Current measurement setup



rent sensor. The voltage drop on the resistor is directly proportional with the current drawn by the circuit. This has two implications. On the one hand, the chosen resistor value must be small enough not to disturb the rest of the circuit (e.g. by incurring a big voltage drop). On the other hand, the same value has to be big enough so that the expected currents register a voltage drop that can be sensed with enough precision. In order to improve the measurement precision and sensitivity, without the drawbacks of a big resistor value, we employ a INA 193 current shunt monitor, which provides a constant gain of 20 V/V on the input voltage drop, and a 4.99 Ω precision resistor with a tolerance of 0.01 %. The output of the current sensing circuit is connected to a Metrix OX 5042 oscilloscope which we used to monitor the current drawn by the node during the different encryption/decryption operations. Determining the current is as simple as dividing the voltage shown on the oscilloscope by the current shunt monitor gain (20 V/V) and the shunt resistor value (4.99 Ω).

5.1.2 Time Measurement

Using the oscilloscope, we also measure the time required for each encryption/ decryption operation. The oscilloscope has a function that accurately measures pulse duration. We create a pulse lasting for the duration of the operation by setting a GPIO pin before the start of the operation and clearing it after it ends. Using this method, we can measure the duration of an operation with minimal overhead: 1 bit set instruction and 1 bit clear instruction, each taking 2 cycles.

Although the proposed measurement scheme is precise, it has the disadvantage of requiring manual intervention. The available oscilloscope cannot be interfaced with a PC, so a measurement point is obtained by uploading a program which encrypts a hardcoded message length in a loop, reading the information from the oscilloscope and repeating the process for all message lengths.

In order to automate the time measurements, we resorted to a software implementation running along side the encryption/decryption operation, that measures the time required. To keep overhead to a minimum, our solution employs the hardware timer module available on the ATmega128RFA1 to count the number of cycles taken by the operation. Each operation is measured by sampling a counter before and after the operation and taking the difference of the two values. The count is then converted to a time value given that the microcontroller operates at $16MHz$.

This time measurement solution allowed us to automate the whole process of evaluating the algorithms for different message sizes. A small overhead can be observed between the software based time measurement and the oscilloscope based one, but the relative difference between the algorithms is unaffected. If absolute numbers are required, the software-based measurements can be corrected by noticing that the overhead increases linearly with the message size when compared with the oscilloscope measurements.

5.2 Results

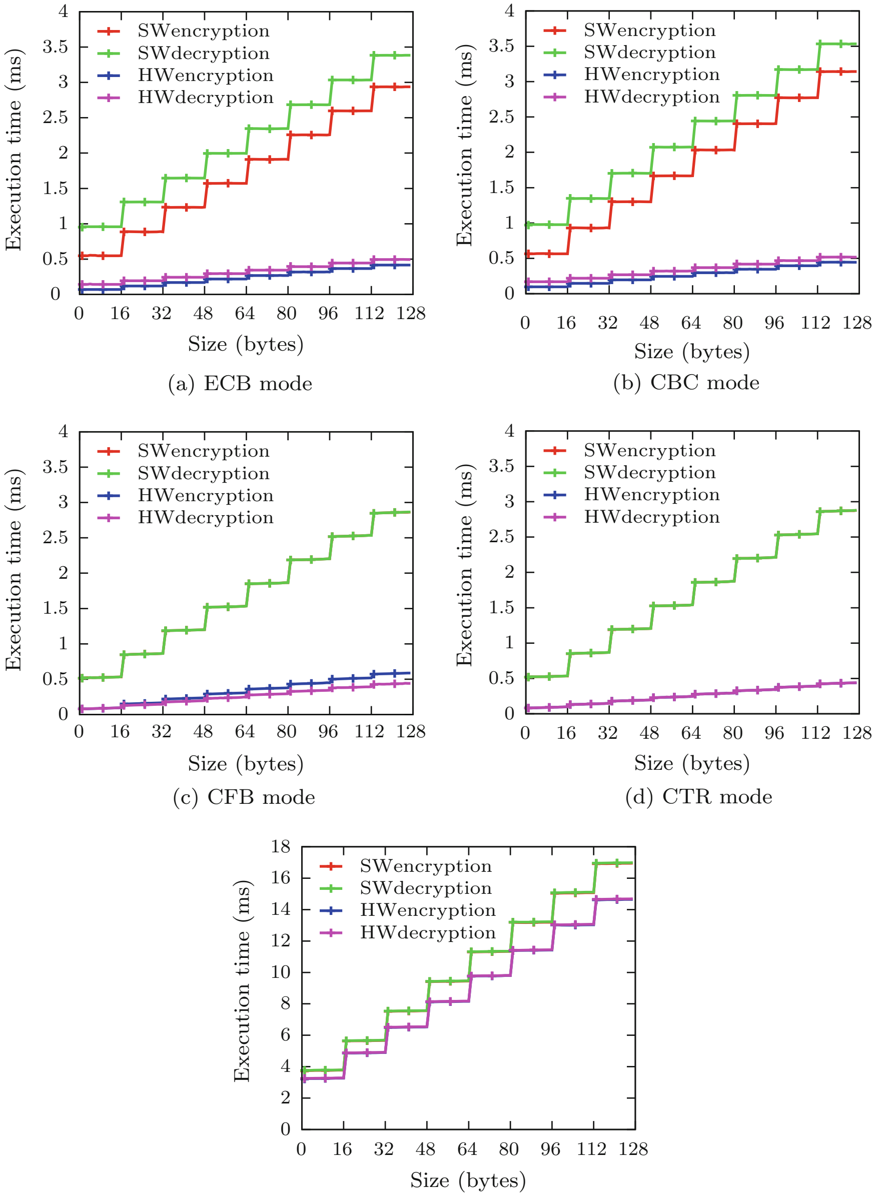
We conducted multiple experiments, to evaluate both the time taken and the energy consumed by AES encryption/decryption and authentication/verification operations. We measured our hardware assisted implementation against the pure software implementation based on look-up tables.

5.2.1 Time Experiments

We started of with measuring the difference between the optimized software implementation and our hardware assisted implementation for each of the five studied modes of operation. For each type of implementation and operation mode, we measured the time taken by an encryption operation and a decryption operation on varying message lengths. We used message lengths from 1 byte to 127 bytes, which is the maximum packet size allowed by the transceiver and the 802.15.4 standard.

As can be seen in Fig. 2, the hardware assisted implementation easily outperforms the optimized software implementation for 4 of the 5 modes. For GCM the difference is not as pronounced as the other modes, but the hardware assisted implementation is still faster for all message lengths. The reason the difference is less pronounced is that, compared to the other four modes, GCM takes a longer time to compute. That time is used by the software implementation of the $GF(2^{128})$ multiply operation, which cannot be accelerated in hardware on the ATmega128RFA1. The time saved for doing the block cipher in hardware is small compared with the multiply operation, which leads to a less pronounced speed-up. The staircase shape of the graph is easily explained by the requirement of every block cipher, including AES, to operate on multiples of the block size. Plaintext sizes that are not a multiple of the block size need to be padded in most cases, thus still incurring the cost of an entire block.

The difference in performance varies between ~7.0× for the ECB mode, which is fully supported in hardware, down to ~1.15× for the GCM mode, which is only partially supported in hardware through the AES single block encryption primitive. The MAC version of GCM has an even lower speed-up, of ~1.01×, which is explained by the fact that in a pure authentication mode, GCM only performs one block encryp-



(a) ECB mode

(b) CBC mode

(c) CFB mode

(d) CTR mode

(e) GCM mode

Fig. 2 Comparison between software and hardware implementations of AES encryption modes

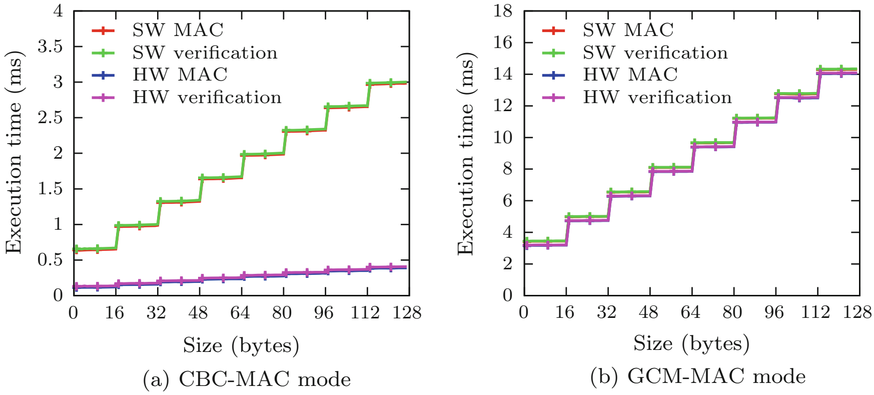
Table 1 Execution speed-up hardware versus software

	Encryption	Decryption	Authentication	Verification
ECB	7.01×–7.94×	6.44×–6.84×	–	–
CBC	5.78×–7.07×	5.75×–6.82×	5.40×–7.72×	4.82×–7.46×
CFB	4.86×–6.51×	5.55×–6.69×	–	–
CTR	5.45×–6.75×	5.47×–6.75×	–	–
GCM	1.15×	1.15×	1.01×–1.08×	1.01×–1.08×

tion regardless of the message length. The difference in performance between the optimized software implementations and our hardware assisted implementations is further summarized in Table 1.

For the ECB and CBC modes we can observe (Fig. 2a, b) the extra preparation step needed by the single block decryption primitive, which makes decryption slightly more time consuming than encryption. No difference can be observed (Fig. 2c, d) between encryption and decryption for the CFB and CTR modes in the software implementation, because they use the same encrypt primitive of AES for both encryption and decryption, albeit with some extra processing. In the hardware assisted case CTR maintains equal performance between encryption and decryption, however CFB encryption gets progressively slower as the message length increases. This is caused by the extra software processing step, which cannot be hidden by pipelining, as is done for CTR, because of data dependencies in CFB encryption. Again, no performance difference can be observed (Fig. 2e) in the GCM case between encryption and decryption. What is relevant for GCM though, is the time required for either the software or hardware implementations which is almost five times longer than even the slowest of the other modes.

The behavior of the authentication modes is shown in Fig. 3. Both modes have an almost equal performance between generating the MAC and verifying it, in both



(a) CBC-MAC mode

(b) GCM-MAC mode

Fig. 3 Comparison between software and hardware implementations of AES based MACs

the software and the hardware implementations. This was to be expected as the verification step implies recomputing the authentication tag and comparing it with the received one. What is notable is that the comparing step adds minimal overhead compared with the tag calculation. Like in the encryption case, the hardware implementation is again much faster for CBC (Fig. 3a), while for GCM (Fig. 3b) the speed-up gained from the hardware assist is dwarfed by the time required for the $GF(2^{128})$ multiplication.

Figure 4 compares the hardware assisted implementations of 4 of the modes (ECB, CBC, CFB and CTR) against each other, during encryption and decryption. GCM was left out of this comparison as its hardware assisted performance was poor compared with the other modes. For encryption, ECB has the lowest runtime for all sizes, which was to be expected, as it does no extra operations on the output of the encrypt primitive to mask patterns in the plaintext. CTR is slightly worse, but not by much. The cost of the extra XOR operation required by this mode is mostly hidden by our pipelined implementation. This is not the case in the non-pipelined implementation measured in [12]. CBC is slightly worse, as its extra XOR operation cannot be

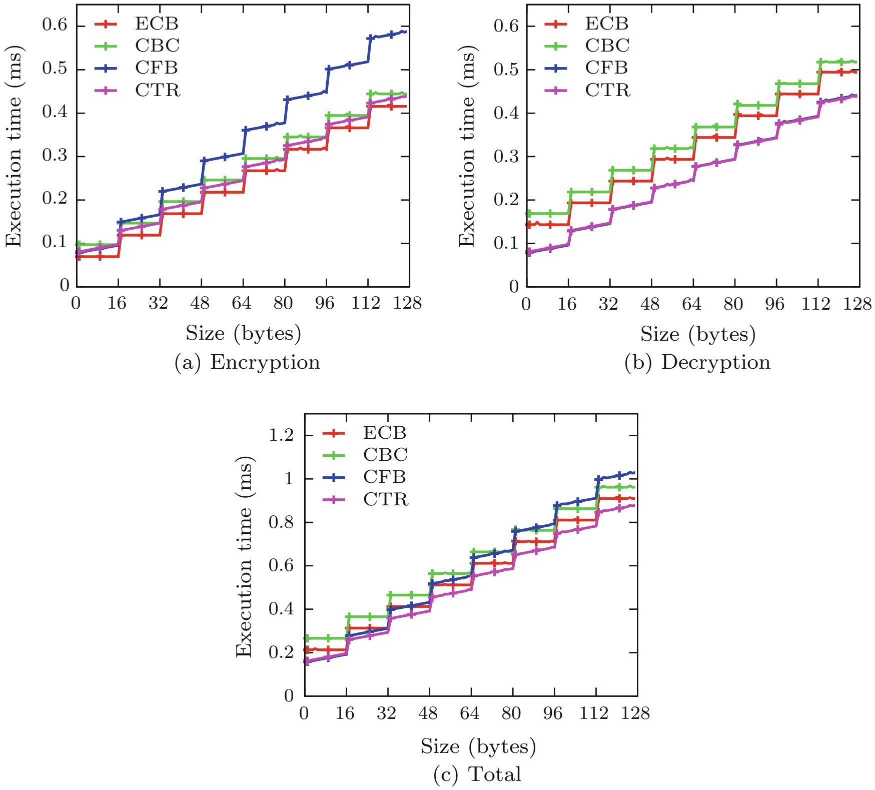


Fig. 4 Comparison between modes of operation with hardware acceleration

pipelined. The difference remains constant though, as the message length increases because only for the first block the XOR operation is emulated in software. For all the other blocks it is implemented by the hardware accelerator. Finally, CFB has the worst performance of the 4 modes mostly attributed to its extra XOR operation that cannot be pipelined like in the CTR case, because of data dependencies in the the algorithm. The cost of the extra operations increases as the encrypted message gets longer.

For decryption, CFB and CTR have a considerable advantage over ECB and CBC, as they only use the encrypt primitive, which has a smaller setup time than the decrypt primitive. Also, in the decryption case, the cost of the emulated XOR operation required by CBC, CFB and CTR is hidden by our pipelined implementation, thus giving a constant difference over all message lengths.

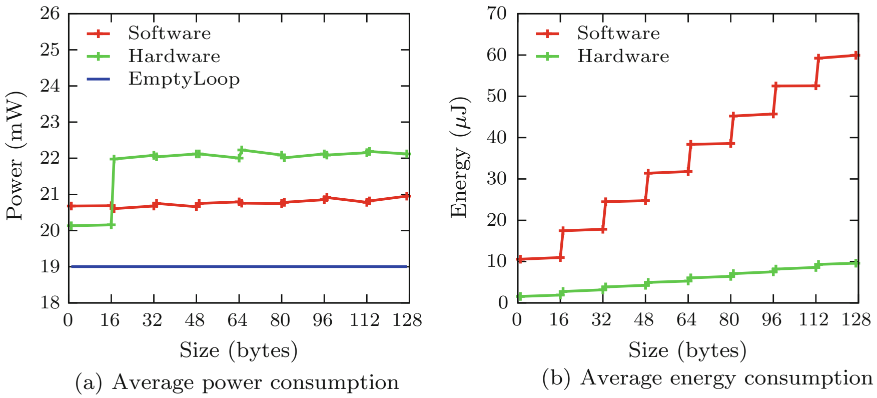
Another performance characteristic shown by both plots is the streaming nature of CFB and CTR. They can be optimized to reduce the performance cost when the message only covers part of a block. This can be seen as the slanting portions of the lines for CFB and CTR in both encryption and decryption. No such behavior is visible for ECB and CBC, which must fully process a whole block, even if only one byte of the message is contained in the block.

If we look at the cumulated time of both encryption and decryption (Fig. 4c), CTR holds a consistent advantage over all the other modes. CFB also holds an advantage over the unsecure ECB up to messages of 40 bytes. Compared with CBC, CFB is faster up to messages of 80 bytes. Thus, even if most WSN hardware offers accelerated support for AES-CBC, CFB and especially CTR can be better alternatives even if they are not completely accelerated in hardware.

5.2.2 Energy Experiments

For energy consumption we concentrated our efforts on determining the cost of using AES in CTR mode. We chose this mode based on the fact that the timing measurements showed it to be the best encryption/decryption mode for all message sizes. We only performed measurements for message encryption, as decryption is identical in terms of the code which is ran. We measured the cost of doing the encryption in software as well as the cost of using our hardware accelerated implementation. For completeness, we also measured the cost of an empty processing loop to compare against the two encryption implementations.

In our experiments, we used the measurement circuit described in Sect. 5.1.1 to measure the average current drawn during encryption, as well as the voltage and duration of the operation, as reported by the oscilloscope. As with the timing measurements, we performed the experiment for different message sizes, from 1 byte to 127 bytes. The oscilloscope was configured to report the mean over 16 samples in order to obtain the average energy consumption of the device. An instantaneous energy consumption is hard to obtain and is irrelevant when considering the long time operation of the node.



(a) Average power consumption

(b) Average energy consumption

Fig. 5 Power and energy consumption of AES encryption in CTR mode

Using the raw current and voltage measurements, we plot the average power drawn with respect to the encryption size. As can be seen in Fig. 5a, for message sizes larger than 16 bytes the hardware implementation consistently draws more power than the software solution. This is in contrast with our previous measurements [12] which showed mostly equal amounts of power drawn by the two implementations. The difference though is that this new hardware implementation uses pipelining to overlap the hardware accelerated block encryption with the emulated XOR operation, thus using more of the transistors on the chip at a given time. This can also be seen in the first part of the graph, for message lengths less than 16 bytes, where a single block exists and pipelining is not possible. In this case the power drawn by the two solutions is more or less equal.

If we plot the average energy consumed by the encryption operation (Fig. 5b), we see a mostly linear increase in energy consumption with increasing plaintext size. The higher power needed by the pipelined hardware assisted implementation is more than offset by the lower running time, thus leading to a lower overall energy consumption. We believe the same conclusion holds for other operation modes, like CBC and CFB, as they mostly use the same operations as CTR, but in a slightly different order.

6 Conclusion

In this paper an updated evaluation of the cost of adding AES-128 encryption to WSN communications has been presented. We expand the work in [12] with more modes of operation and an improved pipelined implementation for some of the modes. Both the time penalty as well as the more important (from the point of view of a WSN) energy penalty have been analyzed for: ECB, CBC, CFB, CTR and GCM and for two implementations: a pure software implementation, based on the optimized table lookup AES and the hardware accelerated implementation, that uses the AES hardware module of the ATmega128RFA1 microcontroller.

We showed how the AES hardware module in the ATmega128RFA1 microcontroller can be used to implement other modes of operation than the ones supported natively. Our solution uses a hybrid approach that runs some operations in hardware and emulates the missing ones in software. Where possible we pipeline the algorithm's execution to completely hide the cost of the emulated operation in terms of processing time. Using this approach, we implemented CBC decryption, as well as three full modes of operation for AES, CFB, CTR and GCM which do not have direct hardware support.

We presented a methodology of accurately measuring the power consumption using low cost components and a way of determining the encryption/decryption duration using only the wireless node itself. We compared the different modes of operation and concluded that a pipelined and hardware assisted implementation of CTR can be faster than even the unsecure and completely hardware accelerated ECB mode. CFB is also a better overall alternative to CBC for message sizes smaller than 80 bytes. This is true even though the hardware accelerator has native support for the CBC mode and it relates to the way decryption works for CBC. In contrast, GCM has performed very poorly in the hardware assisted case, even though a small speedup was obtained when compared to a pure software implementation. If authenticated encryption is required, using CTR for encryption followed by CBC-MAC for authentication represents the best combination in terms of both time performance and energy consumption.

We also built on the work of Zhan [19] and showed that the newer ATmega128 RFA1 microcontroller with an integrated transceiver, used in the Sparrow v3.2 node, can reduce both the duration and the energy consumption of AES operations. This is in contrast to work done on previous sensor nodes, that used a separated microcontroller and transceiver and which had a higher energy cost when running the encryption in hardware as opposed to using a pure software implementation.

References

1. Atmel: 8-bit AVR Microcontroller with Low Power 2.4 GHz Transceiver for ZigBee and IEEE 802.15.4
2. Carman, D.W., Kruus, P.S., Matt, B.J.: Constraints and approaches for distributed sensor network security (final). Technical Report 1, NAI Labs, Cryptographic Technologies Group, Trusted Information System (2000)
3. Daemen, J., Rijmen, V.: The block cipher Rijndael. In: Smart Card Research and Applications, pp. 277–284. Springer (2000). doi:10.1007/10721064_26
4. De Meulenaer, G., Gosset, F., Standaert, O.X., Pereira, O.: On the energy cost of communication and cryptography in wireless sensor networks. In: IEEE International Conference on Wireless and Mobile Computing Networking and Communications, 2008. WIMOB'08, pp. 580–585. IEEE (2008). doi:10.1109/WiMob.2008.16
5. IEEE 802 Working Group: IEEE standard for local and metropolitan area networks - Part 15.4: Low-rate wireless personal area networks (LR-WPANs). IEEE Std 802, 4–2011 (2011)

6. Karl, H., Willig, A.: Protocols and Architectures for Wireless Sensor Networks. Wiley (2007). doi:10.1002/0470095121
7. Katz, J., Lindell, Y.: Introduction to Modern Cryptography. CRC Press (2014)
8. Law, Y.W., Doumen, J., Hartel, P.: Survey and benchmark of block ciphers for wireless sensor networks. ACM Trans. Sensor Netw. (TOSN) **2**(1), 65–93 (2006). doi:10.1145/1138127.1138130
9. Lee, J., Kapitanova, K., Son, S.H.: The price of security in wireless sensor networks. Comput. Netw. **54**(17), 2967–2978 (2010). doi:10.1016/j.comnet.2010.05.011
10. Lipmaa, H., Wagner, D., Rogaway, P.: Comments to NIST concerning AES modes of operation: CTR-mode encryption (2000)
11. McGrew, D., Viega, J.: The Galois/Counter mode of operation (GCM). Submission to NIST. http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/gcm/gcm-spec.pdf (2004)
12. Panait, C., Dragomir, D.: Measuring the performance and energy consumption of AES in wireless sensor networks. In: Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, pp. 1261–1226 (2015). doi:10.15439/978-83-60810-66-8
13. Schneier, B.: Applied Cryptography: Protocols, Algorithms, and Source Code in C. Wiley (1996)
14. Sen, J.: Routing security issues in wireless sensor networks: attacks and defenses. In: Seah, W., Tan, Y.K. (eds.) Sustainable Wireless Sensor Networks, pp. 279–309. InTech (2010). 10.5772/663
15. Stallings, W.: Cryptography and Network Security—Principles and Practice, 5th edn. Pearson Education (2011)
16. Voinescu, A., Tudose, D., Dragomir, D.: A lightweight, versatile gateway platform for wireless sensor networks. In: Networking in Education and Research, 2013 RoEduNet International Conference 12th Edition, pp. 1–4. IEEE (2013). doi:10.1109/RoEduNet.2013.6714202
17. Wang, Y., Attebury, G., Ramamurthy, B.: A survey of security issues in wireless sensor networks (2006). doi:10.1109/COMST.2006.315852
18. Whiting, D., Housley, R., Ferguson, N.: AES encryption & authentication using CTR mode & CBC-MAC. IEEE P802, 11 (2002)
19. Zhang, F., Dojen, R., Coffey, T.: Comparative performance and energy consumption analysis of different AES implementations on a wireless sensor network node. Int. J. Sensor Netw. **10**(4), 192–201 (2011). doi:10.1504/IJSNET.2011.042767

Computer Support for Risk Management in Critical Infrastructures

Andrzej Bialas

Abstract The paper deals with a methodology for the assessment and management of risk in critical infrastructures. A ready-made risk manager, which supports information security- and business continuity management systems, was adapted to a new application domain—critical infrastructure protection and was used in the EU Ciras project as one of its three basic pillars. First, the author reviewed security issues in critical infrastructures, with special focus on risk management. On this basis the assumptions were discussed how to adapt the ready-made risk manager for this domain. The experimentation tool was configured, including risk measures and system dictionaries. The operations of the tool were illustrated by examples from a case study performed in a previous work. The case study dealt with the collaborating railway- and energy critical infrastructures. The aim of this research is to assess the usefulness of such approach and to acquire knowledge for future project works.

Keywords Critical infrastructure • Risk management • Interdependencies • Bow-tie model • Risk management software

1 Introduction

The paper is an expanded and updated version of the paper "Experimentation tool for critical infrastructures risk management" [1], presented at the 3rd International Conference on Innovative Network Systems and Applications within multi-conference 2015 Federated Conference on Computer Science and Information Systems.

Critical infrastructures (CIs) are understood as large scale infrastructures whose degradation, disruption or destruction would have a considerable impact on the citizens' health, safety, security or well-being or would threaten the functioning of

A. Bialas (✉)
Institute of Innovative Technologies EMAG, Leopolda 31, 40-189 Katowice, Poland
e-mail: andrzej.bialas@ibemag.pl

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_12

governments and/or economies. Such infrastructures are, for example, energy-, oil-, gas-, finance-, transport-, telecommunications-, and health sectors. CIs provide products and services of key importance for today's modern societies. They form an extensive, complex network of processes and assets to facilitate exchange of different services between particular infrastructures and, first and foremost, to provide services for the economy, government and citizens. The networking brings many benefits but it is accompanied by new risks which may disturb processes and breach assets engaged in these processes. Due to CIs mutual relationships, not only services are exchanged but also threats are propagated—disruptions in a certain CI may cause dire effects in others. The most important threats and hazards are: natural disasters and catastrophes, technical disasters and failures, espionage, international crime, physical- and cyber terrorism. There are a number of programmes and activities which are part of a new, holistic approach to CI protection. They all come under the term critical infrastructure protection (CIP).

The protection of critical infrastructures has become a serious issue in well developed countries, including the European Union (EU) countries. The European Council (EC) Directive [2] lays down the specifics about the CIP related needs on the EU and member state levels. The Directive formulates the rules of the CI identification based on casualties-, economic- and public criteria, risk analysis and management programmes. Additionally, it defines the term ECI (European critical infrastructure) as a critical infrastructure located in member states, whose disruption or destruction would have a significant impact on at least two member states. There are two ECI sectors distinguished in this document:

- energy (electricity, oil, gas),
- transport (road transport, rail transport, air transport, inland waterways transport, ocean and short-sea shipping and ports).

In 2006 the EPCIP programme was launched (European Programme for Critical Infrastructure Protection), concerning critical infrastructures on both European and national level. The revised and more practical implementation of EPCIP can be found in the EU document [3].

An important issue is the CI resilience, which is understood as an ability of a system to react to and recover from unanticipated disturbances and events.

Critical infrastructures protection programmes are based on risk management. The issue of risk management in CIPs remains a challenge. This is proven by dozens of EU or worldwide CIP R&D projects which focus on risk methodologies and tools (FP6, FP7, Horizon 2020, CIPS).

The paper features some researches that are preliminary activities of the Ciras[1] project [4] which was launched by the international consortium comprising ATOS, CESS, and EMAG—the author's organization. The Ciras project aims at the

[1] This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the European Commission cannot be held responsible for any use which may be made of the information contained therein (Grant Agreement clause).

development of a methodology and tool to properly select security measures in the critical infrastructure domain. The project uses three main inputs:

- an extensive review of the state of the art of the risk management methodologies, especially those for critical infrastructure protection,
- conclusions from the organized Ciras stakeholders' workshops,
- an OSCAD-Ciras feasibility study presented in the paper.

The Ciras approach is based on the FP7 ValueSec [5] methodology. The ValueSec decision making process assumes that the proposed security measure (countermeasures) should be:

- able to sufficiently mitigate the risk volume in order to provide security on an accepted level and to provide benefits for stakeholders,
- cost-effective in order not to diminish the efficiency of operations and not to produce unnecessary costs,
- free of social, psychological, political, legal, ethical, economical, technical, environmental, etc. restrictions (called there "qualitative criteria").

To provide data for a decision maker, the Ciras Tool will be equipped with three components corresponding to the above mentioned issues:

- Risk Reduction Assessment (RRA),
- Cost-Benefit-Assessment (CBA),
- Qualitative Criteria Assessment (QCA).

This three pillars approach has been implemented in the Ciras Tool for the critical infrastructure protection domain. This domain is more complicated than the application domains considered in the ValueSec (mass event, mass transportation, communal security planning, air transport security, cyber smart grid attack).

The paper deals with the RRA component and is focused on how to develop or implement it, satisfying the project requirements. RRA should be relatively simple, able to properly manage the risk in critical infrastructures by selecting security measures with right cost-benefits parameters and free of intangible restrictions. The OSCAD[2] software platform [6] was considered one of the candidates for the RRA component. The paper presents researches which allow to assess whether this platform is able to satisfy the project requirements and whether it can be used as the RRA component of the Ciras Tool. Because the answer to this question is not straightforward, the author performed researches and experiments presented in this paper. To do this, first the experimentation tool, called OSCAD-Ciras, was developed, next a case study was planned, performed and concluded.

The aim of the research presented in the paper is to develop a simple configurable risk management tool for CIs which will be able: to analyze causes and

[2] developed at the Institute of Innovative Technologies EMAG within a project co-financed by the National Centre for Research and Development (NCBiR).

consequences of hazardous events, to process all risk-relevant data, and to consider interdependencies.

The motivation for researches presented in the paper is to get input for the Ciras project. During the experimentations the standard OSCAD software was adapted to the CI application domain according to the requirements specified in the paper [7]. The key issue was if these requirements can be implemented on the ready-made software or some software modifications or extensions are needed. During the case study, the OSCAD platform was properly configured and equipped with the near real data related to the project domain. The case study example concerns the railway CI interrelated with the energy CI. This way the CI dedicated OSCAD-Ciras experimentation tool was developed. The aim of the experimentation is to acquire indispensable knowledge about the usability of this risk manager to work as the RRA component of the Ciras Tool. The RRA component should be able to assess risk before a measure is implemented and reassess the risk for a certain number of security measures alternatives considered for the implementation. This information is supplemented by cost-benefits- and qualitative criteria related factors. RRA should be able to exchange information with the CBA and QCA components during the decision process dealing with the security measures selection.

Section 2 of the paper includes an introduction to risk management in critical infrastructures. Section 3 summarizes the preferred features of the risk management tool discussed in the work [7]. Section 4 presents the functionality of the OSCAD software platform, while Sect. 5 gives the specifics of OSCAD's adaptation to be a CI risk manager and draws some conclusions for future works.

2 Resilience and Risk Management in Critical Infrastructures Protection

Critical infrastructure is a heterogeneous, distributed, adaptive, and very complex socio-technical system which encompasses hardware, software, liveware, environmental, management, and organizational elements. The basic objective of a CI is to provide products and/or services for the society. In order to reach this objective, this complex socio-technical system must be well harmonized, the disturbances within the system must be under control, the system has to work smoothly, and the assets needed to perform the job have to be well protected. The CI countermeasures, selected on the basis of risk, should be properly managed and composed into CIP programmes.

Some critical infrastructures (systems) collaborate with each other, e.g. electricity, rail transport, gas, oil, telecommunications Thus they constitute a more complex structure, called a system-of-systems (SoS). SoS includes different mutual dependencies (i.e. interdependencies) that exist within particular CIs. An interdependency [8] is a mutual relationship between two infrastructures (systems) where the state of each infrastructure influences or is correlated to the state of the other [9].

The CIs failures are usually causally linked, which means that the impacts of incidents may pass across different CIs. In addition, certain CI-specific effects are observed:

- a cascading effect is based on a sequence of component failures [10]. The first failure shifts its load to one or more nearby components. Then these components fail and, in turn, shift their loads to other components. This sequence is repeated;
- an escalating failure happens when there is a disruption in one infrastructure which causes an independent disruption in another infrastructure [8]. The effects of hazardous events may escalate outside the area where they occur and exacerbate the consequences of a given event (in the form of increasing the severity or the time for recovery of the second failure);
- common cause failures are failures implied by a single shared cause and coupling to other systems mechanisms. They may occur almost concurrently.

The interdependencies and related phenomena are not the key issues in this paper but they are taken into account during the risk assessment and management.

Critical infrastructures operators take care about the CI resilience. They apply strategies to deal with disruptive events, mitigate the magnitude of events, shorten their duration, react properly, minimize impacts, recover from a potentially disruptive event, etc. The CI preparedness is very important too, along with the ability to anticipate an event, absorb a threat, adapt the infrastructure to different situations, maintain critical operations and functions in the face of a crisis (robustness), manage resources needed for reaction and recovery, etc.

To ensure the CI resilience, a systematic approach is needed. At the beginning, the critical infrastructure is structurally analyzed and specified. The most critical elements and the most vulnerable points are identified, as well as the internal and external relationships (interdependencies). All these results form a static picture of the CI. Using this model, different scenarios can be considered to reveal dynamic properties of the given CI. This analysis can be considered as the simulation of different phenomena, like propagation of dire effects, identifying the impact of certain threats, common failures, assessing the effectiveness of the reaction to a given threat or disturbance, performing the CI recovery process, etc. This analysis yields a set of the most dangerous and prioritized risk scenarios, which can be further analyzed on a more detailed level. Generally, due to the CIs complexity, it is impossible to analyze all identified risk scenarios. For this reason only the most serious ones are chosen to be encompassed by the risk management process.

To ensure the preparedness and incident response ability, it is necessary to identify the risk source, risk character and value. What is more, it is important to apply the right countermeasure and embed it into the risk management framework, sometimes supported by tools.

Due to CIs complexity, interdependencies, specific effects, different abstract levels applied to manage CIs, and other factors, the comprehensive approach to risk management in critical infrastructures still remains a challenge.

Different risk management methodologies and tools are a subject of current R&D on the national and international levels, including the EU level. The following knowledge sources contain very comprehensive reviews of the R&D results:

- the report [11] of the Institute for the Protection and Security of the Citizen, EC Joint Research Centre (JRC); the report assesses and summarizes 21 existing risk management methodologies/tools on the EU and global level; it identifies their gaps and prepares the ground for R&D in this field, like Ciras project [4];
- the EURACOM report [12]; it features a study of 11 risk assessment methodologies related to the energy sector;
- the book [9]; in its Appendix C it provides a comparison of the features of about 22 commonly used risk analysis methods;
- the ISO 31010 standard [13] characterizes about 30 risk assessment methods for different applications;
- the ENISA website [14] includes an inventory of risk management/assessment methods, mostly ICT-focused.

A very exhaustive review of the state of the art is provided in [15]. The objective of this document was to select the most favourable methods/tools features for implementation during the Ciras project. The document summarizes the assessment of 14 methods (from 46 preselected), 22 tools (from 150 preselected) and considers 19 projects and 8 frameworks.

Usually, each of these methods/tools is focused on a restricted domain and does not address properly the holistic view and resilience. Therefore, it is an open question how to consider CIs interdependencies in the risk management process.

3 Basic Features of Risk Manager for Critical Infrastructures

The paper [7] discusses the basic requirements of the risk manager to be applied in critical infrastructure protection. This section provides a short overview of these issues.

3.1 Conceptual Model of the Risk Manager

The implementation of the bow-tie risk concept in the tool brings obvious advantages for CI risk management [7]. The method allows to identify risk pathways and barriers in CIs to prevent or react to undesired consequences or stimulate desired ones.

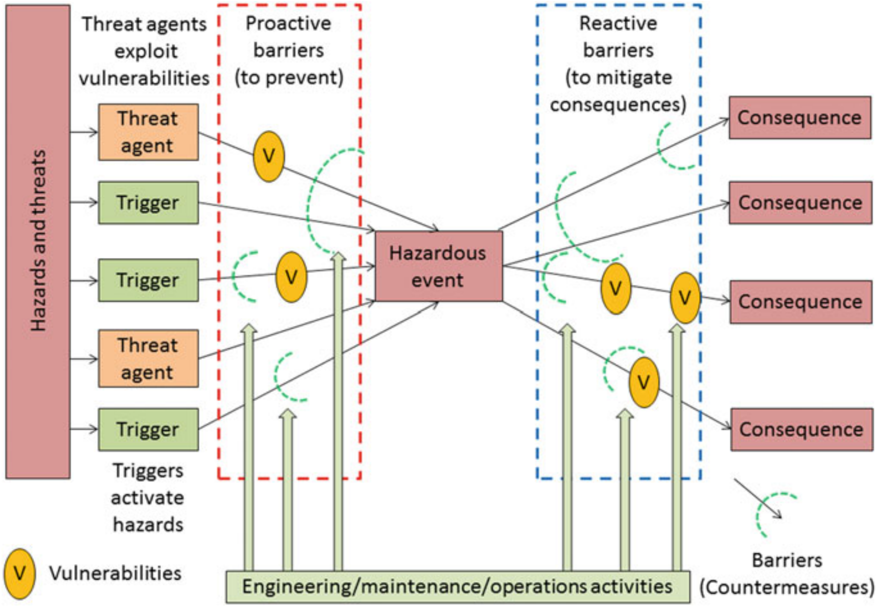

Hazards and threats
Threat agents exploit vulnerabilities
Proactive barriers (to prevent)
Reactive barriers (to mitigate consequences)
Threat agent
Trigger
Trigger
Threat agent
Trigger
Triggers activate hazards
V
Hazardous event
Consequence
Consequence
Consequence
Consequence
V Vulnerabilities
Engineering/maintenance/operations activities
Barriers (Countermeasures)

Fig. 1 General concept of the bow-tie analysis

The bow-tie conceptual model [10, 13] contains multiple and complex causes of the given hazardous event as well as its diversified and multidirectional consequences (Fig. 1).

The triggered hazards or threats, which exploit certain vulnerabilities, can degrade proactive barriers (countermeasures) existing in the system. This situation may result in an event which is hazardous for assets. Such an event usually has diversified and multidirectional consequences. To mitigate them, reactive barriers are applied. These barriers can be weakened or even removed by vulnerabilities. Generally, barriers are identified with different kinds of countermeasures. The countermeasures are applied with respect to the risk value and are monitored and maintained—according to the risk management principles. The bow-tie model is focused on risk assessment and can be used to reassess the risk after new or updated barriers are applied.

The bow-tie analysis is based on this model. For each knot representing a hazardous event, certain causes are identified along with related preventive barriers. Next all potential consequences of the hazardous event are listed, with respect to the reactive barriers. Management activities (engineering, maintenance, trainings, monitoring) support both groups of barriers.

The bow-tie model includes the cause analysis and the consequences analysis. These analyses can be implemented in less or more complex ways [13], e.g. with the use of FTA (Fault tree analysis) [16] or ETA (Event tree analysis) [17].

This model does not have any analysis of interdependencies, therefore it is necessary to supplement it in this respect.

3.2 Risk Related Data and the Risk Register

The tool should support a CI owner in elaborating and maintaining a risk register serving as an inventory of hazardous events. The listed items (data records) should include at a minimum: related hazards/threats, a possible corresponding hazardous event, probability of the event and its consequences. The risk management process is performed during the CI life cycle, so the risk register can be continuously updated. There are some data associated with each item of the risk register, like assets, societal critical functions (SCF) which ensure the basic needs of a society (e.g.: life and health, energy supply, law and order, national security), hazards, threats, vulnerabilities, countermeasures, etc.

3.3 Risk Assessment Parameters and Assessment Process

Risk measures, such as event likelihood and consequences, depend on the applied methodology and are broadly described in literature [9, 10].

The likelihood of a hazardous event can be assessed with the use of a predefined scale, e.g.: fairly normal, occasional, possible, remote, improbable. The consequence severity can be assessed in different dimensions with the use of enumerative scales, e.g.: negligible, minor, major, catastrophic damages. The risk is a function of both likelihood and consequences usually expressed by a risk matrix, as presented in [7].

3.4 Considering the CI Specific Issues

The risk assessment/management methods/tools (Sect. 2) are focused on the given environment which has certain protected assets and processes. However, they do not consider interdependencies between other environments. The interdependencies have be included in the risk management process as they are essential for the CI protection.

Please note that the given hazardous event may be invoked by internal factors as well as external factors, including these coming from other CIs. Apart from this, the hazardous event may cause internal damages and/or may cause problems in the coupled external infrastructures. The risk assessment methodology should be able to take into account the CI specific phenomena mentioned in Sect. 2.

4 OSCAD Software as the Implementation Platform

The OSCAD software was originally elaborated to support business continuity management according to ISO 22301 and information security management according to ISO/IEC 27001. It is used to control factors which disturb business processes or breach information assets in an institution (business, public) and which may bring about negative consequences, to minimize losses when an incident occurs, and to support the organization in its recovery processes.

OSCAD is open and flexible. Therefore, after certain modifications, it can be implemented in different application domains, e.g.: flood protection [18], railway safety management systems [19] and coal mining [20]. The paper discusses the possibility to adapt OSCAD to the CI risk management domain.

The OSCAD platform offers an extensive functionality, though from the risk management perspective, only the following will be useful:

- system dictionaries—allowing to predefine threats, vulnerabilities, countermeasures, categories of assets, risk measures parameters, like likelihood and consequences,
- configuration facilities—to describe the given CI, to define risk matrix, to set analytical parameters, etc.,
- asset and process inventory—to specify the CIs protected assets and/or processes, whose breaches and disturbances are to be considered during the risk assessment process,
- risk assessment and management facilities—the core functionality discussed in this paper.

OSCAD is equipped with tools which analyze causes of hazardous events:

- AORA—Asset Oriented Risk Analyzer,
- PORA—Process Oriented Risk Analyzer,

and tools which analyze their multidimensional consequences:

- ABIA—Asset Oriented Business Impact Analyzer,
- PBIA—Process Oriented Business Impact Analyzer.

The selection of countermeasures is based on the assessed risk value and their total investment/maintenance costs. After selecting for implementation a given countermeasure or a set of measures, the risk is reassessed with respect to the acceptance level assumed for the organization.

5 Implementation of Risk Manager Requirements on the OSCAD Software

The following section presents the author's proposals how to implement the above-listed requirements into the existing OSCAD [6] software platform.

5.1 Bow-Tie Model Implementation on the OSCAD Software Platform

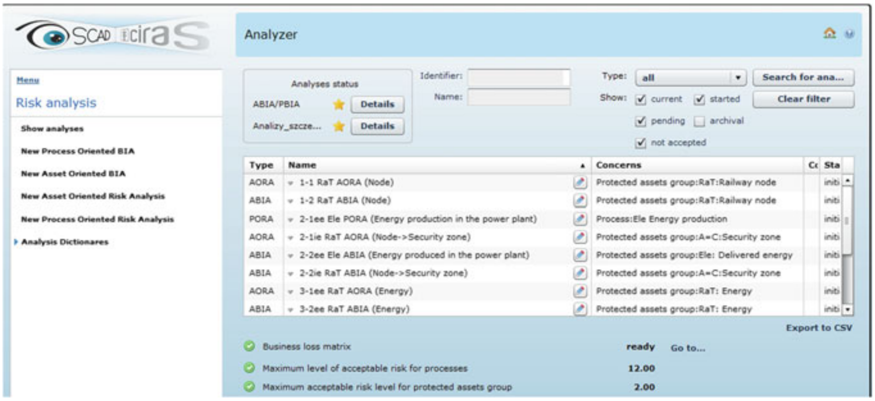
The bow-tie conceptual model is not directly implemented in OSCAD. However, the OSCAD risk analyzing tools can be used to compose it.

The cause analysis part of the bow-tie model is implemented on the basis of AORA or PORA. AORA is responsible for the analysis of each threat-vulnerability pair which can breach the given asset. PORA does the same with respect to the given process.

The consequences analysis part of the bow-tie model is implemented on the basis of ABIA or PBIA. For a given asset (process), which is under a hazardous event, multi-dimensional consequences can be assessed with the use of the loss matrix.

Both parts of the bow-tie model are not coupled directly by the hazardous event, but by the threatened asset (or process) related to this event.

Figure 2 shows examples of analyses composing the bow-tie conceptual model. The left part of the figure presents the "Risk analysis" menu of the OSCAD experimentation tool, called here OSCAD-Ciras. The right side of the figure presents the list of performed analyses. Please note that two analyses, corresponding to the same asset (or process), compose the bow-tie model, e.g.:



ABIA/PBIA ★
Analizy_szcze... ★

Identifier:
Name:
Type: all
Show: ☑ current ☑ started ☑ pending □ archival ☑ not accepted

Type	Name	Concerns	Cc	Sta
AORA	1-1 RaT AORA (Node)	Protected assets group:RaT:Railway node		initi
ABIA	1-2 RaT ABIA (Node)	Protected assets group:RaT:Railway node		initi
PORA	2-1ee Ele PORA (Energy production in the power plant)	Process:Ele Energy production		initi
AORA	2-1ie RaT AORA (Node->Security zone)	Protected assets group:A=C:Security zone		initi
ABIA	2-2ee Ele ABIA (Energy produced in the power plant)	Protected assets group:Ele: Delivered energy		initi
ABIA	2-2ie RaT ABIA (Node->Security zone)	Protected assets group:A=C:Security zone		initi
AORA	3-1ee RaT AORA (Energy)	Protected assets group:RaT: Energy		initi
ABIA	3-2ee RaT ABIA (Energy)	Protected assets group:RaT: Energy		initi

Business loss matrix	ready
Maximum level of acceptable risk for processes	12.00
Maximum acceptable risk level for protected assets group	2.00

Fig. 2 OSCAD risk analyses composing the bow-tie model. OSCAD risk manager elaborated in the EMAG Institute. (Screen shot prepared by the author, 2015)

- "1-1 RaT AORA (Node)" (left part of the bow-tie model),
- "1-2 RaT ABIA (Node)" (right part of the bow-tie model).

Both above mentioned analyses create a pair related to the railway node. Please note that this node can be considered as an element of the Railway transport (RaT) European critical infrastructure (ECI) [2] (see Sect. 5.2).

The following preliminary naming convention for risk assessments is assumed:

- iteration number (1-primary, 2-secondary, 3-third iteration, etc.) followed by "-",
- index of assessment (1 for AORA/PORA, 2 for ABIA/PBIA),
- optional suffix for secondary effects assessment identified during BIA: "ie" or "ee" (see Sect. 5.3),
- CI acronym, e.g. "RaT",
- kind of assessment acronym with the asset in the parentheses, e.g. "AORA (Node->Security zone)".

Remark 1 In the OSCAD tool both asset oriented (AORA-ABIA) and process oriented (PORA-PBIA) analyses can be performed.

5.2 Representation of the Risk Register and Risk Related Data in OSCAD

The basic risk-related data are assets being part of critical infrastructures which need protection.

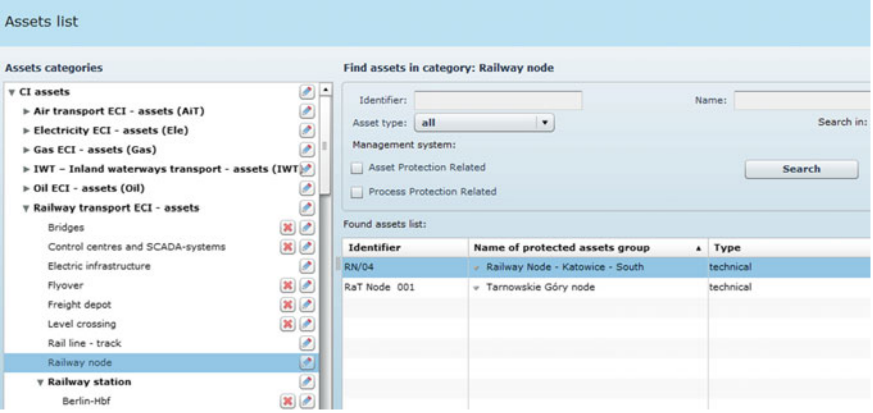
The general ECI (European CI) taxonomy specified in the EC Directive [2] is implemented in OSCAD as a hierarchical structure. The assets belonging to the given ECI are preceded by a label standing for a CI name: Ele (Electricity), Oil (Oil), Gas (Gas), RoT (Road Transport), RaT (Rail Transport), AiT (Air Transport), IWT (Inland Waterways Transport), Sea (Ocean and short-sea shipping and ports).

The left part of Fig. 3 shows the assets hierarchy, while the right part points at the instance "Katowice—South" of the asset group "RaT:Railway node".

All CI assets can be specified hierarchically according to the stakeholders' needs with respect to the number of hierarchy levels. It is possible to create, around the given primary asset, a group of related secondary assets (technical, personal, immaterial, playing role of countermeasures, etc.). This group of assets can be composed in the assets inventory module. The assets can be defined on the general or detailed levels.

Remark 2 OSCAD-Ciras allows to consider the given critical infrastructure on different abstract levels, e.g. on the CI operator level, on the CI particular components levels.

For each of the protected assets, the AORA analysis can be performed. PORA can be done for the processes in a similar way. Using the OSCAD process



Assets list

Assets categories

- ▾ CI assets
  - ▸ Air transport ECI - assets (AiT)
  - ▸ Electricity ECI - assets (Ele)
  - ▸ Gas ECI - assets (Gas)
  - ▸ IWT – Inland waterways transport - assets (IWT
  - ▸ Oil ECI - assets (Oil)
  - ▾ Railway transport ECI - assets
    - Bridges
    - Control centres and SCADA-systems
    - Electric infrastructure
    - Flyover
    - Freight depot
    - Level crossing
    - Rail line - track
    - Railway node
    - ▾ Railway station
      - Berlin-Hbf

Find assets in category: Railway node

Identifier:

Name:

Asset type: all

Search in:

Management system:

□ Asset Protection Related

□ Process Protection Related

Search

Found assets list:

Identifier	Name of protected assets group	Type
RN/04	Railway Node - Katowice - South	technical
RaT Node 001	Tarnowskie Góry node	technical

Fig. 3 Hierarchical structure of protected assets—taxonomy proposed in [2]. (OSCAD screen shot prepared by the author, 2015)

inventory, processes and their subprocesses can be defined on the general or more detailed levels. The paper does not focus on the process-oriented approach.

In critical infrastructures multilayered protection systems are usually applied. In the bow-tie model different barriers (countermeasures) are marked, representing this kind of protection. To perform a risk analysis for different barriers, security zones, etc., which play the role of countermeasures, an auxiliary category is defined: A = C (countermeasures considered as assets), for example "A = C:Security zone" can be added to the "Railway node", and an additional risk analysis for it can be performed (risk analysis in OSCAD is focused on assets or processes, not on countermeasures). This feature allows to take into account internal escalation effects during the risk analysis. This issue will be explained latter.

Remark 3 Certain assets playing the role of countermeasures are distinguished in OSCAD-Ciras (A = C category). This way the countermeasures can be encompassed by the risk assessment process and it is possible to analyze the internal escalation effects.

Generally, a hazardous event can be considered a specific representation of the threat [10]. The formula assumed in OSCAD and specifying the threats scenario is:

[*Threat agent*] exploiting [*vulnerability*] causes [*adverse action*] to [*asset*] or [*process*],

and parameters in square brackets have to be refined.

To put it simply, a threat agent, representing a force which initiates the scenario, is identified as the hazard trigger. Assuming that the phrase "exploiting [*vulnerability*]" concerns threats only, the following remark can be specified.

Remark 4 In the OSCAD-Ciras tool threats and hazards have the same representation—they are simply the "OSCAD threats" in system dictionaries.

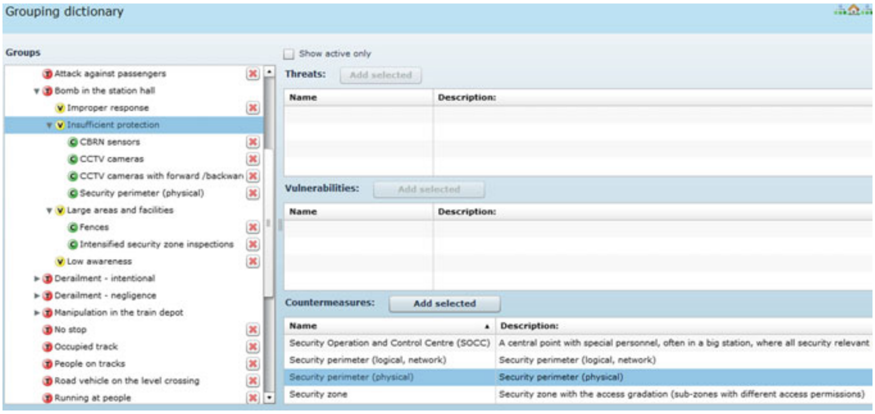
The threat specification includes terms essential for the risk analysis. Threats specified during the AORA/PORA analyses play the role of risk register items. OSCAD has the incident management functionality (registering, assessment, solving, lessons learnt, statistics). The incidents which have already occurred are assigned to the threat items too. For this reason, the predicted risk scenarios and occurred incidents (materialized risk scenarios) are consistent. OSCAD is able to build statistics of incidents. This auxiliary option related to real-time risk management, not discussed here, can be used for more advanced applications in the future.

To sum up, OSCAD defines the risk register as a set of risk scenarios resulting from AORA or PORA and compatible with the incident inventory.

OSCAD has predefined lists of threats, vulnerabilities and countermeasures. Though they are flat, a special grouping mechanism is applied as the hierarchical grouping dictionary. On the upper hierarchy level these threats can be ordered first according to critical infrastructures taxonomy, and then according to their character. For OSCAD-Ciras the following threats categories are assumed: Behavioural/Social, Natural/Force majeure, Organizational, Technological. For the given threat (T), relevant vulnerabilities (V) are given, and to the given pair threat-vulnerability, recommended countermeasures (C) can be assigned.

Figure 4 presents the hierarchical structure of the grouping dictionary and some examples concerning railway transport:

- four vulnerabilities are assigned to the threat “Bomb in the station hall”: “Improper response”, “Insufficient protection”, “Large areas and facilities” and “Low awareness”,
- for the threat-vulnerability pair “Bomb in the station hall”-“Large areas and facilities”, the following countermeasures are predefined: “Fences”, “Intensified security zone inspections”.



- T Attack against passengers
- T Bomb in the station hall
  - V Improper response
  - V Insufficient protection
    - C CBRN sensors
    - C CCTV cameras
    - C CCTV cameras with forward /backwan
    - C Security perimeter (physical)
  - V Large areas and facilities
    - C Fences
    - C Intensified security zone inspections
  - V Low awareness
- T Derailment - intentional
- T Derailment - negligence
- T Manipulation in the train depot
- T No stop
- T Occupied track
- T People on tracks
- T Road vehicle on the level crossing
- T Running at people

Name	Description:

Name	Description:

Name ▲	Description:
Security Operation and Control Centre (SOCC)	A central point with special personnel, often in a big station, where all security relevant
Security perimeter (logical, network)	Security perimeter (logical, network)
Security perimeter (physical)	Security perimeter (physical)
Security zone	Security zone with the access gradation (sub-zones with different access permissions)

Fig. 4 Grouping dictionary with data relevant to rail transport. (OSCAD screen shot prepared by the author, 2015)

Remark 5 The OSCAD-Ciras tool is very flexible in creating dictionaries of threats, vulnerabilities and countermeasures (different levels of detail, predefined categories, domain-specific dictionaries, grouping of the predefined items). These predefined relations speed up the countermeasures selection during the risk management process.

5.3 Risk Assessment Parameters and Assessment Process in OSCAD

For the AORA and PORA analyses two issues should be defined: likelihood of the event and its consequences. For the experimentation purpose the likelihood and consequences measures were elaborated on the common literature basis. The scales of measures are discussed in [7] and summarized in Tables 2 and 1 of this publication.

Figure 5 shows the implementation of the likelihood scale of measure from [7]/Table 2 in OSCAD-Ciras.

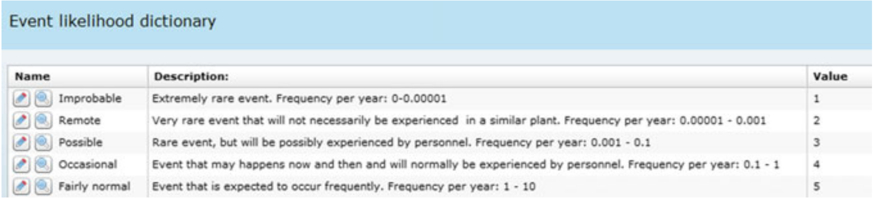
Figure 6 shows the implementation of the scale of measure of consequences presented in [7]/Table 1 in OSCAD-Ciras.

The risk value (AORA/PORA) is calculated with the use of a simple formula:

$$\text{Risk value} = \text{Event likelihood} * \text{Event consequences} \tag{1}$$

The scales of measures should be defined for the ABIA/PBIA analyses as well.

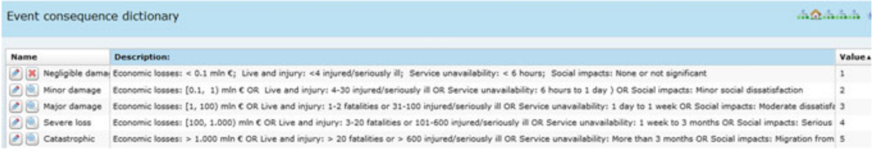
The measures of multidimensional consequences of the hazardous event (Fig. 7) are key issues for the ABIA/PBIA analyses. Three categories of consequences are distinguished:



Event likelihood dictionary

Name	Description:	Value
Improbable	Extremely rare event. Frequency per year: 0-0.00001	1
Remote	Very rare event that will not necessarily be experienced in a similar plant. Frequency per year: 0.00001 - 0.001	2
Possible	Rare event, but will be possibly experienced by personnel. Frequency per year: 0.001 - 0.1	3
Occasional	Event that may happens now and then and will normally be experienced by personnel. Frequency per year: 0.1 - 1	4
Fairly normal	Event that is expected to occur frequently. Frequency per year: 1 - 10	5

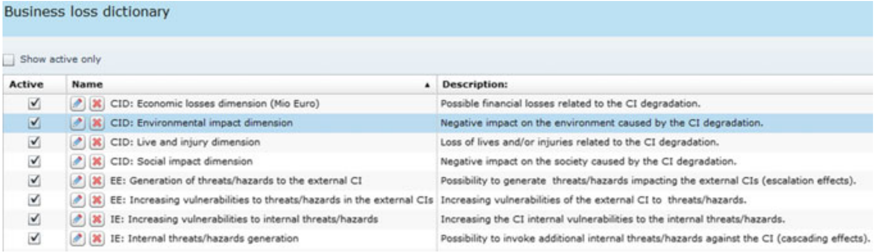
Fig. 5 Event likelihood scale of measure. (OSCAD screen shot prepared by the author, 2015)



Event consequence dictionary

Name	Description:	Value
Negligible dama	Economic losses: < 0.1 mln €; Live and injury: <4 injured/seriously ill; Service unavailability: < 6 hours; Social impacts: None or not significant	1
Minor damage	Economic losses: [0.1, 1) mln € OR Live and injury: 4-30 injured/seriously ill OR Service unavailability: 6 hours to 1 day) OR Social impacts: Minor social dissatisfaction	2
Major damage	Economic losses: [1, 100) mln € OR Live and injury: 1-2 fatalities or 31-100 injured/seriously ill OR Service unavailability: 1 day to 1 week OR Social impacts: Moderate dissatisfa	3
Severe loss	Economic losses: [100, 1.000) mln € OR Live and injury: 3-20 fatalities or 101-600 injured/seriously ill OR Service unavailability: 1 week to 3 months OR Social impacts: Serious	4
Catastrophic	Economic losses: > 1.000 mln € OR Live and injury: > 20 fatalities or > 600 injured/seriously ill OR Service unavailability: More than 3 months OR Social impacts: Migration from	5

Fig. 6 Event consequences scale of measure. (OSCAD screen shot prepared by the author, 2015)



Business loss dictionary

☐ Show active only

Active	Name	Description:
☑	CID: Economic losses dimension (Mio Euro)	Possible financial losses related to the CI degradation.
☑	CID: Environmental impact dimension	Negative impact on the environment caused by the CI degradation.
☑	CID: Live and injury dimension	Loss of lives and/or injuries related to the CI degradation.
☑	CID: Social impact dimension	Negative impact on the society caused by the CI degradation.
☑	EE: Generation of threats/hazards to the external CI	Possibility to generate threats/hazards impacting the external CIs (escalation effects).
☑	EE: Increasing vulnerabilities to threats/hazards in the external CIs	Increasing vulnerabilities of the external CI to threats/hazards.
☑	IE: Increasing vulnerabilities to internal threats/hazards	Increasing the CI internal vulnerabilities to the internal threats/hazards.
☑	IE: Internal threats/hazards generation	Possibility to invoke additional internal threats/hazards against the CI (cascading effects).

Fig. 7 Event impacts measures with CID, IE and EE categories

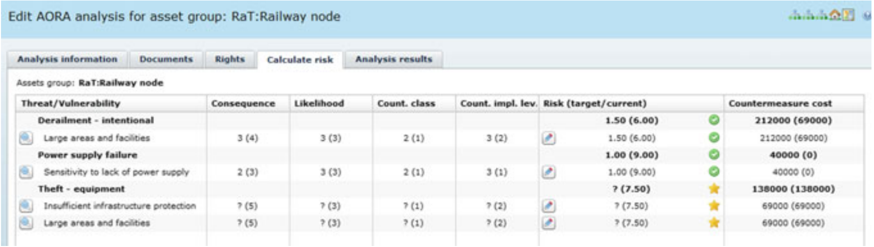
- CID (CI degradation) category, which expresses different kinds of damages within the given CI, like economic losses, environmental impact, loss of lives and injuries of people, social impact;
- IE (Internal escalations) expresses new internally generated threats or new or increased vulnerabilities which influence the considered CI, caused by the hazardous event,
- EE (External escalations) expresses generated threats which impact the external CIs or new or increased vulnerabilities in the external CIs, caused by the hazardous event.

Business loss categories (CID, IE, EE) and their subcategories are used to construct the later discussed BIA matrix, which is the basic tool for the ABIA/PBIA assessment process (Fig. 10).

The implementation of the bow-tie model is presented by the pair AORA-ABIA with respect to the given asset (here: railway node of the RaT infrastructure). The process approach (PORA-PBIA), though possible, is not discussed here.

The aim of AORA is to identify and assess the risk value related to a hazardous event in a railway node as a part of the railway critical infrastructure. Please note that AORA is focused on the assessment of causes of the hazardous event. Its example is shown in Fig. 8.

Please note three threats (“Derailment—intentional”, “Power supply failure”, “Theft—equipment”) and the related vulnerabilities. For each pair



Edit AORA analysis for asset group: RaT:Railway node

Assets group: **RaT:Railway node**

Threat/Vulnerability	Consequence	Likelihood	Count. class	Count. impl. lev.	Risk (target/current)		Countermeasure cost
Derailment - intentional					**1.50 (6.00)**	✓	**212000 (69000)**
Large areas and facilities	3 (4)	3 (3)	2 (1)	3 (2)	1.50 (6.00)	✓	212000 (69000)
Power supply failure					**1.00 (9.00)**	✓	**40000 (0)**
Sensitivity to lack of power supply	2 (3)	3 (3)	2 (1)	3 (1)	1.00 (9.00)	✓	40000 (0)
Theft - equipment					**? (7.50)**	★	**138000 (138000)**
Insufficient infrastructure protection	? (5)	? (3)	? (1)	? (2)	? (7.50)	★	69000 (69000)
Large areas and facilities	? (5)	? (3)	? (1)	? (2)	? (7.50)	★	69000 (69000)

Fig. 8 Example of the AORA analysis for a railway node. (OSCAD screen shot prepared by the author, 2015)

threat-vulnerability, which has certain influence on the asset, the risk value can be determined according to the above presented formula. Inherent risk ("risk before") is in parentheses, while current risk ("after measures applications")—without parentheses. The same rule applies to the cost of countermeasures. Each pair threat-vulnerability is considered a risk register item.

If the risk value is greater than the risk acceptance level, extra (other) countermeasures can be selected (Fig. 9).

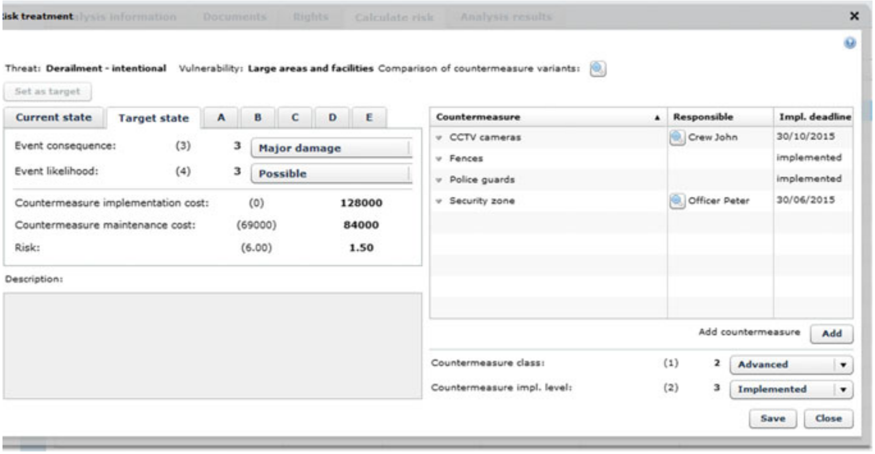
The OSCAD-Ciras risk manager allows to consider up to five security measures alternatives (A–E). The decision maker chooses one as the target variant for implementation. Each alternative should be a coherent and applied together package of countermeasures. Examples of such packages are: "CCTV cameras", "Fences", "Police guards", "Security zone" (Fig. 9). The OSCAD risk manager has more features, not discussed here, like setting the assurance class for the countermeasure, along with its status of implementation, cost, etc.

The aim of ABIA is to identify and assess multidirectional impacts of the hazardous event breaching the given asset, e.g. the above mentioned railway node belonging to RaT ECI.

The loss matrix (Fig. 10) is the basic ABIA tool. For each CID, IR, EE subcategory, there are some losses are assessed with the use of 5 levels. A number of subcategories and levels are configurable. As a result of these operations, the CI degradation is assessed.

Additionally, we can identify new threats (or vulnerabilities) caused by a hazardous event:

- in the same infrastructure (IE category); they usually concern assets which are also countermeasures (A = C category); the AORA-ABIA pair must be performed with respect to the breached asset (a barrier), e.g. with respect to a



Threat: Derailment - intentional Vulnerability: Large areas and facilities Comparison of countermeasure variants:

Event consequence:	(3)	3 Major damage
Event likelihood:	(4)	3 Possible
Countermeasure implementation cost:	(0)	128000
Countermeasure maintenance cost:	(69000)	84000
Risk:	(6.00)	1.50

Description:

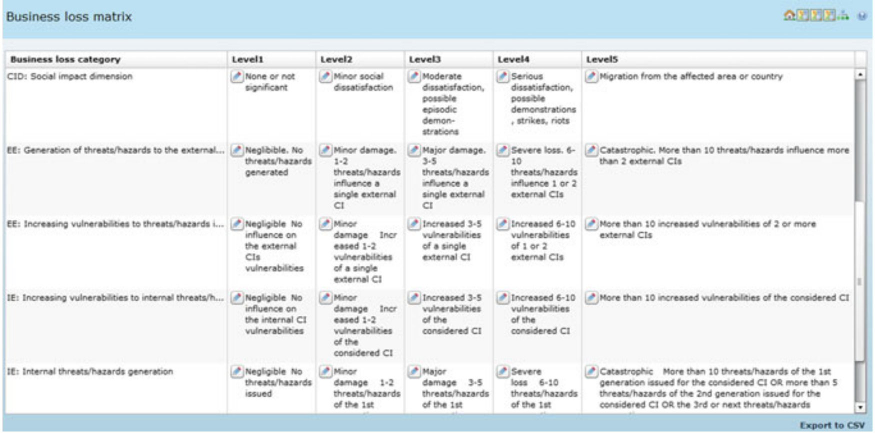
Countermeasure	Responsible	Impl. deadline
CCTV cameras	Crew John	30/10/2015
Fences		implemented
Police guards		implemented
Security zone	Officer Peter	30/06/2015

Add countermeasure

Countermeasure class: (1) 2 Advanced

Countermeasure impl. level: (2) 3 Implemented

Fig. 9 CI risk management—countermeasures selection in OSCAD-Ciras. (Screen shot prepared by the author, 2015)



Business loss category	Level1	Level2	Level3	Level4	Level5
CID: Social impact dimension	None or not significant	Minor social dissatisfaction	Moderate dissatisfaction, possible episodic demon-strations	Serious dissatisfaction, possible demonstrations , strikes, riots	Migration from the affected area or country
EE: Generation of threats/hazards to the external...	Neglibible. No threats/hazards generated	Minor damage. 1-2 threats/hazards influence a single external CI	Major damage. 3-5 threats/hazards influence a single external CI	Severe loss. 6-10 threats/hazards influence 1 or 2 external CIs	Catastrophic. More than 10 threats/hazards influence more than 2 external CIs
EE: Increasing vulnerabilities to threats/hazards i...	Negligible No influence on the external CIs vulnerabilities	Minor damage Incr eased 1-2 vulnerabilities of a single external CI	Increased 3-5 vulnerabilities of a single external CI	Increased 6-10 vulnerabilities of 1 or 2 external CIs	More than 10 increased vulnerabilities of 2 or more external CIs
IE: Increasing vulnerabilities to internal threats/h...	Negligible No influence on the internal CI vulnerabilities	Minor damage Incr eased 1-2 vulnerabilities of the considered CI	Increased 3-5 vulnerabilities of the considered CI	Increased 6-10 vulnerabilities of the considered CI	More than 10 increased vulnerabilities of the considered CI
IE: Internal threats/hazards generation	Negligible No threats/hazards issued	Minor damage 1-2 threats/hazards of the 1st	Major damage 3-5 threats/hazards of the 1st	Severe loss 6-10 threats/hazards of the 1st	Catastrophic More than 10 threats/hazards of the 1st generation issued for the considered CI OR more than 5 threats/hazards of the 2nd generation issued for the considered CI OR the 3rd or next threats/hazards

Fig. 10 The loss matrix as the basic BIA tool. (OSCAD screen shot prepared by the author, 2015)

breached security zone (A = C), to identify internal secondary effects resulting from the breach;

- in the dependent infrastructures (EE category); similarly, threats/vulnerabilities which influence external CIs are identified; this requires an extra AORA-ABIA pair for external CIs with respect to the affected asset (EE category).

5.4 Considering the CI Specific Issues in OSCAD-Ciras

OSCAD-Ciras does not have a specific tool to analyze interdependencies, particularly the strength of coupling between CIs. This task must be solved outside the system. One of the ways to do it is to prepare a map of interdependent CIs. With this map it is possible to further analyze the risk within a set of interdependent infrastructures.

OSCAD-Ciras is equipped with facilities allowing to explicitly distinguish CI internal and external causes of hazardous events, internal non-escalating consequences, consequences generating hazards/threats in the same infrastructure, and consequences generating external hazards/threats for other collaborating infrastructures.

Remark 6 Before the risk assessment/management process starts in OSCAD-Ciras, the interdependencies should be known from the perspective of the assessed CI. It should be clear which CIs depend on the given CI (they can be affected) and on which CIs the given CI depends (which CIs can affect the given CI).

6 Conclusions

The paper presents a part of preliminary researches related to the Ciras project. It is focused on the OSCAD-Ciras experimentation tool whose validation is presented in the paper [21]. The presented validation experiment encompasses the following simple scenarios:

1. A hazardous event is triggered in the railway CI causing damages in an important railway node. Apart from the RaT CI degradation (CID), the node security zone is breached (IE) and the coal transport for the neighbor power plant (Ele CI) is blocked in the node (EE).
2. The damaged security zone makes the node more vulnerable to thefts and vandalism. To assess this situation, a pair of AORA-ABIA is launched to check secondary effects (internal escalation) within the railway node.
3. The blocked coal transport (external threat to Ele CI) implies an extra pair AORA-ABIA for Ele CI to check if power production was disturbed (vulnerability: reduced stock of fuel). It is revealed that the power production disturbance affects the railway CI (a negative, backward impact caused by breaching the railway node).
4. The railway power supply is assessed by the next pair of AORA-ABIA, and no extra external escalation is detected because the railway uses redundant power lines.

This short feasibility study confirms the possibility to adapt the ready-made OSCAD platform to CI risk management according to the previously [7] identified requirements. This paper is focused on the experimentation tool, exemplifying it by data examples from the mentioned feasibility study. The ready-made OSCAD software platform was configured and filled in with the data related to the railway CI collaborating with the electricity CI. The data include: assets, threats, vulnerabilities, countermeasures, risk assessment parameters, and formulas.

The OSCAD-Ciras tool offers extensive software support for the risk management process in critical infrastructures. It was the basis of the validation experiment [21] which confirmed the possibility to use it as the RRA component in the Ciras framework. Apart from the data setup, configuration, small GUI modification, no software changes were needed by now. However, for better integration of the RRA, CBA and QCA pillars, such changes will be necessary in the future. They will encompass new web services to exchange information and the GUI extension (see Fig. 9), e.g.:

- to obtain investment costs, future costs and future benefits related to the countermeasures and/or security alternatives from the CBA component,
- to obtain score values related to the countermeasures and/or security alternatives from the QCA component.

These two issues have been solved by the project consortium and are out of scope of this paper.

During the presented preliminary Ciras research, knowledge is gathered about the risk management process in critical infrastructures. It was checked if the proposed approach is useful and how far the CI specific phenomena can be considered. The pros, cons and limitations were identified. All these issues are important to define the final shape of the three pillar based Ciras tool by the consortium members.

The novelty of the paper is to introduce the categorization of the hazardous event consequences, to distinguish the direct CI degradation (CID) and the internal (IE) and external (EE) escalation/cascading effects being the CI specific phenomena. Apart from this, the CID impacts can be assessed in a certain number of predefined time horizons (not discussed here).

The main contribution of the paper is the development of a configurable risk management tool for critical infrastructures. The paper presents how the previously elaborated requirements are implemented on the ready-made software platform. The research includes: domain data identification, elaboration of the software dictionaries, risk manager configuration and validation on the elaborated scenarios.

The paper proposes a new risk assessment method which considers interdependencies between CIs. The research presented in the paper gives substantial contribution to the CIRAS project. During the experiments there was knowledge acquired about the shape of the key component responsible for risk assessment (RRA) of the CIRAS Tool.

These issues need further researches, especially the definition of adequate risk measures. Please note that AORA and ABIA operate on “consequences”. Their definitions must be harmonized. Different variants of these problem solutions are analyzed by the author and by other project team members.

The second open question is how to manage particular risk assessments (pairs of AORA-ABIA). Please note that the launch of a new pair for secondary effects depends on the results of previous assessments—it has dynamic character. The research on the process oriented risk assessment (PORA-PBIA) is also an open issue.

Acknowledgements The author thanks the colleagues from the CIRAS project consortium for discussing the presented concept.

References

1. Białas, A.: Experimentation tool for critical infrastructures risk management. In: Proceedings of the 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 775–780 ISBN 978-1-4673-4471-5 (Web). IEEE Catalog Number: CFP1385 N-ART (Web)
2. Council Directive 2008/114/EC of 8 December 2008 on the identification and designation of European critical infrastructures and the assessment of the need to improve their protection

3. Commission Staff Working Document on a new approach to the European Programme for Critical Infrastructure Protection Making European Critical Infrastructures more secure. European Commission. Brussels, Aug 28 2013, SWD(2013) 318 final
4. Ciras project, http://cirasproject.eu/ (access date: November 2015)
5. ValueSec project, www.valuesec.eu (access date: November 2015)
6. OSCAD project, http://www.oscad.eu/index.php/en/ (access date: Nov 2015)
7. Bialas, A.: Critical infrastructures risk manager—the basic requirements elaboration. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) Theory and Engineering of Complex Systems and Dependability, Proceedings of the Tenth International Conference on DepCoS-RELCOMEX, June 29–July 3 2015, Brunów, Poland. Advances in Intelligent Systems and Computing, vol. 365, pp. 11–24. Springer, Cham (2015). doi:10.1007978-3-319-19216-1_2
8. Rinaldi, S.M., Peerenboom, J.P., Kelly, T.K.: Identifying, understanding and analyzing critical infrastructure interdependencies. IEEE Control Syst. Mag., 11–25 (2001)
9. Hokstad, P., Utne, I.B., Vatn, J. (Eds): Risk and Interdependencies in Critical Infrastructures: A Guideline for Analysis (Springer Series in Reliability Engineering). Springer, London (2012). doi:10.1007/978-1-4471-4661-2_2
10. Rausand, M.: Risk Assessment: Theory, Methods, and Applications. Series: Statistics in Practice (Book 86). Wiley (2011)
11. Giannopoulos, G., Filippini, R., Schimmer, M.: Risk assessment methodologies for Critical Infrastructure Protection. Part I: A state of the art. European Union (2012)
12. Deliverable D2.1: Common areas of Risk Assessment Methodologies. Euracom (2007)
13. ISO/IEC 31010:2009—Risk Management—Risk Assessment Techniques
14. ENISA: http://rm-inv.enisa.europa.eu/methods. Accessed June 2015
15. Baginski, J., Bialas, A., Rogowski, D. et al.: D1.1—State of the Art of Methods and Tools, CIRAS Deliverable. Responsible: Institute of Innovative Technologies EMAG (February 2015), Dissemination level: RE/CO (i.e. available only for: beneficiaries, stakeholders and European Commission)
16. EN 61025 Fault tree analysis (FTA) (IEC 61025:2006), CENELEC (2007)
17. EN 62502 Event tree analysis (ETA) (IEC 62502:2010), CENELEC (2010)
18. Białas, A.: Risk assessment aspects in mastering the value function of security measures. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New results in dependability and computer systems. Advances in Intelligent and Soft Computing, vol. 224. Springer, Cham, pp. 25–39. http://link.springer.com/chapter/10.1007%2F978-3-319-00945-2_3#page-1 doi:10.1007/978-3-319-00945-2_3
19. Bialas, A.: Computer support for the railway safety management system—first validation results. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.): Proceedings of Ninth International Conference on DepCoS-RELCOMEX. June 30—July 4, 2014, Brunow, Poland. Advances in Intelligent Systems and Computing, vol. 286. Springer, Cham (2014), pp. 81–92. doi:10.1007/978-3-319-07013-1
20. Białas, A.: Business continuity management, information security and assets management in mining, Mechanizacja i Automatyzacja Górnictwa, No 8(510), Instytut Technik Innowacyjnych EMAG, Katowice (2013). English version: pp. 125–138
21. Białas, A.: Research on critical infrastructures risk management. In: Rostański, M., Pikiewicz, P., Buchwald, P. (eds.) Internet in the information Society 2015—10th International Conference Proceedings. Scientific Publishing University of Dąbrowa Górnicza (2015), pp. 93–108

An Innovative Web Platform for Flood Risk Management

João L. Gomes, Gonçalo Jesus, João Rogeiro, Anabela Oliveira, Ricardo Tavares da Costa and André B. Fortunato

Abstract This paper presents an innovative real-time information system for enhanced support to flood risk emergency in urban and nearby coastal areas, targeting multiple users with distinct access privileges. The platform addresses several user requirements such as (1) fast, online access to relevant georeferenced information from wireless sensors, high-resolution forecasts and comprehensive risk analysis; and, (2) the ability for a two-way interaction with civil protection agents in the field. The platform adapts automatically and transparently to any device with data connection. Given its specific purpose, both data protection and tailored-to-purpose products are accounted for through user specific access roles. This paper presents the platform's overall architecture and the technologies adopted for server-side, focusing on communication with the front-end and with the wireless sensor network, and the user interface development, using state-of-the-art frameworks for cross-platform standardized development. The advantages of the adopted solution are demonstrated for the Tagus estuary inundation information system.

Keywords Flood risk management • WebGIS • Monitoring network • Responsive applications

J.L. Gomes · G. Jesus · J. Rogeiro · A. Oliveira (✉)
Laboratório Nacional de Engenharia Civil, Information Technology in Water
and Environment Group, Av. do Brasil 101, 1700-066 Lisbon, Portugal
e-mail: aoliveira@lnec.pt

J.L. Gomes
e-mail: jlgomes.web@gmail.com

G. Jesus
e-mail: gjesus@lnec.pt

J. Rogeiro
e-mail: jrogeiro@lnec.pt

R.T. da Costa · A.B. Fortunato
Laboratório Nacional de Engenharia Civil, Estuaries and Coastal Zones Division,
Av. do Brasil 101, 1700-066 Lisbon, Portugal

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent
Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_13

1 Introduction and Motivation

This paper is an expanded and updated version of the paper "Molines—towards a responsive Web platform for flood forecasting and risk mitigation" [1], presented at the 3rd International Conference on Innovative Network Systems and Applications within multi-conference 2015 Federated Conference on Computer Science and Information Systems.

Natural and hydraulic structures related floods are severe threats to life and property. The main goal of flood risk management in aquatic environments is to reduce human losses and the damages related to floods, and should be supported by adequate hazard monitoring and timely early warning of the events.

Some of the world's most densely populated cities are located in estuarine low-lying areas facing thus a high risk of inundation with a potential for significant economic costs and the loss of lives. These areas are highly vulnerable due to the growing human activity in their margins. Simultaneously, the hazards in these environments are severe due to the combined effects of oceanic, atmospheric and river forcings. Furthermore, they are increasing due to the effects of climate change, such as sea level rise, growing storminess and more extreme river flows. Floods in estuaries are associated to particular climatological conditions, namely very high tidal levels and large fresh-water discharges, or of high tides and storm surge conditions [2]. In addition to these progressive, slow phenomena, that are possible to predict a few days in advance, episodes of very intense and concentrated in time rainfall can lead to urban flooding in areas with insufficient drainage conditions and flash floods in small watershed tributaries to the estuary [3]. The effects of high water levels in estuaries can also be exacerbated by human interventions in the system, particularly in urban areas where drainage system behavior has to be considered.

Recently, the processes of prediction, detection, notification and population warning have become increasingly assured by automated systems, such as SAGE–B [4]. These information systems can be valuable assets for risk management, supporting all fundamental data related to flood events and the emergency elements needed for rescue in the predicted flooded areas. Unfortunately, most flood management systems still suffer from significant functional limitations, due to the difficulties in the access to monitoring data and unreliable, scattered, multiple sources of information, the use of inadequate flood forecasting due to inaccurate modeling tools, that either neglect relevant processes or are coarsely applied to the site at risk, and insufficient sharing of information across multiple emergency actors [5].

With the recent use of reliable automatic data acquisition systems and highly efficient and accurate numerical models, the most important constraints for the operational use of real-time information systems have been minimized, allowing for adequate forecasting of relevant events [6, 7]. The integration of these tools into interactive and flexible computational GIS-based platforms has paved the way to a change of paradigm in routine and emergency management of coastal resources and harbor operations [8, 9]. These platforms take advantage of novel technologies to provide on-line, intuitive and geographically-referenced access to real-time data and model predictions, and to produce on-demand services. However, much remains to be done on the interoperability between data providers and data consumers, cross-platform and multiple users' flexibility and speed of access to data.

The project MOLINES (Modelling floods in estuaries. From the hazard to the critical management) aims at integrating existing and new wireless sensor networks, accurate model forecasts at both urban and estuarine scales and information technology (IT) technology to create a Web platform that can contribute to a fast, coordinated mobilization of emergency agents and other managing entities for a timely response to inundation events in the Tagus estuary.

This paper presents the platform's overall architecture and the technologies adopted for server-side, focusing on the communication with the front-end and with the mobile devices (including citizen data and wireless sensor networks), and the user interface development, using well established frameworks for cross-platform web applications development. The advantages of the adopted solution are demonstrated herein through the Tagus estuary inundation information system, supporting flood emergency for the Lisbon metropolitan area and the sharing of information across the relevant emergency actors. Unlike most existing platforms, the platform presented here is generic, interactive, facilitating the coordination between emergency management agents and the individual contribution of civil protection agents in the field, can be deployed elsewhere and provides a single point of access to all relevant georeferenced inundation information, from the real time forecast GIS layers to the alert bulletin. It aims at contributing to a coordinated strategic planning and emergency response in urban and nearby estuarine regions, optimizing the alert to authorities, duly supported by real time monitoring and predictions of inundation.

This paper describes the platform, its architecture, and all innovative aspects related to the user interface (UI), product creation and choice of technologies. Besides this introduction, Sect. 2 provides a background on IT technologies and platforms for real time information access, identifying the key aspects to be addressed. Section 3 presents the concept and implementation of the solution, focusing on requirements and technology choices. The application to the MOLINES case study is briefly presented in Sect. 4, and Sect. 5 closes the paper with some considerations for future work.

2 Background

Technology is dramatically changing our ability to prepare for and respond to extreme events, facilitating the management of crisis incidents [10]. Information systems and technologies contribute to a better communication and action in complex systems, by helping in disaster response and in collecting information, analyzing it, sharing it, and timely disseminating it to the people at risk. In particular, timely information sharing amongst emergency actors is critical in emergency response operations [11]. Several research projects have been devoted to emergency and disasters management to create modelling and simulation techniques and tools for the emergency management. Relevant examples are the dynamic and adaptive models for operational risk management [12, 13].

Information technology is enhancing disaster management and communications through tools such as computer networks, virtual reality, remote sensing, GIS, and decision support systems. During the mitigation and preparation phases of an emergency, the use of satellite communications and spatial analysis systems can be extremely valuable [14]. In recent years, many web-based emergency response systems have been developed and several studies shown the great complexities surrounding the design of this kind of systems [15]. Often, developments in other areas are overlooked and resources are spent looking for a solution that has been already implemented and proved in other environments. An example of an IT system to manage emergency situations is the Global Disaster Information Network (GDIN www.state.gov/www/issues/relief/gdin.html). More elaborated examples with complex architectures, integrating geographic information systems, spatial databases and the internet are described in [16, 17, 18]. In [17] a WebGIS is presented addressing risk management related issues, providing authenticated users with access to searchable information, depending on their authorization level. Hence, they achieve the goal of having a platform accessible anywhere with an internet connection, and multiple levels of access to different hierarchic roles. This is a similar approach to the one presented herein, except an existing tool has been adapted to the use-case, when compared with a tailored-made solution.

Building on these experiences and in the scope of several projects (INTERREG SPRES; FP7 Prepared), Laboratório Nacional de Engenharia Civil (LNEC) has been developing and applying a suite of Web platforms denoted as WIFF—Water Information Forecast Framework [8] to provide access to real time information to decision makers. These platforms were conceived for a single type of users and to provide full access to real time sensor data and model predictions, constituting at the same time a repository of past information, being available at each deployment site to the relevant end-users. For the SPRES platform, real time products were integrated with emergency planning information (hazard, vulnerability and risk maps as well as mitigation action sheets) to constitute a one-stop-shop for all data relevant to oil spill prevention and mitigation [8].

These platforms take advantage of novel technologies to provide on-line, intuitive and geographically-referenced access to real-time data and model predictions, and to produce on-demand services in support of routine management of coastal resources and harbor operations. Technology support include (a) Drupal, a PHP-based Content Management System, to access model metadata, status and products, (b) map server support (Geoserver) providing Web Map Services (WMS) to allow for geospatial placement of monitoring and forecast products, and model output query capabilities, and (c) Flex, using the OpenScales library to handle geospatial information, for the WebGIS development.

However, the need for interaction between the multiple emergency actors and to have fast access to real time information (of both conventional data streams and on-the-fly in-the-field information during flood emergencies) from several users simultaneously raises new requirements for these new platforms. Additionally, the new system should be cross-platform, i.e., to be built in a way that it is automatically and transparently adaptable to any device with a data connection, providing access to emergency information anywhere.

3 Concept and Implementation of the Solution

The main goal for the platform described herein is to provide a quick and responsive tool for flood risk forecast and assessment. End-users should be able to access information on the platform with no hassle and in any device from anywhere, providing that a data connection is available.

Moreover, since we are working directly with the civil protection agents as project partners, there is a major focus on developing on important issues for them. Specifically, our platform aims at fulfilling ease on usability and providing tools for quick decision taking by them, providing a product tailored for their needs. This tool is also being developed with modularity and reusability in mind, so that very little modifications to the platform code need to be made if the forecast product changes from “water levels” to other variables of interest. To achieve this, back and front-end are bind in such a way that the former provides access to the data while the front-end consumes (via a set of REST services) and displays the products without being content-aware, i.e., it shows data without considering data types. Since the front-end is not content-aware, a new instance of the platform, for visualizing and analyzing new types of data, can be created simply by changing the products made available by the back-end and performing basic adaptations to the front-end. Moreover, since the back-end serves data in a standard way (REST services), the front-end itself may also be substituted by another consuming service, be it a mobile application, a web-page or other data consuming service. This allows for interoperability between platforms, allowing other users to use these services straightforwardly, after being successfully authenticated and authorized.

This flow of information is illustrated in Fig. 1, where a separation of concepts, between back-end and front-end is clearly visible. The back-end consists of an instance of CakePHP, a MVC PHP framework, with a PostgreSQL storage database (with PostGIS extension), coupled with several instances of Geoserver and Perl and Python scripts.

The code developed with CakePHP handles user control and user accesses, based on access control lists and roles, and streamlines access to the database via REST requests from the front-end. Geoserver, an open source server for geospatial data sharing, manages the georeferenced imagery (both in raster and vector formats) and serves them using open standards, such as WMS. This also promotes interoperability by allowing that different systems exchange data with each-other using known standards. Geoserver uses data, both from the PostgreSQL/PostGIS database, but also from results produced by the flood prediction models, in the form of

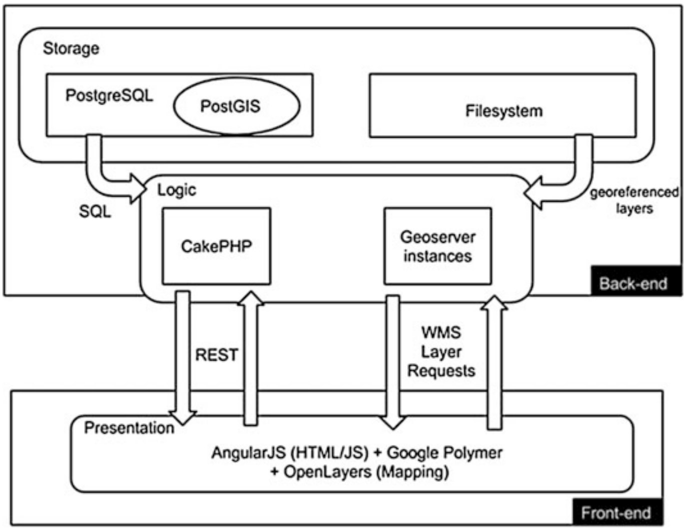


Fig. 1 Technological architecture of the solution

shapefiles, allowing for model results probing directly on the data served through the UI.

Since a fair amount of layer products (generated from result data, stored as shapefiles) are published every day on Geoserver, it tends to become cluttered with data after just a few days, taking much more time to fulfill requests. To reduce these problems some maintenance must be performed on a recurrent basis. The original approach, presented in the first version of this article, consisted of moving existing content to an alternative location on the filesystem daily through a script running as a cronjob. Moved products could be later reloaded into Geoserver, when requested by the user on the front-end. On one hand this strategy allowed for a considerable performance increase in data access and gave the best priority of access to the most recent flood predictions. On the other hand the scripts had to be maintained and up-and-running all the time and space on disk was being taken to keep backups of outdated, easily reproducible data.

Over the course of the project, a more efficient approach has been studied and implemented, which involves caching mechanisms and keeping only the most recent forecast predictions. Instead of keeping track of previous forecast products in the filesystem, these are discarded when a newer forecast is produced. If needed, these can be produced in about 10 min and quickly loaded into Geoserver again. This way we save on available disk space (which is always scarce) and maintain the geographical server clutter free.

Moreover, to accelerate and optimize map image delivery, a geographic web caching mechanism is being used. A caching mechanism is an information technology technique that stores static data so future requests for that information can be served faster, reducing bandwidth usage, server loading and loading lag. The data stored in a cache might be the result of an earlier computation (as is our case), or the duplicate of data stored elsewhere. For this purpose we are using GeoWebCache, which comes bundled with Geoserver. It pre-renders (caches) the layer data that will eventually be served to the user, before the user requests it. This startegy avoids the need for the geographical server to renders requested layers on-demand and provides more responsive maps. To better attend the most frequent user needs, we cache the most requested type of products (the forecast for the current day) and several zoom levels of interest. The cached zoom levels range from a broad zoomed-out view of the case study's domain to a close zoomed-in level, where fine details can be visualized. Although the caching process takes a while to complete and consumes a lot of CPU resources for a considerable amount of time (depending on coverage, zoom levels and number of layer products), layer loading times became approximately 20 times faster than in the previous developed systems. Since Geoserver is now serving static pre-calculated data, the CPU is barely used for this. This way the layer loading is almost instantaneous, giving the user a quick-access to various levels of zoom of several forecasting products, which may be of crucial importance in emergency response.

The front-end consists of a single-page responsive web application which allows users to visualize all the products served by the back-end, in an intuitive interface available for multiple devices. With the goal of ubiquitousness in mind, less-intensive technologies were chosen: (1) HTML5 and CSS3, as the building base of all web-applications, (2) AngularJS, a javascript framework that offers dynamic templating and two-way databinding, (3) Google Polymer, a Google design specification implementation library, and (4) OpenLayers, a library for handling geospatial data and mapping tools on the client-side. Although some technologies, like AngularJS and Google Polymer, are still in a beta stage, they are supported and maintained by Google and have a huge community contributing for their development (AngularJS has over 7 K commits on github at the time of writing, more than older and well-known javascript libraries such as JQuery with less than 6 K commits). Using these technologies appears thus a good choice for future developments, since they shape the way we build the web (www.polymer-project.org) and fully fulfill the user requirements.

As referred before, the front-end maintains communication with the back-end via a set of REST services made available by the back-end. Responses to these requests come in JSON, a lightweight data-interchange format easy to read/write by humans and to parse/create by machines.

The server-side information flow system has been planned and built to easily gather real-time data from the wireless sensor network, to parse the relevant information and store it in a persistent database system, to use the measurements to automatically compare them with forecasting model results, and to provide the forecast results to the end-user. A detailed description of this flow of information can be seen in Fig. 2. From top to bottom, this system building blocks are (1) a wireless sensor network (WSN), comprised of several sensor nodes; (2) a data gathering server, which communicates with the WSN gathering and parsing real-time data; (3) prediction model instance and corresponding redundancy

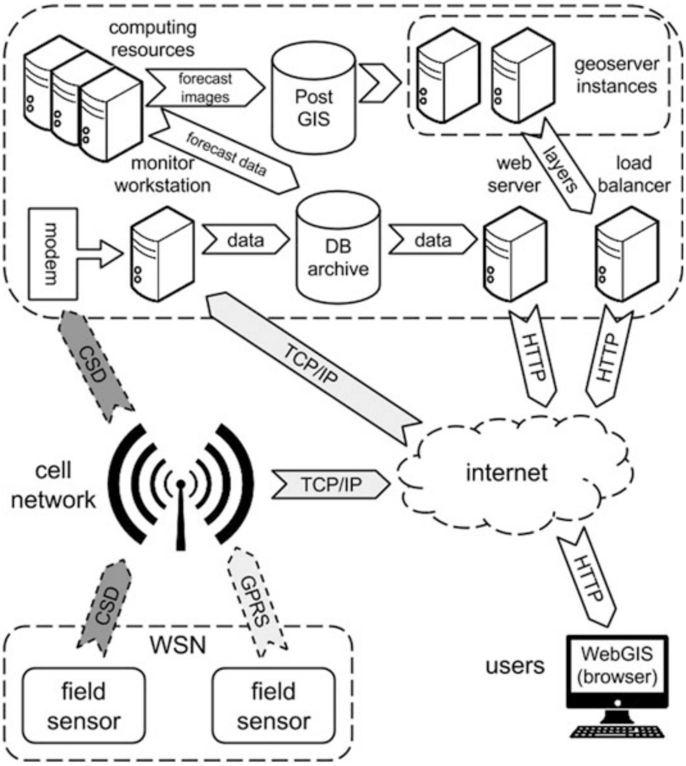


Fig. 2 Network information data flow

instance (to guarantee a fallback), that produce forecasting results; (4) several instances of Geoserver (again to guarantee a fallback, but also to guarantee data availability for a great amount of web-requests), that start by consuming forecast results in the form of visual imagery which later renders to the end-user in the form of WMS layers. All the instances of Geoserver are managed by a load-balancer that decides which instance should handle the client requests. The WSN consists of several real-time station nodes scattered in the domain of interest, which record water level data and transmit it to a central server. This central server is basically a set of scripts that trigger the transmission via either GPRS into an FTP server or Circuit Switched Data (CSD) directly into the file system. After the transmission is performed successfully, data is handled by the central-server to parse and store it, and to create input files for the prediction models. On their side, prediction models produce water level and wave propagation forecast results for the next 48 h which are then consumed by the geographical web server. Geoserver accepts both raster images (geotiff) and vector files (shapefiles), but can also produce images from Postgis database data. These images are georeferenced and then presented in the user interface on a layer map.

4 Application to the Case Study

The flexibility and usefulness of the platform is illustrated here, applied to the MOLINES case study, mainly through examples of the functionalities available on the interface. For this project, the requirements were the following: (1) the platform would account for different user roles, providing differentiated access to dedicated products; (2) the platform should be able to host georeferenced products from the static risk analysis (hazard, vulnerability) and the dynamic real time forecasts; (3) the platform should be agile, providing fast access to the alerts and their products; (4) the platform should be prepared to incorporate and show in a georeferenced way the information uploaded by the civil protection agents in the field during emergencies; and (5) to be able to assess products at the different spatial scales of interest (from the Atlantic ocean to the Seixal bay) and to infer which phenomena, generated at the large scale, is responsible for an observed inundation at local scale.

This application user interface is composed of a top header with title, a sidebar for displaying the various links to the main functionalities offered and a detailed content area. The sidebar hides when opening the application in smaller devices (smartphones or tablets), and is accessible by a button on the top header, as seen in Fig. 3. This allows the main content to be shown in full screen, taking advantage of all available space on the device screen.

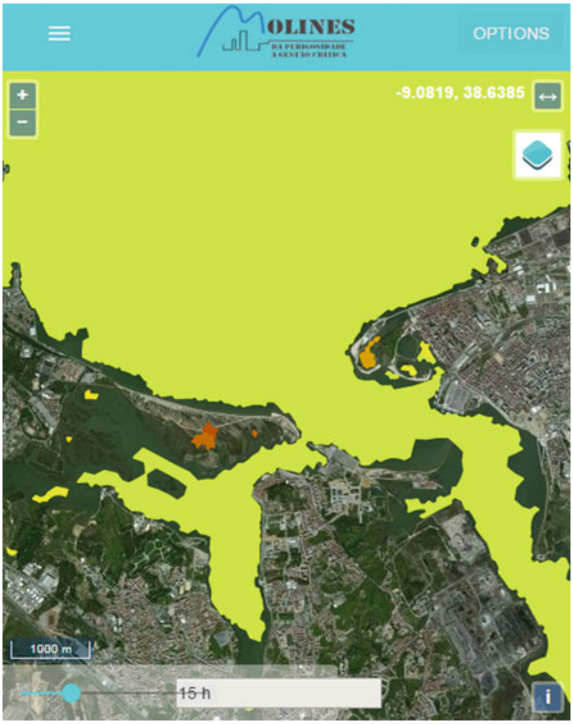
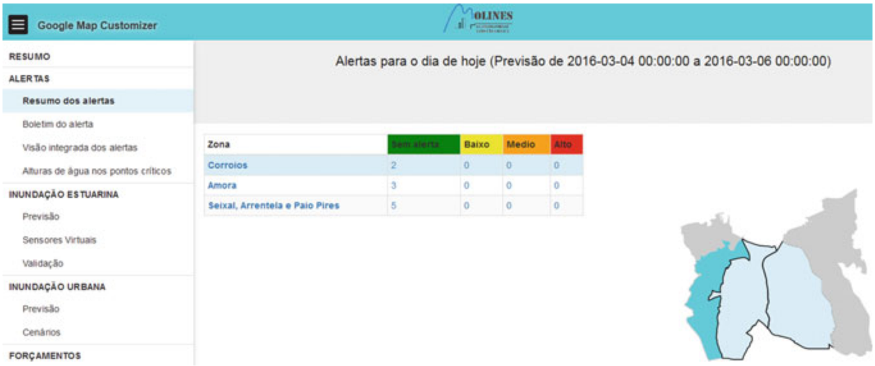


Fig. 3 Example of the interface adapted to a mobile device

Since this is a platform for support to detailed risk management, it must provide the users with quick and simple ways to access all the emergency events detected on the study zone, organized along different alert levels and at specific critical points, defined by the civil protection agents. This information is provided as a summary of the highest inundation events bound for the next 48 h, shown in a table, grouped by geographic zone and alert type. The geographical representation of the specific areas is shown in SVG form. This functionality can be seen in Fig. 4: a summary of study zones and inundation risk alerts triggered for each zone as well as their alert type. Also, when hovering over the zones on the table in Fig. 4, the corresponding zone on the map gets highlighted for a better identification of where this zone is.

Following the links on the referred table takes the user to a more detailed listing of the locations at risk. This detailed listing is structured as follows (and can be seen in Fig. 5): At the top of the page there's a map with all the critical points at risk marked, so the user knows where each point is located. Moreover, on this map the user can use the Street Map View from Google, allowing him to have an even better notion of the assets at stake at each location, in case the user isn't familiar with the zone. Below the map, there is a list with all the critical points marked above, and each item of the list contains: the name of the critical point, the maximum alert level for the current forecast, an alert bulletin with information on how to react to such inundation event and an alert summary bar for the next 48 h. An example of this bar can be seen on Fig. 6, and it shows the inundation alert level (Green for None, Yellow for Low, Orange for Medium and Red for High) for the next 48 h, split in slots of 15 min.

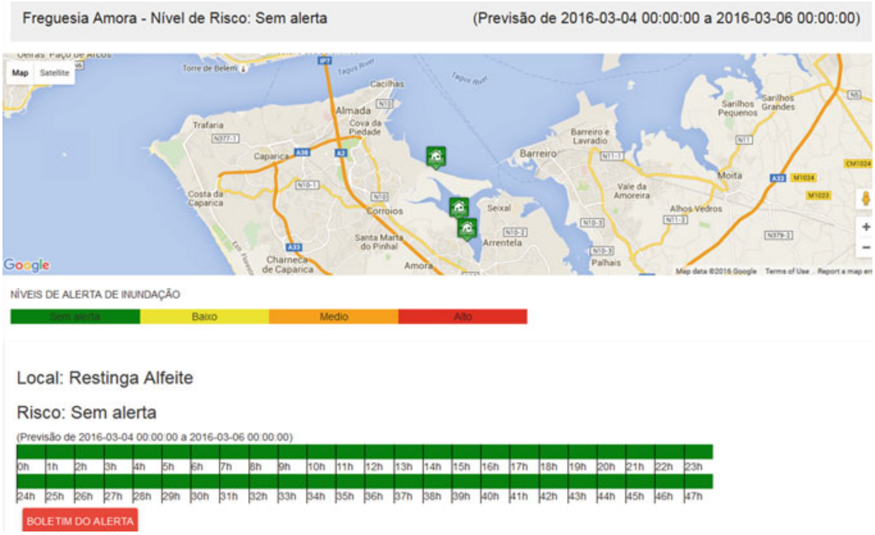
This information is very useful for the civil protection agents involved, since they can have a better notion of which critical points will be at flood risk and when will it happen. An integrated view of all critical points in a single page, with the corresponding risk level color bars, regardless of their zone, is also available on the platform (Fig. 7).



Alertas para o dia de hoje (Previsão de 2016-03-04 00:00:00 a 2016-03-06 00:00:00)

Zona	Sem alerta	Baixo	Medio	Alto
Corroios	2	0	0	0
Amora	3	0	0	0
Seixal, Arrentela e Paio Pires	5	0	0	0

Fig. 4 Summary of risk events



Freguesia Amora - Nível de Risco: Sem alerta (Previsão de 2016-03-04 00:00:00 a 2016-03-06 00:00:00)

NÍVEIS DE ALERTA DE INUNDAÇÃO

Sem alerta Baixo Medio Alto

Local: Restinga Alfeite

Risco: Sem alerta

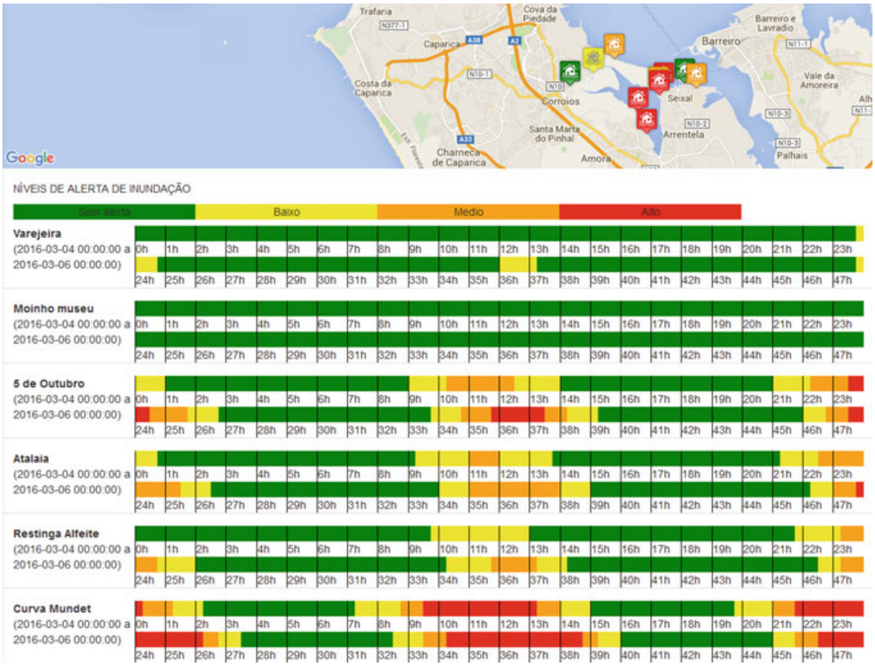
(Previsão de 2016-03-04 00:00:00 a 2016-03-06 00:00:00)

0h 1h 2h 3h 4h 5h 6h 7h 8h 9h 10h 11h 12h 13h 14h 15h 16h 17h 18h 19h 20h 21h 22h 23h

24h 25h 26h 27h 28h 29h 30h 31h 32h 33h 34h 35h 36h 37h 38h 39h 40h 41h 42h 43h 44h 45h 46h 47h

BOLETIM DO ALERTA

Fig. 5 Detailed location of areas at risk



NÍVEIS DE ALERTA DE INUNDAÇÃO

Sem alerta Baixo Medio Alto

Varejeira (2016-03-04 00:00:00 a 2016-03-06 00:00:00)
0h 1h 2h 3h 4h 5h 6h 7h 8h 9h 10h 11h 12h 13h 14h 15h 16h 17h 18h 19h 20h 21h 22h 23h
24h 25h 26h 27h 28h 29h 30h 31h 32h 33h 34h 35h 36h 37h 38h 39h 40h 41h 42h 43h 44h 45h 46h 47h

Moinho museu (2016-03-04 00:00:00 a 2016-03-06 00:00:00)
0h 1h 2h 3h 4h 5h 6h 7h 8h 9h 10h 11h 12h 13h 14h 15h 16h 17h 18h 19h 20h 21h 22h 23h
24h 25h 26h 27h 28h 29h 30h 31h 32h 33h 34h 35h 36h 37h 38h 39h 40h 41h 42h 43h 44h 45h 46h 47h

5 de Outubro (2016-03-04 00:00:00 a 2016-03-06 00:00:00)
0h 1h 2h 3h 4h 5h 6h 7h 8h 9h 10h 11h 12h 13h 14h 15h 16h 17h 18h 19h 20h 21h 22h 23h
24h 25h 26h 27h 28h 29h 30h 31h 32h 33h 34h 35h 36h 37h 38h 39h 40h 41h 42h 43h 44h 45h 46h 47h

Atalaia (2016-03-04 00:00:00 a 2016-03-06 00:00:00)
0h 1h 2h 3h 4h 5h 6h 7h 8h 9h 10h 11h 12h 13h 14h 15h 16h 17h 18h 19h 20h 21h 22h 23h
24h 25h 26h 27h 28h 29h 30h 31h 32h 33h 34h 35h 36h 37h 38h 39h 40h 41h 42h 43h 44h 45h 46h 47h

Restinga Alfeite (2016-03-04 00:00:00 a 2016-03-06 00:00:00)
0h 1h 2h 3h 4h 5h 6h 7h 8h 9h 10h 11h 12h 13h 14h 15h 16h 17h 18h 19h 20h 21h 22h 23h
24h 25h 26h 27h 28h 29h 30h 31h 32h 33h 34h 35h 36h 37h 38h 39h 40h 41h 42h 43h 44h 45h 46h 47h

Curva Mundet (2016-03-04 00:00:00 a 2016-03-06 00:00:00)
0h 1h 2h 3h 4h 5h 6h 7h 8h 9h 10h 11h 12h 13h 14h 15h 16h 17h 18h 19h 20h 21h 22h 23h
24h 25h 26h 27h 28h 29h 30h 31h 32h 33h 34h 35h 36h 37h 38h 39h 40h 41h 42h 43h 44h 45h 46h 47h

Fig. 6 Overview of the alert system: time sequence of alert levels at the critical points

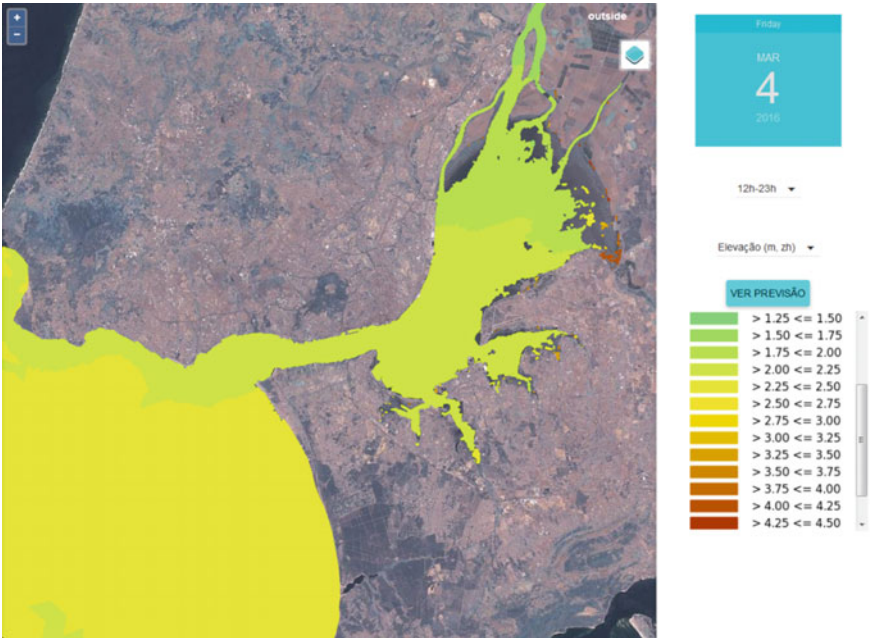


Fig. 7 Estuarine water level real time forecast

The alert information is produced and supported by a comprehensive forecast system, that predicts the circulation and waves from the Atlantic ocean to the area of interest (Fig. 8). This forecast system is based on a multi-scale implementation of several high resolution numerical models, accounting for tides, waves, storm surges and river flows [19]. Based on these results, specific water level (Fig. 7) and wave height products are available as GIS layers. These are in shapefile format, served through Geoserver and allow for several levels of zoom, as described in Sect. 3, for the region of interest. The user can also use the platform to analyze the generation of the phenomena, at regional scale, that lead to the alert level in the region of interest. Indeed, simpler map products are also available for regional circulation and wave predictions.

Other functionalities include a section to provide access to the real-time data gathered from WSN, which also allows an automatic comparison with model forecast results. This functionality allows the end-users to access data being measured at a point of interest but also to validate model predictions with real-time data. Indeed, this comparison is fundamental for the emergency agents and other decision-makers to provide them a measure of reliability on the model predictions (Fig. 7) and confidence on the actions to be promoted in the field.

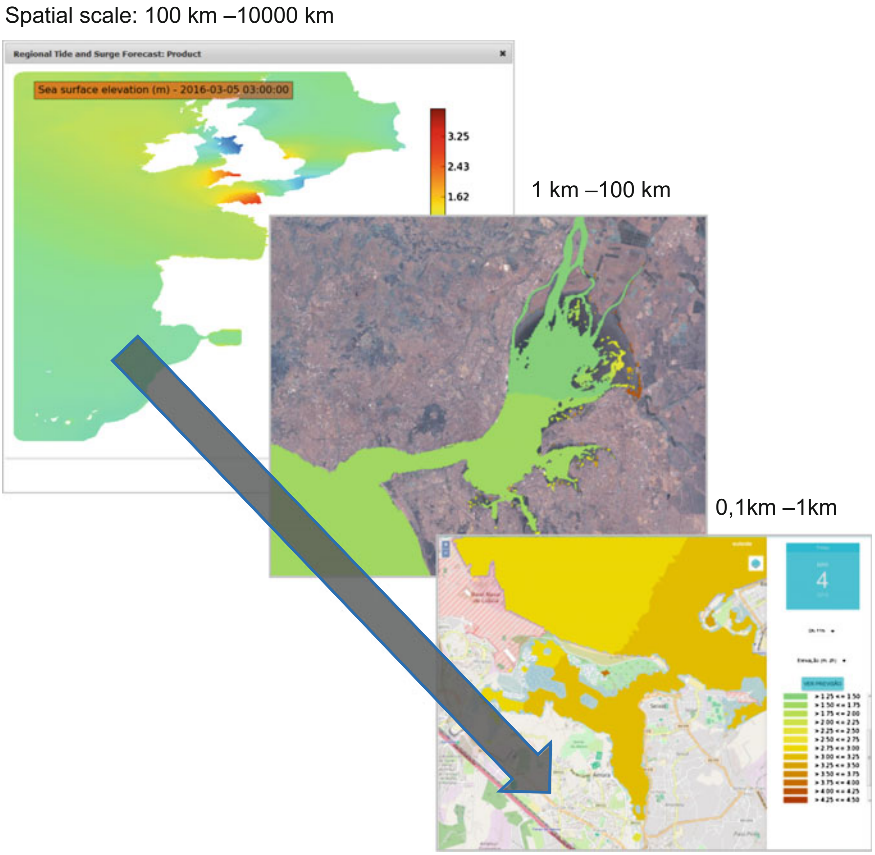


Fig. 8 From oceanic to estuary scale

5 Discussion and Future Work

Herein, an interactive, flexible and multiple user roles Web platform is presented, which takes advantage of novel technologies to provide fast access to all relevant online, intuitive and geographically referenced real-time data and model predictions for urban and estuarine floods.

Future work on the platform is planned to further improve its performance. For instance, one of the strategies includes using GeoJSON, an open standard format for encoding geographical information features using Javascript Object Notation, on the client-side to render georeferenced data layers on top of maps, instead of depending on Geoserver to serve those layers. This would put the workload on the client-side instead of the server-side, which would produce faster results overall.

We also aim at having these IT platforms integrated in the every day's workflow of end-users. Their operation requires considerable computational efforts that may not available in many decision-makers IT infrastructures. As LNEC's computational resources are limited, a strategy should be looked up to provide the best solution for end-users. To allow for stakeholders to run their own prediction model instances and operate these IT platforms, future work also includes the creation of prediction model deployments on the cloud. The evaluation of this solution for the modeling systems presented herein is currently on-going [20].

For the MOLINES application, the future work will be concentrated on the integration, in the interface, of the uploaded data provided by the agents in the field. Challenges are the automatic check on the reliability of this information and the way to integrate them in the interface in a simple and easy to probe manner. Other add-ons include also the issuing of the alert based on the model predictions.

Acknowledgements The authors thank Paula Freire and Ana Rilo (LNEC), Alexandre Tavares and Pedro Santos (CES), and Luis Sá (Portuguese Civil Protection Authority), for helping with the platform requirements. They also thank the SCHISM community for making their model code available at http://ccrm.vims.edu/schism/. This work was funded by the Portuguese National Foundation for Science and Technology through project MOLINES (PTDC/AAG-MAA/2811/2012).

References

1. Gomes, J.L., Jesus, G., Rogeiro, J., Oliveira, A., Costa, R., Fortunato, A.: Molines—towards a responsive Web platform for flood forecasting and risk mitigation. In: Proceedings of the Federated Conference on Computer Science and Information Systems, vol. 5, pp. 1171–1176. ACSIS (2015). doi:10.15439/2015F265
2. Townend, I., Pethick, J.: Estuarine flooding and managed retreat. Society **360(**1796), 1477–1495 (2002). doi:10.1098/rsta.2002.1011
3. Ugarelli, R., Leitão, J.P., Almeida, M.C., Bruaset, S.: Overview of climate change effects which may impact the urban water cycle (PREPARED 2011.011 report). PREPARED: enabling change project (2011)
4. Jesus, G., Oliveira, A., Santos, M.A., Palha-Fernandes, J.: Development of a dam-break flood emergency system. In: Proceedings of the 7th ISCRAM Conference, p. 5
5. Pradhan, A.R., Laefer, D.F., Rasdorf, W.J.: Infrastructure management information system framework requirements for disasters. J. Comput. Civil Eng. **21**(2), 90–101. doi:10.1061/(ASCE)0887-3801
6. Carracedo, P., et al.: Improvement of pollutant drift forecast system applied to the Prestige oil spills in Galicia Coast (NW of Spain): development of an operational system. Mar. Pollut. Bull. **53**(5–7), 350–360 (2006). doi:10.1016/j.marpolbul.2005.11.014
7. Rodrigues, M., et al.: Application of an estuarine and coastal nowcast-forecast information system to the Tagus estuary. In: Proceedings of the 6th SCACR—International Short Course/Conference on Applied Coastal Research, Lisboa, Portugal, pp. 10 (2013)
8. Oliveira, A., et al.: An interactive WebGIS observatory platform for enhanced support of coastal management. J. Coastal Res. (sp66), 507–512. ISSN 0749-0208
9. Deng, Z.Q. Namwamba, F., Zhang, Z.H.: Development of decision support system for managing and using recreational beaches. J. Hydroinform. **16**(2), 447–457 (2014). doi:10.2166/hydro.2013.185

10. Rinaldi, S., Peerenboom, J., Kelly, T.: Complexities in identifying, understanding, and analyzing critical infrastructure interdependencies. IEEE Control Syst. Mag. 11–25 (2001). doi:10.1109/37.969131
11. Corbacioglu, S., Kapucu, N.: Organizational learning and self-adaptation in dynamic disaster environments. Disasters **30**(2), 212–233 (2006). doi:10.1111/j.0361-3666.2006.00316.x
12. Beroggi, G.E.G., Wallace, W.A.: Operational risk management: a new paradigm for decision making. IEEE Trans. Syst. Man, Cybern. **24**, 1450–1457 (1994). doi:10.1109/21.310528
13. Beroggi, G.E.G., Wallace, W.A.: Multi-expert operational risk management. IEEE Trans. Syst. Man Cybern. Part C **30**, 32–44 (2000). doi:10.1109/5326.827452
14. Marincioni, F.: Information technologies and the sharing of disaster knowledge: the critical role of professional culture. Disasters **31**(4), 459–476 (2007). doi:10.1111/j.1467-7717.2007.01019.x
15. Kyng, M., Nielsen, E.T., Kristensen, M.: Challenges in designing interactive systems for emergency response. In: Proceedings of the 6th ACM Conference on Designing interactive Systems, pp. 301–310. ACM Press (2006). doi:10.1145/1142405.1142450
16. Herold, S., Sawada, M., Wellar, B.: Integrating geographic information systems, spatial databases and the internet: a framework for disaster management. In: Proceedings of the 98th Annual Canadian Institute of Geomatics Conference, pp. 13–15 (2005)
17. Fahland, D., Gläber, T.M., Quilitz, B., Weibleder, S., Leser, U.: HUODINI—flexible information integration for disaster management. In: Proceedings ISCRAM2007, pp. 255–138 (2007)
18. Kulkarni, A.T., et al.: A web GIS based integrated flood assessment modeling tool for coastal urban watersheds. Comput. Geosci. **64**, 7–14 (2014). doi:10.1016/j.cageo.2013.11
19. Fortunato, A.B., Tavares da Costa, R., Rogeiro, J., Gomes, J., Oliveira, A., Li, K., Freire, P., Rilo, A., Mendes, A., Rodrigues, M.: Desenvolvimento de um sistema operacional de previsão de temporais na costa portuguesa. In: Proc. VIII Congresso sobre Planeamento e Gestão das Zonas Costeiras dos Países de Expressão Portuguesa, 15p. (2015)
20. Rogeiro, J., Azevedo, A., Rodrigues, M., Oliveira, A.: Running high resolution coastal forecasts: moving from grid to cloud resources. In: Kruis, J., Tsompanakis, Y., Topping, B.H.V. (eds.) Proceedings of the Fifteenth International Conference on Civil, Structural and Environmental Engineering Computing 2015. doi:10.4203/ccp.108.218

Part III
Network Applications

A Distributed Active Vibration Control System Based on the Wireless Sensor Network for Automotive Applications

M. Zielinski, F. Mieyeville, D. Navarro and O. Bareille

Abstract This paper presents a new approach of an adaptive vibration control system for automotive applications. We assume that a porting of a centralised system in a distributed system can improve its effectiveness. We present a wireless sensor network (WSN) for vibrations damping. These autonomous sensors are able to measure the vibrations, damp the vibrations and to harvest energy from vibrations by using a single piezoelectric element. We present the simulations and the measurements results. The new approach of distributed active vibration control system based on the wireless sensor network is presented. The designed distributed wireless network node reduces the vibrations of the plate with the efficiency up to 9.4 dB.

1 Introduction

This work is an extended version of paper "A low power Wireless Sensor Node with Vibration Sensing and Energy Harvesting capability" which was presented at iNetSApp, FedCSIS 2014 [1].

The control of vibration and noise is essential in the design process of an automotive industry. From several years, new concept cars equipped with high technology systems used to improve passenger comfort are available. There are several approaches of systems used to reduce the vibrations. The standard passive solutions take into account the application of viscoelastic materials or the modification of the mechanical structures. The second possibility is to use the active vibration control (AVC) systems. These active solutions have some complex

M. Zielinski (✉) · F. Mieyeville · D. Navarro
Institut des Nanotechnologies de Lyon, Ecole Centrale de Lyon,
Université de Lyon, 69134 Ecully, France
e-mail: mateusz.zielinski@ec-lyon.fr

O. Bareille
Laboratoire de Tribologie et Dynamique des Systèmes,
Ecole Centrale de Lyon, Université de Lyon, 69134 Ecully, France

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_14

structures: a central processing unit, sensors, amplifiers and actuators. In the active approach the smart materials are attached directly to mechanical structures to provide the active control of vibration and noise.

The passive methods increase the weight of the cars. They have a major influence on the energy and fuel consumption [2]. The use of the active vibration control can reduce the weight of conventional passive systems, helping to push towards lighter and more fuel efficient vehicles. Over the last years the active vibration control has been widely studied by researchers [3].

The following section provides the short state of the art of the existing AVC systems for the automobile applications. Then our approach of the AVC system based on the WSN is presented and compared to the existing centralized active methods. The results contain: description and measurements of the designed mechanical system, presentation of designed WSN node. The feasibility of the wireless nodes to provide the vibration damping and sensing is proven by the results.

2 Active Vibration Control Systems: State of the Art

In the literature, we can find several solutions applied to reduce vibration and noise in car bodies. An active noise system using the feed-forward methods for tonal engine noise control was proposed in 1988 [2]. This proposed system was composed of microphones and loudspeakers placed in the passenger cabin. The maximum reduction of about 10 dB was recorded at a frequency of 100 Hz. The overall improvement is noted as a reduction of 4 dB.

Shi-Hwan et al. [4] presented an active control system of road booming noise. This system is composed of four reference sensors, two error sensors and two control actuators. In the presented case study, a car is moving at the speed of 60 km/h. The road characteristics are examined and the low frequencies are found as dominant (around 250 Hz). The authors achieved a 5–6 dB reduction of road booming noise at the vicinity of the error microphones. This work showed also the computational power limits of the various algorithms.

Fuller and von Flotow [5] showed the active vibration control system with an active absorber. A test vehicle was equipped with an inertial-mass shaker and a high efficiency calculation unit (dSpace MicroBox). A significant reduction up to 37 dB is achieved only for the very low frequency (up to 50 Hz). However, the disadvantage of the proposed solution is high adaptation time and the tight frequency range due to the usage of the Filtered X Least Mean Squares (FxLMS) algorithm.

The improvement of the smart materials leads to the new, piezoelectric solutions [6]. N. Alt et al. determined the oil pan as the most important source of the vibrations in the car engine and have proposed the active vibration control system based on the piezoelectric elements. In the experimental setup using the collocated control, the reduction of 12 dB is achieved. Respectively for adaptive feedback control, the results are 20 dB and for adaptive feed-forward control, 24 dB. The authors draw attention to the high costs of the system, which prevent the introduction to mass production. Additionally, the results show the generation of noise and vibration, for the higher frequencies (above 50 Hz).

The piezoelectric elements are also used in the active noise control system for the windshield of the car [7]. The authors reported the reduction of 7.45 dB (116 Hz) and 4.36 dB (145 Hz) using the State-Feedback control. The prototype system is composed of: three piezoelectric actuators, six accelerometers, one force sensor and one microphone. As a control unit the PC computer with the dSpice software product is used.

Tom Weyer and Hans Peter Monner are considering the vibrations of the car roof panel [8]. The authors note that the common resonant frequencies for the motor and the car body can cause vibration propagation. Therefore, the authors show the active vibration compensation using the piezoelectric actuators. The FxLMS algorithm is implemented in the dSpace 1005 Power PC computer. The attenuation of 20–30 dB is achieved for the frequency range around 42–62 Hz. The system is composed of six piezoelectric pairs (actuator and sensor).

In summary, the vibrations in the cars are primarily generated by the engine and by the interaction with the road surface. We can identify the disadvantages of the proposed active systems: high power computational units, large amount of cabling, long adaptation time, and lower efficiency for the frequencies above 50 Hz. In some cases, authors disclosed the increase of the noise or vibration in the mechanical systems. It was also noted that the high costs of the proposed systems can disallow mass production. However, the active vibration control system in the automobile application is desired by the market and a great effort has been made to improve the existing methods and solutions.

The aim of this work is to propose the new distributed approach of the active vibration control system, which can reduce the amount of the consumed energy with the vibration damping efficiency comparable to the existing wired solutions.

3 Integration of the WSN for Active Vibration Control

3.1 Global Structure (New Approach)

Figure 1 presents our new approach of the global system structure for distributed AVC for automotive applications. Wireless nodes are implemented within the body of the car. Nodes are organized in the star topology towards the collecting node.

Each node in the system is powered from the harvested energy from the vibrations and provides mechanical damping. Nodes measure the value of the vibrations and if necessary, send data to the collecting node. If there are no vibrations in the system, the nodes do not have energy to work but there is no need to cancel the vibrations.

When vibrations occur, the WSN nodes are initiated and provide the first level of the mechanical damping. This paper concerns a low power wireless sensor network with vibration sensing, vibration damping and energy harvesting capability.

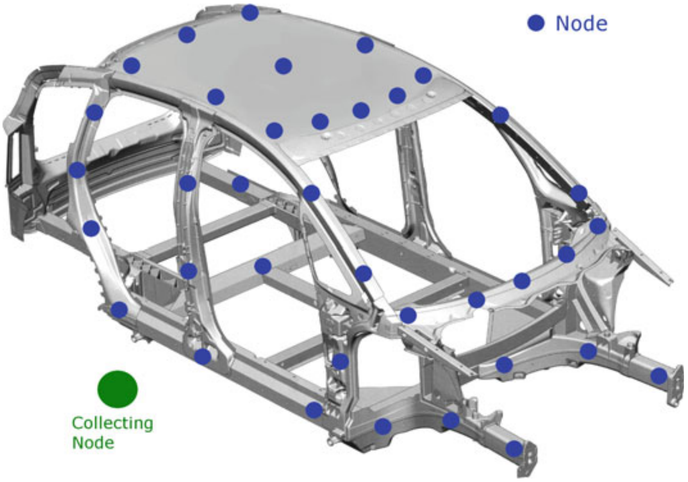


Fig. 1 Global system structure for distributed AVC for automotive application

3.2 Distributed AVC System Versus Existing Centralised AVC Systems

WSNs have rather low transmission rates. Due to delays, an implementation of the efficient real-time system, necessary to provide data processing for centralised AVC, is not possible [9].

In spite of this, WSN could be used to provide active control. A distributed approach used in place of the centralised one can be a solution. In our approach, intelligent nodes provide local action which reduces the amount of the information to transfer, compared to the centralised approach.

The distributed autonomous nodes with sensing feature coupled to semi-active vibration dissipation are the solution proposed in this paper.

Replacing a big centralised (wired) system with low power nodes can improve AVC in the scope of functionality, energy consumption, maintenance, and production costs.

4 WSN Node Design

Figure 2 presents the schema block for proposed WSN node. We distinguish three parallel circuits: energy harvesting, vibration damping, and vibrations sensing. All of them are connected to the one piezoelectric patch transducer. Harvested energy is kept in the storage and used to supply the microprocessor and wireless transmission unit. The following paragraphs describe the design of the proposed WSN node.

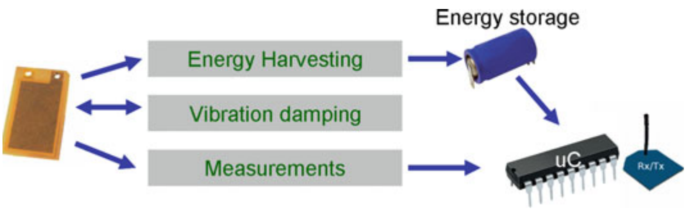


Fig. 2 WSN node schema block

4.1 Impedance Adaptation of the WSN Node

Piezoelectric effect can be considered as a bidirectional energy conversion. A mechanical strain on the piezoelectric element generates the electrical charge and respectively the electrical charge over the piezoelectric element generates the mechanical strain. To understand electrical properties of the piezoelectric patch transducer several measurements have been done. Figure 3 presents achieved results.

The output current and voltage values are measured over the piezoelectric patch transducer in function of the resistive load for constant frequency (Fig. 3). We can notice that the piezoelectric element is a real current source and for the optimal resistive load provides the maximal power. It clearly shows that the energy harvesting circuit must be designed in accordance with the electrical properties of the piezoelectric patch to receive the big electrical current value (optimal resistive load value).

According to Fig. 3 we can observe also that the high value of the resistive load reduces the amount of the energy received from the piezoelectric element. It proves the usage of the piezoelectric element with high resistive load for vibrations sensing (the small value of the leakage current is expected).

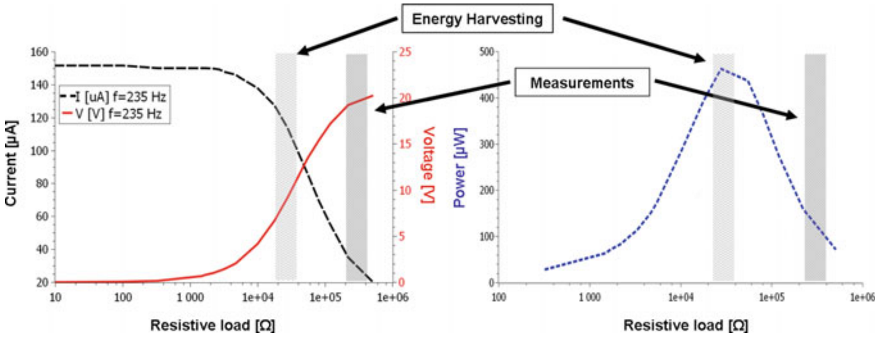


Fig. 3 Measured output electrical characteristics for the piezoelectric patch transducer

As it is presented the dynamic impedance adaptation and dynamic switching with disconnection capability is necessary to connect several parallel circuits with the only one piezoelectric element.

4.2 Vibration Sensing

Designed circuit for vibration sensing is presented on Fig. 4. It can be divided into several parts: impedance adaptation, gain and offset control, filter, and measurement.

Presented circuit is composed of the voltage divider and impedance adaptation circuit (R1 and R2) and the AD8138 low distortion differential analog-to-digital (ADC) driver from Analog Devices. The low pass filter is used to cut-off high frequencies over the output of the ADC driver (R7 with C1 and R8 with C2). The differential ADC driver provides also offset voltage (Pin 2 connected to the 1.65 V). Hence, the negative and positive values are measured. The low-power differential amplifier is supplied from single 3.3 V, which simplifies the supply circuit. An internal ADC of the microcontroller is used. This solution provides low energy consumption since there is no additional ADC to supply.

The voltage over the piezoelectric element corresponds in phase to the acceleration of the mechanical vibrations.

4.3 Vibration Damping and Energy Harvesting (Series SSHI Method)

The vibration damping and energy harvesting in the designed WSN node is based on the Series Synchronous Switching Harvesting with Inductor (SSHI) method [10]. The Series SSHI circuit is presented on the Fig. 5. It is a so-called

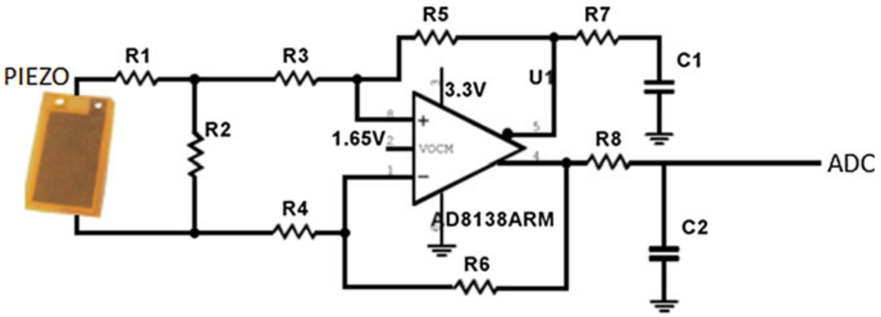


Fig. 4 Vibration measurement circuit

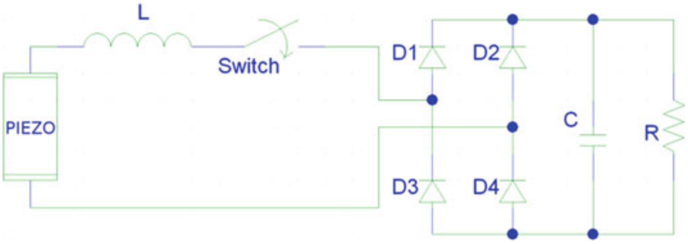


Fig. 5 The general "Series SSHI" circuit schema

"synchronised switch damping" (SSD) semi-passive method developed to address the problem of vibration damping. However, techniques based on SSD method provide efficient energy harvesting by increasing the energy flow between the piezoelectric element and the electrical load. Hence to it, the designed WSN node can provide the vibration damping and energy harvesting using the only one piezoelectric element.

The Series SSHI technique consists in a non-linear processing of the voltage delivered by the piezoelectric. In the series SSHI method, the inbuilt piezoelectric capacitance (PIEZO) and external inductance L creates the series resonant circuit. The switch keeps the circuit in the open-circuit. While the extrema of the mechanical displacement is detected (the maximal value of the electrical charge is generated in the piezoelectric element), the switch is closed for half of the electrical resonant period. It causes inversion of the piezoelectric element voltage because of the resonant circuit. In the same time, the electrical charge is stored in the storage capacitor C. Additionally we are using the bridge rectifier to provide full-wave rectification.

The designed series SSHI circuit contains two IRL630 NMOS transistors driven by the microcontroller (full-wave switch circuit with low current leakage and low short-circuit resistance). The usage of the logic-level transistor simplifies the circuit; the output of the microcontroller can be used to drive the electronic switch. The circuit works with the short switching times so we are using the Schottky diodes to increase their switching time.

4.4 Integration of a WSN Node (Simulations)

Designed analog circuits are simulated using the SPICE models in the NI Multisim software [11]. Figure 6 presents the general schema of the simulated WSN node.

The figure above contains the simple model of the microcontroller output pin used to control the Series SSHI switch. It is modelled as an ideal voltage source (V1). Then we can find the T1 transformer used as a galvanic isolation between the

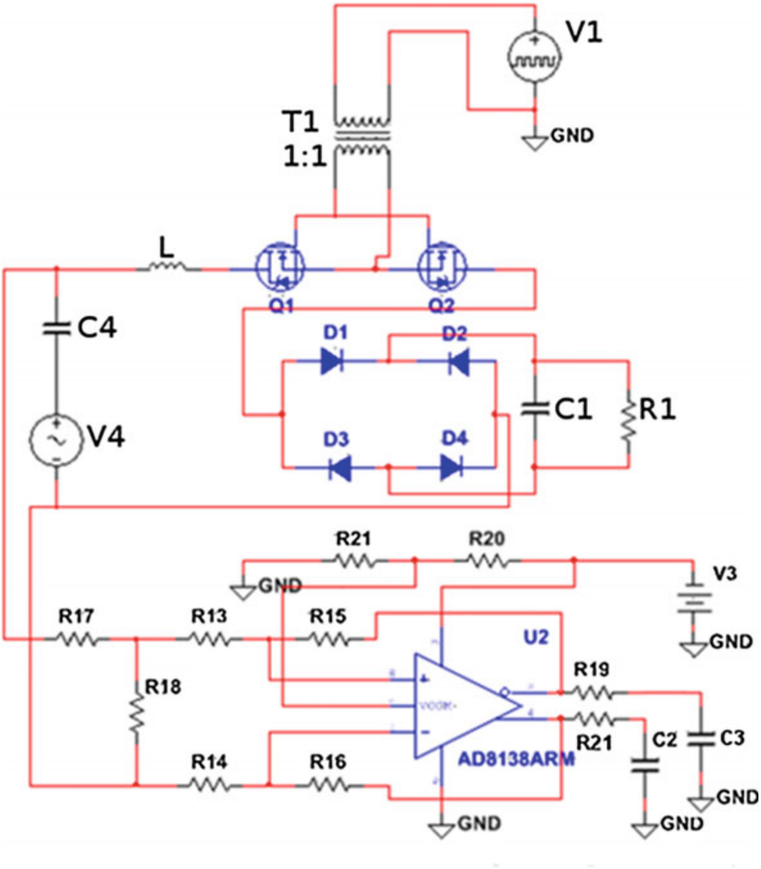


Fig. 6 Simulated analog part of the WSN node

microcontroller and the electronic switch to provide the switch circuit non-referenced to the ground.

A pulse transformer is, in principle, a simple, reliable and highly noise-immune method of providing isolated gate drive. It can be advised for applications were duty cycle is small [12]. The Series SSHI method is a system with low duty cycle because the period of the mechanical displacement is much longer than period of the electrical circuit. It makes the pulse transformer a promising solution for our design.

The piezoelectric element is simulated by the simplified electrical model composed of the voltage source and the capacity (V4 and C4). Then the vibration sensing circuit and the Series SSHI circuit are connected in parallel with the model of the piezoelectric element.

The series SSHI circuit contains two transistors and the inductor L. Additionally, the bridge (diodes: D1, D2, D3 and D4) is used to transform voltage from AC to

DC. Finally, the load and the storage capacitor are modelled by the resistor R1 and the capacitor C1.

4.5 Integration of a WSN Node (Prototyping)

According to the presented system approach and the specifications, the WSN node for the AVC system has been designed. The node is composed of the Microchip products: PIC16LF88 microcontroller and MRF24J40 radio transmitter. Moreover, the designed and already presented circuits (vibration sensing and series SSHI method for energy harvesting and vibration damping) are implemented according to the simulated schemas.

Figure 7 shows the photo of the prototyped device used in the experiment. The device dimensions (50.8 mm × 68.6 mm) are almost the same as the dimensions of the piezoelectric patch.

The microcontroller has been chosen because of its low energy consumption. The 8-bits microcontroller with the internal 10-bits analog-to-digital converter (ADC) is sufficient for our application. Moreover, the usage of the internal ADC reduces the energy consumption in comparison to the external devices. The MRF24J40 radio transmitter supports the IEEE 802.15.4 communication standard. The WSN node measures the amplitude and the frequency of the vibrations and transfer data using the non-beacon transmission mode. The wireless communication applied in the node needs a very low amount of the energy to provide the data transfer [9].

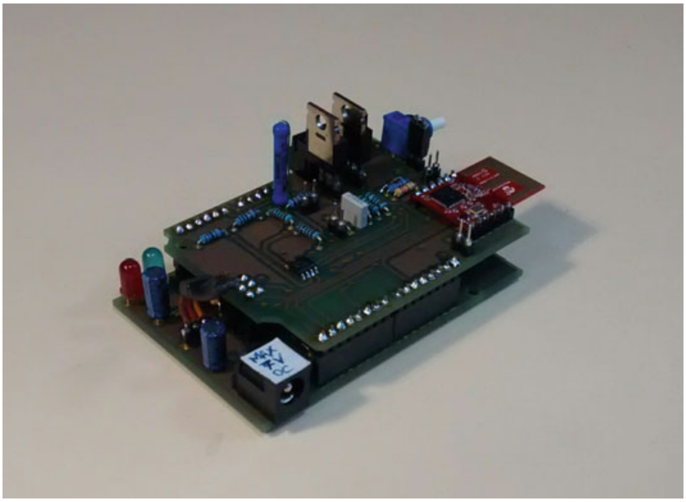


Fig. 7 Picture of the designed WSN node for AVC system

4.6 Validation of the WSN Node Design (Simulations Versus Measurements)

The WSN node is now designed, simulated and prototyped. The next step is to compare the results of the simulations and the measurements to verify the design and the choice of the used methods and components.

Figure 8 presents a block diagram of the designed WSN node and the three measurement points.

In the point 1 we look at the simulated and measured value of the signal conditioned in the vibrations sensing circuit. In the point 2 we track the value of the signal used to control the electronic switch of the series SSHI method (generated by the microcontroller on the basis of the measurements). In the point 3 we look at the output characteristic of the piezoelectric element (voltage and current). Figure 9 presents the comparison of the achieved, simulated and measured, results for the presented measurement points.

The simulated and measured signals in the point 1 are firstly the sinusoidal waves. During this period of the time, the Series SSHI method is inactive; the signal received by the microcontroller corresponds to the mechanical deformations.

The single vibration period is needed to measure the frequency and the amplitude of the vibrations by the WSN node. Then, extrema of the mechanical strain is detected. In this moment the microcontroller activates the Series SSHI method by generating the control signal (see measurements in point 2). This signal is used to control the electronic switch of the Series SSHI method.

In the point 3, we observe the current and the voltage values over the piezoelectric element. We can notice that the current flows only when the electronic switch is closed. We can also notice the voltage inversion caused by the resonance circuit created by the internal capacity of the piezoelectric element and the external inductance L.

The simulated and measured results are in good correlation. However, the ideal voltage source used in the simulations does not take into account the inductive and

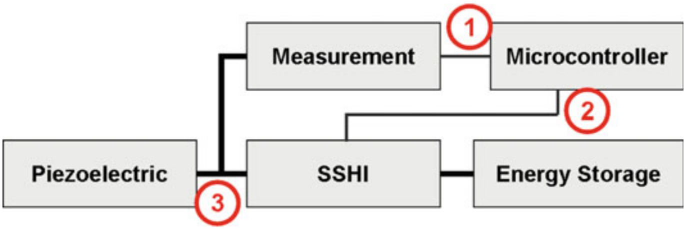


Fig. 8 Block diagram of the designed WSN node with the marked measurements points

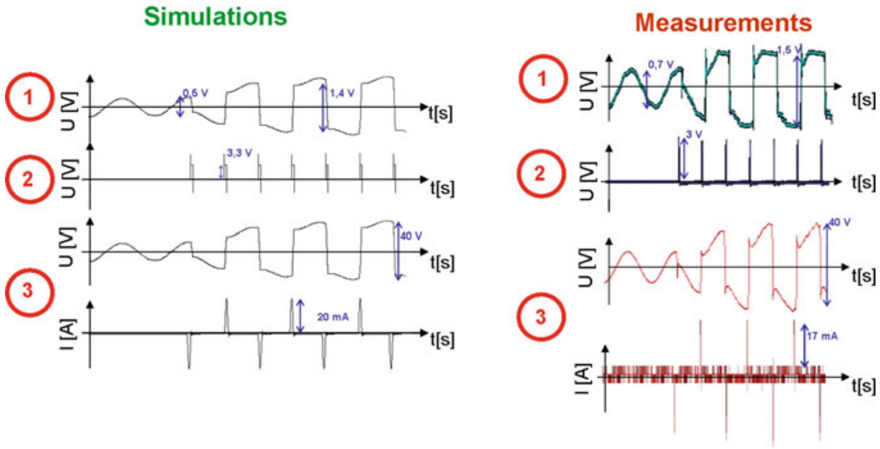


Fig. 9 The comparison of the simulated and measured signals

capacitive characteristics of the real piezoelectric element. Due to it, the differences are visible; they are mainly caused by the simplified model of the piezoelectric element or by the measurement noise.

Figure 10 presents the theoretical voltage and current values over the piezoelectric element for the Series SSHI method. We can compare it with the results from the simulations and measurements. It proves the design of the WSN node.

The designed WSN node provides the vibration sensing, mechanical damping and energy harvesting using the only one piezoelectric element. The node is independent and autonomous; it does not use any additional control signal. Moreover, using the wireless communication the node is able to send the measured values (frequency and amplitude of the vibrations). Since the WSN node for the AVC system is realised we can evaluate its feasibility using the mechanical structure as an experimental setup.

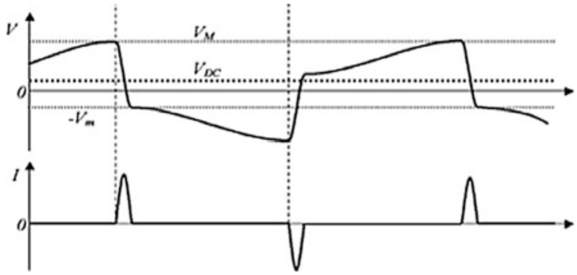


Fig. 10 Theoretical signals for the Series SSHI method (voltage and current) [10]

5 Experimental Setup

5.1 Specification for the AVC System in the Automobile Application

According to the state of the art, the vibrations in the body of car are the low frequency signals (up to 300 Hz) [13]. They are generated, among others, by the engine or by the interaction between the wheels and the road surface.

It is possible to identify the common vibration modes for different parts of the car body. It affects the propagation of vibrations. Therefore, the AVC system distributed on the surface of the car can be an interesting solution.

Nowadays, the car bodies are mostly made of the steel elements. Engineers are looking for lighter replacements [14, 15]. The most interesting material is aluminium. Despite the disadvantages related to its weld-ability and form-ability, aluminium is more popular than magnesium or polymer composites. Therefore, aluminium elements are used for our experiments.

The piezoelectric elements offer advantages such as: the high actuator force, the fast answer regarding changes, the bi-directionality of the piezoelectric phenomenon. It makes them more interesting than other damping solutions (viscoelastic, electrostatic, electromagnetic etc.). The inorganic piezoelectric elements are commonly used in the sensors and energy harvesting systems [16]. Hence, the P-876.A15 PICeramic actuator has been chosen. It is an elastic transducer which can also be applied to curved surfaces (dimensions: 61 mm × 35 mm × 0.8 mm). It is made of a modified lead zirconium titanate (PZT) material, optimised for actuator applications [17].

5.2 Mechanical System Used in the Experiments

Figure 11a presents the mechanical system designed and implemented in accordance with the established specifications. It is composed of the aluminium frame (profiles side length 0.045 m). The electrical vibrator attached to the frame using four nylon strings (diameter 0.01 m). The aluminium square panel (side length 0.332 m, thickness 0.001 m) attached to the frame by two nylon strings (diameter 0.001 m).

The square panel is considered as a simplified model of the body of the car. The electrical vibrator and panel are connected using the thin steel rod. The mechanical system has the following global dimensions: height 1.64 m, width 0.58 m and depth 0.49 m.

The dimensions and position of aluminium panel, piezoelectric elements and the excitation point are presented on Fig. 11b. The chosen piezoelectric elements are positioned on the panel surface in order to verify local, as well as, global influence of the designed AVC system. The excitation force is measured using the quartz

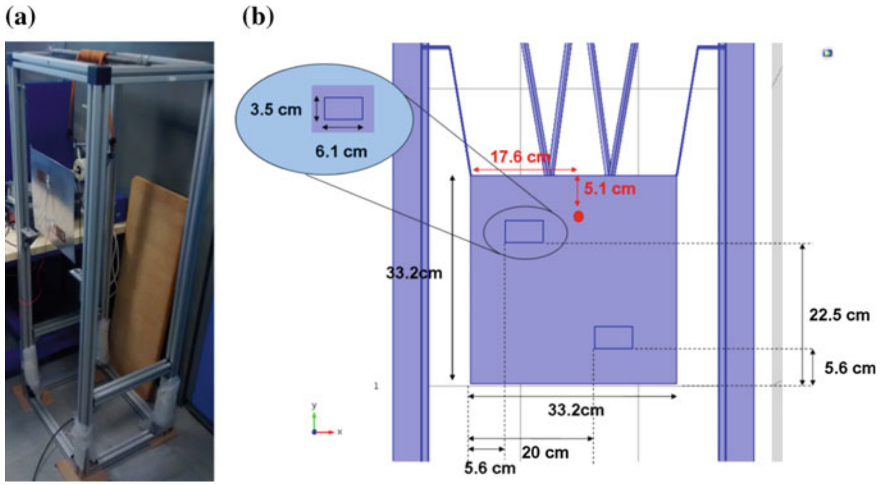


Fig. 11 **a** The mechanical system used in the experiments (photo). **b** Position and dimensions of the aluminium plate, excitation point and piezoelectric elements

force sensor manufactured by the PCB company (PCB 208C02). The force sensor is located between the electrical vibrator and the thin rod. The vibration velocity of the aluminium panel is measured using the laser doppler vibrometer: CLV-3D Compact 3D Laser Vibrometer produced by Polytec. Both measurements are realised in the z-axis according to the coordinate system presented on Fig. 11b. Designed system is excited with the signal generated by a function generator.

Figure 12 presents the point of the velocity measurements for the aluminium plate and its measured frequency response. We have also marked on this figure the frequencies: 65, 113, 130, 165 and 235 Hz, which correspond to the mechanical structure vibration modes.

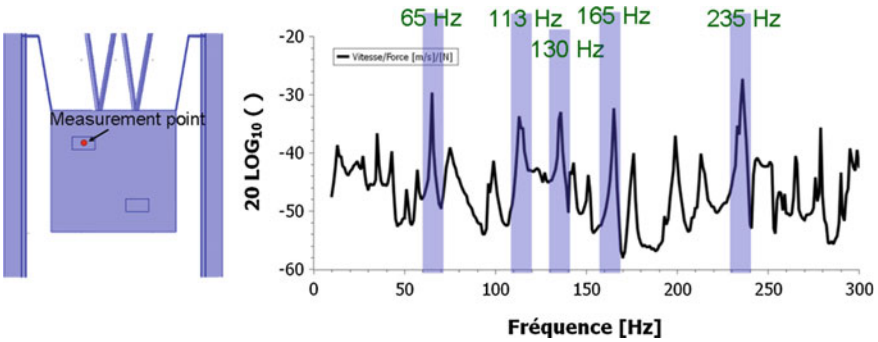


Fig. 12 Frequency response of the mechanical system

Since the mechanical system is presented and described, we can validate the feasibility of the designed WSN node. We will validate the vibrations damping capability for the marked vibration mode frequencies.

6 Validation of the WSN for the AVC

Firstly, we validate the single WSN node. In this case the only one WSN node connected to the one piezoelectric element is used to verify the local vibration damping capability. Then, the network of the two autonomous WSN nodes is used to verify the global vibration damping capability of the designed system.

6.1 Local Vibration Damping

The WSN node is connected to the left piezoelectric element mounted on the experimental aluminium plate (Fig. 13).

The panel is excited with a harmonic force with a constant magnitude. The amplitude of the vibration velocity is measured for two cases: when the designed WSN node is not active and when it is active. Afterwards, the velocity ratio is calculated (the results are presented in the decibel scale). Table 1 presents the results for the local vibration damping for different vibration frequencies.

According to the results presented in the Table 1, we notice the different efficiency of the vibration damping. Moreover, for the frequencies: 65 and 130 Hz, the vibration damping is not achieved.

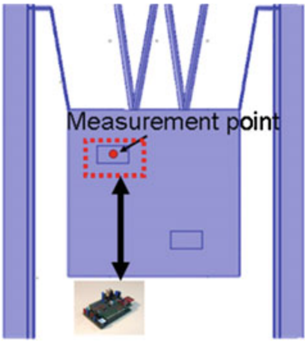


Fig. 13 The local vibration damping experimental configuration

Table 1 The local vibration damping efficiency for the different vibration frequencies

Frequency	65 Hz	113 Hz	130 Hz	165 Hz	235 Hz
Damping	0 dB	0.77 dB	0 dB	8.00 dB	9.34 dB

The FEM simulations are used to explain these differences. The experimental structure is simulated using the COMSOL 5.1 software. The deformations of the mechanical structure for the frequency of 65 Hz are presented on the Fig. 14.

The Fig. 14 shows the element piezoelectric which is not deformed. It explains why the designed WSN node is not able to damp the mechanical vibrations of 65 Hz. The same conclusions have been achieved for the frequency of 130 Hz.

The maximal vibration damping efficiency is achieved for the frequency of 235 Hz. The Fig. 15 presents the simulated deformations of the mechanical structure for this frequency.

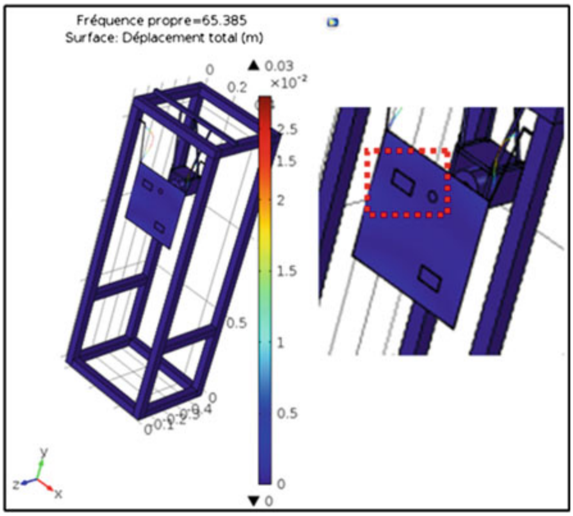


Fig. 14 Simulated deformations of the experimental setup for the frequency of 65 Hz

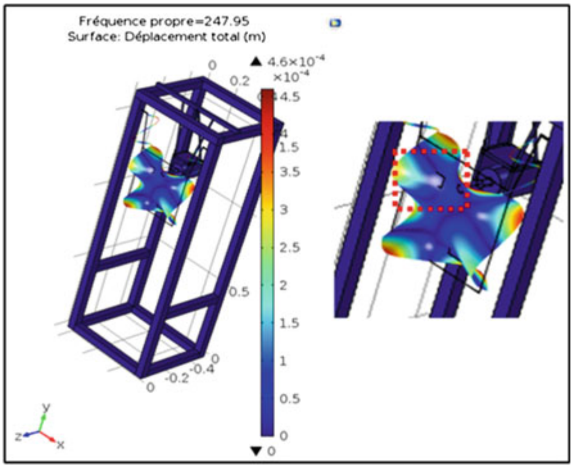


Fig. 15 Simulated deformations of the experimental setup for the frequency of 235 Hz

The Fig. 15 proves the deformation of the left piezoelectric element used to damp the mechanical vibrations. Moreover, for this frequency, the left piezoelectric element is placed next to the vibrations wave anti-node; thus, the maximum efficiency of damping is achieved.

The Fig. 16 presents the efficiency of the passive resistive method (the resistive load which is connected directly to the piezoelectric element) in comparison with the designed WSN node. The vibration efficiency is presented in decibel scale in function of load value. The mechanical excitations have the frequency of 235 Hz and are constant in the amplitude.

Figure 16 shows the difference in efficiency for the two considered methods. The passive method achieved the maximal efficiency of 4.31 dB for the load of 28 kΩ. For this load the WSN node is more efficient and has achieved the efficiency of 5.81 dB. The WSN node achieved the maximal efficiency of 9.34 dB for the load of 100 Ω. While, the passive method has achieved the efficiency of 3 dB for this load value. The presented results prove the local vibration damping capability of the designed WSN node.

6.2 Global Vibration Damping

The next step is to validate the global vibration damping with the designed distributed AVC system based on the WSN. In this case the two piezoelectric elements mounted on the aluminium plate are used. Both are connected to the designed WSN nodes.

Since two piezoelectric elements are used, both of them have to be deformed to provide the mechanical damping. Analysis of the experimental structure has shown the vibration frequency of 113 Hz as the most suitable for this experimental

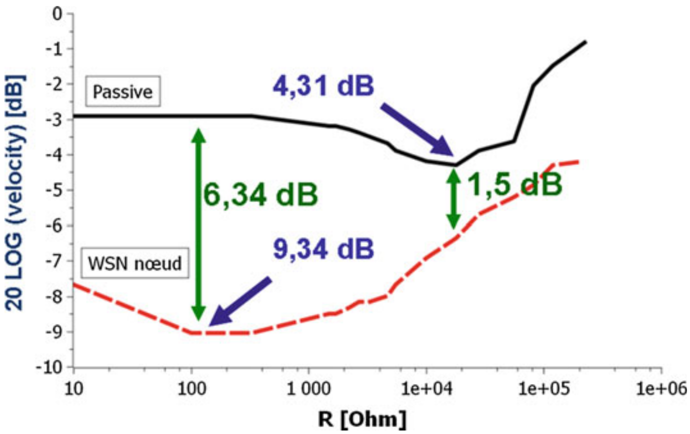


Fig. 16 The vibration damping efficiency: the comparison between the passive method and proposed WSN node

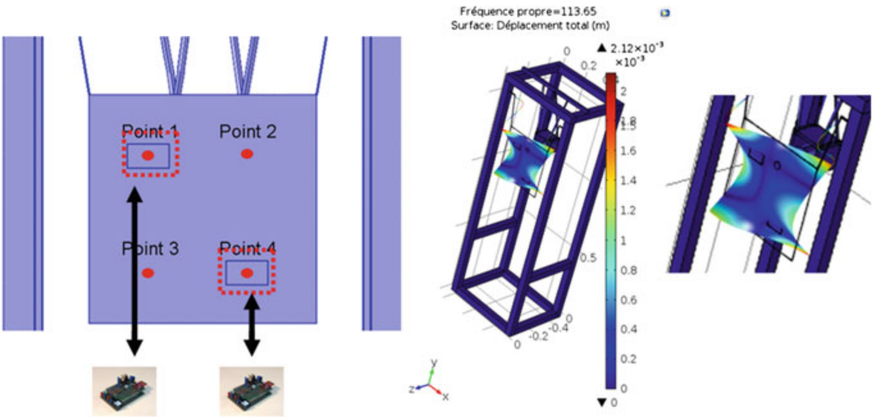


Fig. 17 The experimental setup for the global vibration damping and the simulated deformation

Table 2 The global vibration damping efficiency for the frequency of 113 Hz

	Point 1 (dB)	Point 2 (dB)	Point 3 (dB)	Point 4 (dB)
Left WSN node is active	0.77	0.75	0.89	1.53
Right WSN node is active	0.77	0.39	0.89	0.99
Both WSN nodes are active	1.62	1.52	2.41	2.10

setup. The Fig. 17 presents the experimental configuration and simulated mechanical deformation of the mechanical system for the chosen vibration frequency.

In this case, the vibration velocity is measured in the four points marked on the Fig. 17. Firstly, we measure the efficiency for two WSN nodes separately (local vibration damping). Then, the efficiency of the distributed system composed of two nodes is measured. The results are presented in Table 2.

The results prove the efficiency of the proposed distributed active vibration control. The designed WSN provides the additional damping action. We can notice the increase of the mechanical damping efficiency with the number of the nodes.

7 Conclusion

Centralised and wired systems for active vibration control are costly and use a large quantity of energy. A distributed solution based on an energy aware wireless sensor network has been proposed as a replacement for the centralised system. The autonomous WSN node needs to be designed to provide efficient wireless network for distributed active vibration control. In this paper the global approach and the system assumptions are established and used as input data for the design. The proposed design of the WSN node is in accord with the prescribed requirements.

Designed node provides: vibration sensing, shunting the piezoelectric element and wireless communication. Furthermore, the series SSHI technique, chosen for the design, provides damping of the mechanical vibrations and the energy harvesting capability.

Designed circuits for sensing vibrations and shunting the piezoelectric element are presented and described in details. The WSN node is modelled using the SPICE models. Achieved simulation results are consistent with the expected ones and validate the design. The WSN node prototype has been constructed. The simulation results are compared with the measurements. The measurement results correspond to the simulation results.

The distributed wireless AVC system is presented. The vibration damping is verified using the proposed experimental mechanical system. We have created two test scenarios. The first one is used to validate the local vibration damping capability. The measurements show the importance of the piezoelectric element position. The efficient vibration damping can be achieved only by using the actives elements placed next to the nodes of the vibration waves. The results prove also the necessity of the impedance adaptation. The second scenario is used to validate the global aspect of the designed distributed system. The two piezoelectric patches are used to damp the mechanical vibrations. The measurements prove the distributed approach. We have achieved the additional action provided by the network of the nodes.

The results confirm the application of the low power wireless nodes in the distributed AVC system. The achieved efficiency of 9.4 dB is comparable with the existing systems. Moreover, the scalability of the system is proved. The increase of the network will improve the global mechanical damping efficiency. Finally, results prove the usage of the autonomous wireless sensor network nodes in the vibration damping application.

The following step is the validation of the energy harvesting capability in the designed WSN in order to confirm the auto supply possibility. The designed and described mechanical system and the proposed distributed AVC system will be used to measure the value of the energy harvested from the vibrations by the implemented Series SSHI method. Then, the distributed vibration control algorithm will be proposed.

References

1. Zielinski, M., Mieyeville, F., Navarro, D., Bareille, O.: A low power wireless sensor node with vibration sensing and energy harvesting capability. In: iNetSApp, Federated Conference on Computer Science and Information Systems 2014, pp. 1065–1071. http://dx.doi.org/10.15439/978-83-60810-58-3
2. Elliott, S.J.: A review of Active Noise and Vibration Control in road vehicles. ISVR Tehcnical Memorandum No 981 (2008). http://eprints.soton.ac.uk/id/eprint/65371
3. Svaricek, F., et al.: Automotive Applications of Active Control, Croatia, pp. 380 (2010). http://cdn.intechopen.com/pdfs-wm/11899.pdf. ISBN 978-953-307-117-6

4. Shi-Hwan, O., Kim, H.-S., Park, Y.: Active control of road booming noise in automotive interiors. J. Acoust. Soc. Am. **10**(1121/1), 1420390 (2002)
5. Fuller, C.R., von Flotow, A.H.: Active control of sound and vibration. IEEE Control Syst. **10** (1109/37), 476383 (1995)
6. Alt, N., Lahey, H.-P., Nehl, J., Nussmann, C., Weiser, J.: Reduction of power train induced vehicle exterior noise by piezo-foil technology. In: InMar—Intelligent Materials for Active Noise Reduction, EU Project (2006). http://www.inmar.info
7. Misol, M., Algermissen, S., Monner, H.P.: Experimental study of an active window for silent and comfortable vehicle cabins, Chap. 36. In: Adaptive, Tolerant and Efficient Composite Structures, Research Topics in Aerospace, pp 439–447 (2013). https://dx.doi.org/10.1007/978-3-642-29190-6_36
8. Weyer, T., Monner, H.P.: PKW Innenlrmreduzierung durch aktive Beruhigung der durch die Motorhar-monischen erregten Dachblech-Schwingungen. Institut fr Faserverbundleichtbau und Adaptronik (2003). www.dlr.de/fa/Portaldata/17/Resources/dokumente/institut/2003/2003_01_pkw_weyer.pdf
9. Mieyeville, F., Ichchou, M., Scorletti, G., Navarro, D., Du, W.: Wireless sensor networks for active vibration control in automobile structures. Smart Mater. Struct. **21** (2012). http://dx.doi.org/10.1088/0964-1726/21/7/075009
10. Lefeuvre, E., Badel, A., Richard, C., Petit, L., Guyomar, D.: A comparison between several vibration-powered piezoelectric generators for standalone systems. Sens. Actuators A **126**, 405–416 (2006). http://dx.doi.org/10.1016/j.sna.2005.10.043
11. NI Multisim Component Evaluator White Paper, May 2012. http://www.ni.com/white-paper/9452/en/
12. Vishay Siliconix. Application Note AN-937; Gate Drive Characteristics and Requirements for HEXFET Power MOSFETs. Document Number: 91421 (2010). http://www.vishay.com/docs/91421/appnote9.pdf
13. Neri, I., et al.: A real vibration database for kinetic energy harvesting application. J. Intell. Mater. Syst. Struct. **23**(18), 2095–2101 (2012). doi:10.1177/1045389X12444488
14. Hall, J.N.: 50 years perspective of automotive engineering body materials and an analysis of the future. Great Designs in Steel (2008). http://www.steel.org/16
15. Mayer, H., Venier, F., Koglin, K.: The ASF Body of the Audi A8. FISITA International Federation of Automotive Engineering Societies. www.fisita.com/email/atz/EXTRA_ASFA8.pdf
16. Ramadan, K.S., Sameoto, D., Evoy, S.: A review of piezoelectric polymers as functional materials for electromechanical transducers. Smart Mater. Struct. (2014). doi:10.1088/0964-1726/23/3/033001
17. Whitepapers. DuraAct Patch Transducer P-876. Physik Instrumente (PI). GmbH und Co. (2015). http://piceramic.com

Improvements of Video Delay in OTT Live TV Service

Marek Dąbrowski, Robert Kołodyński, Wojciech Zieliński, Raoul Monnier and Patrick Marchal

Abstract The goal of this paper is to understand and quantify the end-to-end delay observed by users of Over The Top (OTT) Live TV services, using Adaptive Bit Rate (ABR) technology. The analysis and testbed measurements reveal to what extend the main architecture elements—encoder, packager, Content Delivery Network (CDN) and player—contribute to this overall delay. Some improvements at the architecture level and encoder implementation are proposed and partially evaluated by field experiments.

Keywords OTT · ABR · Live TV · Video · Delay

1 Introduction

Most commercially offered TV services over the Internet use so-called Adaptive Bit Rate (ABR) technology, also referred to "adaptive HTTP streaming" [1]. It assumes that the player is able to adapt to temporary network conditions by choosing among several profiles (versions of a stream encoded with a certain bitrate) which are available on a server. The continuous stream is divided into fragments of certain sizes ("chunks") and delivered to clients using standard HTTP protocol. The format of delivered video fragments and manifest file (an index which allows clients to reach specific stream version) is governed by a streaming protocol.

When the new generation of streaming formats started to be deployed in 2008–2009, the goal was to enable live video streaming over unmanaged networks with a good Quality of Experience (QoE). Such streaming formats as Microsoft

M. Dąbrowski (✉) · R. Kołodyński · W. Zieliński
Orange Polska, Centrum Badawczo-Rozwojowe, Warsaw, Poland
e-mail: marek.dabrowski@orange.com

R. Monnier · P. Marchal
Harmonic, Cesson Sévigné, France
e-mail: raoul.monnier@harmonicinc.com

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_15

Smooth Streaming (MSS), Apple HTTP Live Streaming (HLS), Adobe HTTP Dynamic Streaming (HDS), and later MPEG DASH, have been deployed worldwide so that, today, most premium TV channels can be watched flawlessly on any connected device. Latency (also called "delay" in this paper) however, is a sort of collateral victim of these successful OTT deployments: ABR was designed initially for Video on Demand (VoD) services and no special care was taken to optimize delay as it is not key for a good VoD experience. As one can easily check, there is quite a time delay between the presentation of a live channel over terrestrial, satellite or even IPTV platform, and the presentation of the same channel over an OTT platform. An overall latency in the 20–60 s range is currently observed in the field.

Up to now, latency has been considered as a minor issue compared to other tasks such as developing applications, deploying platforms, being online, working on business models, and growing audience. For many services such as VoD, Catch-up TV and for many linear channels, latency is actually not an issue at all. However there are a number of use cases where high latency becomes an issue as time goes. A few examples are listed here:

- Premium live event coverage (Premium sporting events)
- Second screen applications (interacting with the audience: E.g. voting)
- Betting channels
- Social viewing (comments are posted in real time while watching a show like "The Voice" or others).

In order to address delay sensitive services, it is necessary to understand where the delay in the OTT system is located and to identify the major delay contributors in the chain. Figure 1 depicts typical the architecture of OTT content delivery system and identifies major components which may contribute to e2e (end-to end) delay experienced by user.

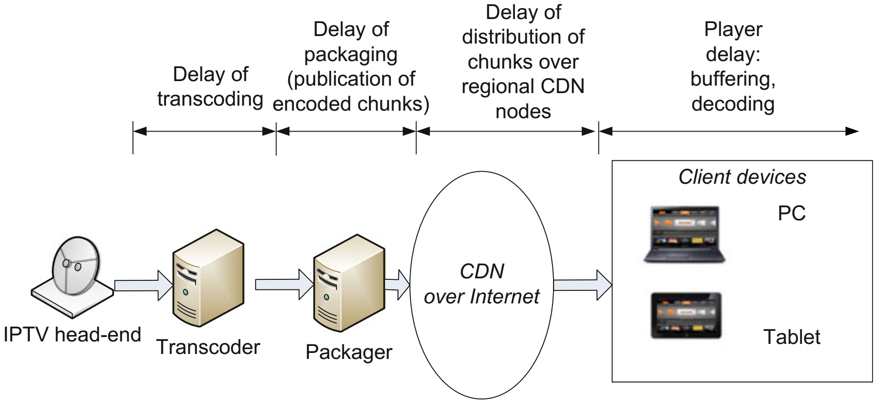

Delay of transcoding
Delay of packaging (publication of encoded chunks)
Delay of distribution of chunks over regional CDN nodes
Player delay: buffering, decoding
Client devices
PC
Tablet
IPTV head-end
Transcoder
Packager
CDN over Internet

Fig. 1 Delay components in live OTT video services

- IPTV head-end: Input content for OTT delivery chain is obtained from IPTV or satellite TV headend.
- Transcoder applies video compression, using several profiles appropriate for transmission over the Internet. H.264 is currently the most popular compression standard, with HEVC (H.265) considered as a future candidate.
- Packager applies the streaming format (MS Smooth Streaming, MPEG-DASH, HLS, …). It divides a continuous stream to chunks of fixed size, prepares the manifest file and publishes the files on HTTP server.
- CDN (Content Delivery Network) is used for the stream delivery to regional nodes in a wide area network. Since HTTP standard is used for message delivery, a typical Internet CDN is capable of supporting video streaming.
- Video player on the end-user device performs buffering, decoding and video display. Length of receiving buffer, which is a major source of end-to-end (e2e) delay, is the result of a compromise between short e2e latency (small buffer) and better resilience against packet-level jitter and losses that may occur in the transport network (long buffer).

1.1 Related Works

Impact of video delay on user perception of live TV service has been previously studied in paper [2]. In controlled experiment with real users, its authors have shown that video latency, even as small as several seconds, is noticeable and criticized by viewers of live TV sporting events. The study was however focused on more traditional TV distribution methods (broadcast, cable, satellite, IPTV multicast). Here we aim to study to what extent the problem will be visible in new OTT TV delivery systems. Similar approach has been taken by [3], where authors propose adaptive streaming protocol modifications to minimize live delay. The reported testbed measurements covered delay between packager and player, while on the contrary our paper focuses on end-to-end approach. The problem of OTT video delay has also been recognized in some industrial presentations, like [4, 5], which described commercial products that aim to minimize the delay. The experimental verification of these solutions is however not published.

1.2 Research Objectives

To progress beyond the state-of-the art in the area of live OTT streaming, this paper presents a theoretical as well as experimental evaluation of OTT architecture from the point of view of user-perceived video delay in live streaming service. The causes of delay in the streaming architecture are precisely identified. Following that, architectural improvements are proposed to reduce it, while maintaining the user experience. The work presented in this paper is an extended version of [6].

2 CELTIC NOTTS Project

The work presented in this paper has been carried out in the scope of EUREKA/CELTIC research project NOTTS (Next-Generation Over-The-Top Services) [7]. NOTTS joins a wide range of industry and academic partners from five European countries. Its goal is to investigate the technical problems experienced by service providers of OTT multimedia services, propose realistic solutions, and evaluate these solutions in testbeds and real networks. The project covers the whole ecosystem of media distribution, from scalable coding, media distribution architectures and workload models, to client monitoring and business model analysis.

3 Adaptive Streaming Protocol Analysis

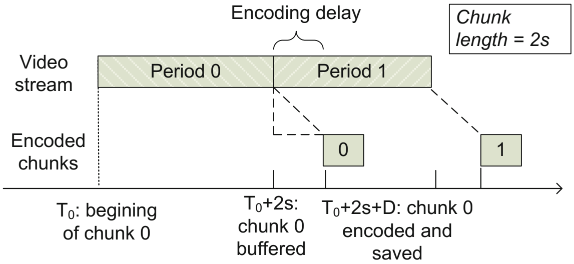
In this section, essential characteristics of adaptive streaming technology will be analyzed from the point of view of impact on e2e video delay.

3.1 Transcoder Behavior

The transcoder takes as input a continuous video stream, decodes it and encodes it again, producing video fragments suitable for further processing by the packager. The encoding standard used in tested scenarios is H.264, the same as the input stream. The format of output file is "fragmented MP4" (fmp4), containing the amount of video equal to the packager's chunk duration. Remark that the encoder and packager use the same configuration of chunk duration, and are thus not totally independent in their operation.

Illustrative explanation of encoder impact on video delay is presented in Fig. 2. Time of processing video chunk (fragment) inside the encoder process is denoted as D_{enc}.

Fig. 2 Illustration of encoder behavior


Encoding delay
Chunk length = 2s
Video stream
Period 0
Period 1
Encoded chunks
0
1
T_0: begining of chunk 0
T_0+2s: chunk 0 buffered
T_0+2s+D: chunk 0 encoded and saved

Remark that the configuration of the encoding profile may impact the value of this delay, as better quality profiles surely require more processing at the encoder.

3.2 Packager

The following two parameters are crucial for operating the packager (see Fig. 3):

- Chunk (fragment) length: amount of video (expressed in time units) that is encoded and packaged in a single HTTP message transmitted over the network. As an example, default value in Microsoft Smooth Streaming protocol [8] is 2 s.
- Number of lookahead fragments: succeeding fragments that have to be collected by the packager before releasing a given chunk. The default value (for MS Smooth Streaming) is 2.

As an example, let us assume that the *chunk length* is 2 s. The upper timeline in Fig. 3 shows a continuous video stream that is being served to the encoder. At the end of each period of 2 s, the encoder produces a chunk. Thus, the chunk numbered 0, containing video period starting at *T0* and lasting 2 s, is produced at time *T0 + 2 s* and at the same moment it is stored by the packager in its internal buffer for further processing. However, since the *lookahead* parameter is set to 2, the packager will wait for next 2 consecutive chunks, because some information about these chunks must be built in the header of chunk 0. Since chunk number 2 is available at time *T0 + 6 s*, only then, chunk number 0 can be published and made available for clients.

Remark that player may request live stream at an arbitrary moment T_p (see Fig. 3). The first (newest) chunk available at this random moment T_p, is chunk number 0, which is already aged *3*chunk length*, plus the duration of Δ, which is random. We may suppose that Δ is uniformly distributed between 0 and *chunk length*, with average value *chunk length/2*.

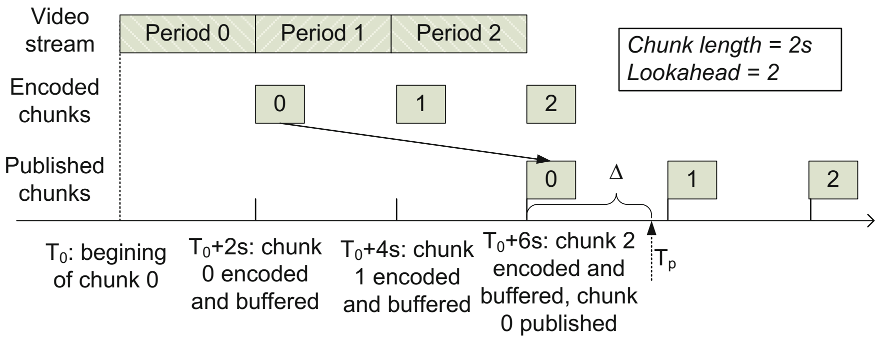


Fig. 3 Illustration of packager behavior

Thus, on average, the packager introduces a delay equal to (l is the *lookahead*, and t_f is the *chunk length*):

$$D_{pack} = (l + 1) \times t_f + \frac{t_f}{2} \qquad (1)$$

3.3 Player

Video player in the terminal is a major delay contributor to the e2e delay budget. MS Smooth Streaming protocol introduces the following three parameters which have a significant impact on the player behavior when it starts receiving a live video stream:

- Buffer: size of receiver buffer (number of seconds of stored video). Default value is 5 s.
- Backoff: when the player requests a live stream, it actually does not reach for the recent (current) video chunk, but rather for content that is delayed by a sum of backoff and offset parameters. Default value is 6 s.
- Offset: together with backoff time, the value of this parameter determines playback delay in relation to actual "live" position. Default value is 7 s.

When the player requests to receive a live video stream, it downloads first a manifest file, which describes the technical parameters which are necessary for the player to decode the stream and advertises the chunks that are available for download on the server. Timestamp of the latest (newest) chunk advertised by Manifest will be denoted as t_0.

However, the player does not normally get the chunk t_0. First, it goes back in time by the value of backoff plus offset. The sum of backoff and offset determines the timestamp of a chunk, from which the player starts downloading video fragments to fill its buffer (t_{start}). Now, the player immediately requests for next chunks, until it fills its buffer or reaches the limit determined by the offset value (player may not ask for chunks newer than "t_0 − *backoff*"). We should now distinguish two situations: *buffer* $\leq$ *offset* and *buffer* > *offset.*

Player Behavior When Buffer Is Smaller or Equal to Offset

The player immediately requests for a sufficient number of chunks to fill the entire buffer. It gets them as fast as network bandwidth can support. Now it is ready to start video playback, beginning with the oldest chunk stored in the buffer. The timestamp of the first chunk that will be displayed by player is:

$$t_{start} = t_0 - backoff - offset \qquad (2)$$

The video delay as seen by the user will thus be t0-tstart, that is:

$$D_{play1} = backoff + offset \tag{3}$$

Remark that chunk 0 does not really contain “live” position of video stream, due to delay introduced previously by operating the encoder and packager.

The behavior of the player is illustrated below in Fig. 4, which depicts chunks that are advertised when the player joins a live stream. The advertised window length is equal to 60 s. The player parameters assumed for the purpose of the example are: *buffer* = 6 s, *backoff* = 6 s, *offset* = 20 s.

The first (newest) chunk received by the player has a timestamp equal to $t_{0-2}6$ s. Since the buffer size is smaller than the offset, all the buffer may be filled immediately by retrieving 3 chunks (6 s), without waiting for any new chunks to be produced by the server. After retrieving enough chunks to fill the buffer to the required length, the player starts playing back, starting with the chunk t_{start}.

Player Behavior When Buffer Is Greater Than Offset

In the case when *buffer* > *offset*, the buffer cannot be immediately filled because the player is not allowed to fetch chunks that are newer than t_0 − *backoff*. So, it immediately (as fast as the network bandwidth can support it) fetches the amount of video chunks corresponding to the duration of an offset, and then waits for new chunks to arrive, in order to fill the remaining part of the buffer. After time *(buffer* − *offset),* the buffer is filled and the player can start playing back the video, beginning from the chunk with timestamp t_{start}. But t_{start} is now additionally delayed from t_0 by *(buffer* − *offset)* because the player had to wait that time to fill the buffer. So:

$$\begin{aligned} t_{start} &= t_0 - backoff - offset - (buffer - offset) \\ &= t_0 - backoff - buffer \end{aligned} \tag{4}$$

The video delay is equal to $t_0 - t_{start}$, that is:

$$D_{play2} = backoff + buffer \tag{5}$$

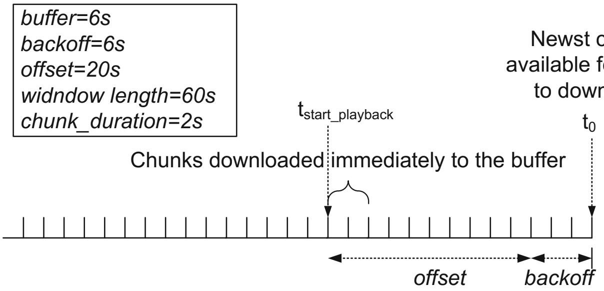


Fig. 4 Illustration of player behavior in the case *buffer* < *offset*

Described behavior of the player is illustrated below in Fig. 5. The advertised window length is equal to 60 s. In this example, player parameters are the following: *buffer* = 20 s, *backoff* = 6 s, *offset* = 7 s.

The first (newest) chunk received by the player is the one with timestamp $t_{start} = t_0 - backoff - offset$. Since the buffer length is greater than the offset, all the buffer may not be filled immediately. The player thus retrieves rapidly (as fast as bandwidth can support it) only "offset" portion of video chunks, and waits (*buffer-offset*) to gather enough newly arrived chunks to fill the rest of the buffer. Then, the player starts playing back, starting with the chunk t_{start}.

Summarizing and merging Eqs. (3) and (5) corresponding to different cases of player parameter settings, the formula for player delay can be written as:

$$D_{play} = backoff + \max(buffer, offset) \tag{6}$$

Remark that since packets sent over the network may be delayed or lost, causing a retransmission, the delay calculations should be treated as being "at least" values and the actual delay experienced may be greater than these values.

Playback Startup Delay

We may expect that playback startup delay (between the moment when the user clicks on the "play" button and the moment when the content actually starts playing) should grow with the size of the player buffer length. This is quite understandable because, while joining the live stream, the player must wait until the buffer is sufficiently filled, according to its configured value. More precisely, on the one hand, if the player buffer size is configured with a smaller value than the offset, the player immediately asks for video chunks to fill the buffer completely. The chunks are thus downloaded almost instantly and the time the player waits for filling the buffer is practically not observable. On the other hand, if the buffer size configured in the player is greater that the offset, the player cannot retrieve immediately the number of chunks required to fill the buffer. Thus, it has to wait

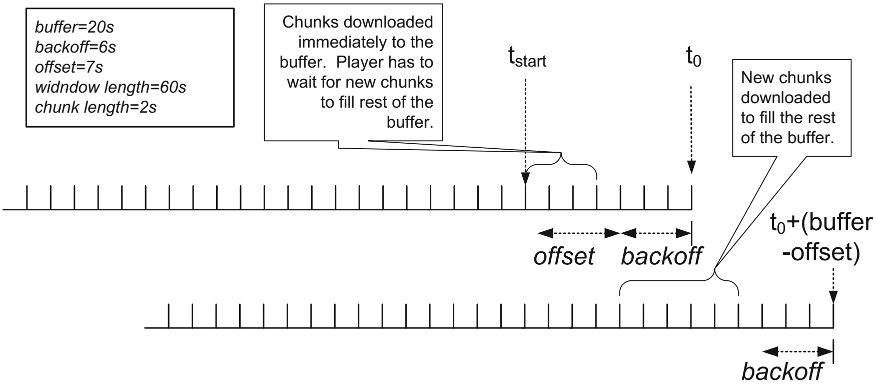

buffer=20s
backoff=6s
offset=7s
widndow length=60s
chunk length=2s
Chunks downloaded immediately to the buffer. Player has to wait for new chunks to fill rest of the buffer.
t_{start}
t_0
New chunks downloaded to fill the rest of the buffer.
offset
backoff
t_0+(buffer -offset)
backoff

Fig. 5 Illustration of player behavior in the case *buffer* > *offset*

until a sufficient number of new chunks appear on the origin server. The time it has to wait is equal to *buffer* minus *offset* (amount of video time that is missing in the current window stored on the origin server):

$$D_{play\_start} = \max(0, buffer - offset) \tag{7}$$

4 Experimental Measurements with MS Smooth Streaming Testbed

4.1 Testbed Architecture and Instrumentation

For experimental evaluation of the delay introduced in MS Smooth Streaming delivery chain, measurements have been carried out in a testbed which reflects the architecture of commercial OTT TV service of Orange Polska. It consists of the following elements:

- Encoder: Ffmpeg v2.2 transcodes the content into H.264 stream packaged in fmp4 (fragmented MP4) format.
- Origin Server: Unified Streaming Platform (USP) v1.5.7 packages fmp4 content into Smooth Streaming files and produces the manifest file. The content and manifest are served to clients by an Apache HTTP server.
- CDN: Akamai Verivue.
- PC player: a web-based player developed with MS Silverlight development tool.
- Mobile player: a reference application provided by the vendor of the streaming player software.

End-to-End Delay Measurement

The instrumentation used for measuring delays in this OTT testbed is presented in Fig. 6. The configuration of the encoder machine allows us for adding current timestamp as an overlay, visually "burned" in video picture. This entry-point timestamp (measurement point A) can be visually compared with the current time on the user device (measurement point B), after passing the entire delivery and decoding process. Both clocks (in measurement point A and B) are synchronized with a central clock by an NTP protocol.

The testbed allows us to perform measurements including - or not - the impact of the CDN. In the first case (path 1 on Fig. 6), the end device retrieves the content directly from the Origin Server, through a Local Area Network (LAN). The impact of the network latency can thus be considered as negligible and CDN is totally eliminated. In the second case (path 2), the player reaches the content through a test CDN, consisting of a single cache node.

The test executor launches the video player on a tablet which is connected to the test network. He reads the timestamps on the screens and calculates the delay. Current timestamp in point A (T_A) is embedded in each video frame. Simultaneous

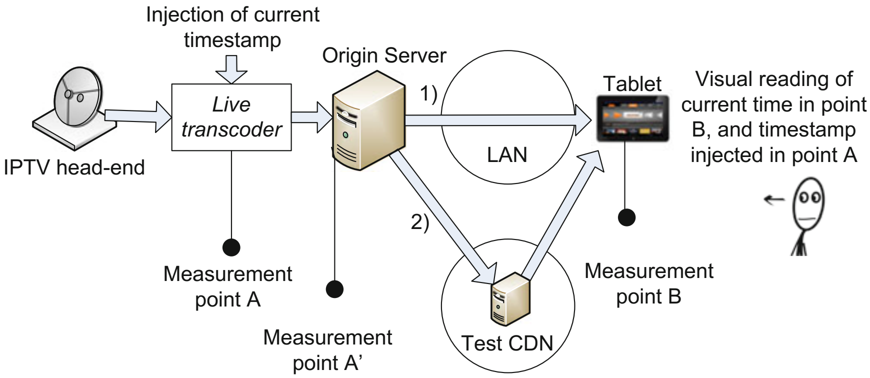


Fig. 6 End-to-end measurement testbed setup

readout of this timestamp and current timestamp (absolute time) in measurement point B (T_B) lets us estimate the total time of processing in the entire content delivery chain. The e2e delay can be calculated in at any moment as:

$$D_{e2e} = T_B - T_A \tag{8}$$

Measurement accuracy of this method is limited to visual readout of timestamps. Normally, human tester may read the timestamp from computer and tablet screen with granularity of around 1 s. More fine-grained measurement of time would require some automation of the method and more precise instrumentation. For limiting the impact of human error, each measurement was repeated several times. Taking into account that typically e2e delay in OTT delivery chain may be in the order of 20 s to 1 min, the granularity of the method seems to be sufficient.

Remark that presented method actually measures e2e delay, which is a sum of several delay components:

$$D_{e2e} = D_{enc} + D_{pack} + D_{CDN} + D_{play} \tag{9}$$

Additional actions must be taken to split this delay into particular components, as explained below.

Encoder Delay Measurement

Factors which impact the delay introduced by the encoding process include: encoder implementation efficiency, performance of hardware on which it is being run, whether the encoder itself is software or hardware based, numerous parameters that can be set on the encoder and may alter its performance.

The transcoder installed in the testbed and used in the scope of this study is a software-based solution ffmpeg 2.2, running on Centos 6.5 64 bit system, installed as virtual machine (Oracle VM VirtualBox, 1 GB RAM, 2 CPU). The virtual machine was running on Windows Server 2008 R2 Standard 64-bit (HP ML150: 2xIntel Xeon CPUE5504 2 GHz, 4 GB RAM).

Figure 7 gives more details about the configuration of the transcoder, presenting video processing steps and the detailed points where timestamp was embedded into the video for the purpose of measuring delays.

As the first step within the transcoder module, FFmpeg decodes the input stream to produce a raw video which is then encoded to a Smooth Streaming compatible format by the encoder. However, prior to encoding, FFmpeg process "burns" a timestamp in each produced video frame. The timestamp value in point A (T_A) corresponds to current system time when given frame has entered the transcoding process.

The output of the encoder is an fmp4 file, containing an amount of video corresponding to the duration of a chunk. Remark that, although the encoding and packaging processes are logically separated, the encoder is not totally independent from the packager as it prepares an encoded portion of the video which suits the packager's chunk size.

The encoded chunks (in fmp4 format) are then saved into the storage area of the transcoder machine. The time files are recorded in the file system is considered as a timestamp in point $A'(T_{A'})$. By comparing timestamp A' of a chunk with timestamp A of the last video frame of each chunk, we can estimate the delay introduced by the whole transcoding process.

$$D_{enc} = T_{A'} - T_A \tag{10}$$

Packager Delay Measurement

In order to evaluate the impact of the packager in the e2e delay budget, we have performed a set of measurements using the same methodology as described for the e2e delay, but with a specific setting of the player. By setting *buffer* = 1, *offset* = 0 and *backoff* = 0, we reduce the impact of player practically to zero. Without backoff and offset, the player gets the newest available chunk while joining the live stream (see Fig. 8). Since this single chunk is sufficient to fill the buffer, the player may start playing back immediately after receiving it. In a real network situation, such a configuration is not recommended since it is very sensitive to network impairments. However, in the "idealized" testbed environment, we were able to properly play a live stream with such a non-realistic parameter setting.

Since the player starts playing back immediately after receiving the newest chunk from the origin server, we may expect that observed delay in measurement

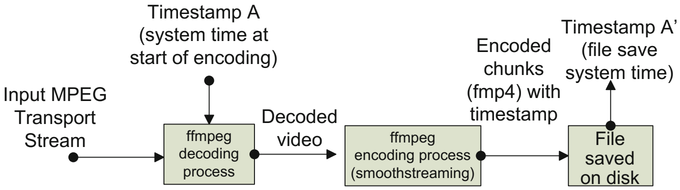


Fig. 7 Encoder configuration (with timestamp embedding) for measurements of encoding delay

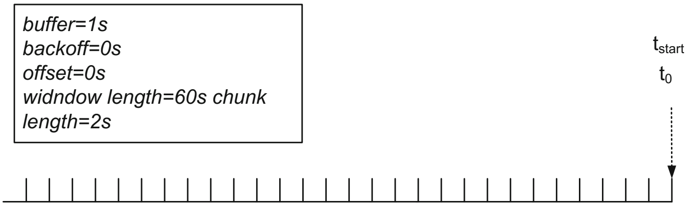


Fig. 8 Illustration of player behavior in the case *buffer* = 1, *offset* = 0, *backoff* = 0

point B is only related to the packager and encoding delay ($Dplay = 0$). So, the assumed procedure was to measure the e2e delay without CDN ($D_{CDN} = 0$) and subtract the encoding delay, obtained by the previous measurement of D_{enc}.

$$D_{pack} = D_{e2e} - D_{enc} \tag{11}$$

CDN Delay Measurement

As depicted in Fig. 6, testbed configuration allows for performing measurements with or without CDN in the delivery chain. The assumed indirect methodology for evaluating the impact of CDN only assumes comparing the end-to-end delay results measured "with" and "without" the CDN.

$$D_{CDN} = D_{e2ewithCDN} - D_{e2ewithoutCDN} \tag{12}$$

Player Delay Measurement

The methodology to evaluate the impact of the player assumes performing e2e delay measurements without CDN ($D_{CDN} = 0$) and then subtract the delays of the encoder and packager. Thus,

$$D_{play} = D_{e2e} - D_{enc} - D_{pack} \tag{13}$$

4.2 Test Results

Encoder Delay

The encoder delay was measured according to the methodology described in Sect. 4.1, with chunk length changed from 1 s to 10 s (remark that although chunk length is a parameter of the packager, the encoder must be configured accordingly in order to produce encoded video fragments that are suitable for the packager).

Several encoding profiles were tested (baseline, main), with FFmpeg-specific modes: medium, fast, ultrafast. The results of the experiments are presented in Table 1. Reported delays are an average calculated over five repetitions of each experiment.

Table 1 Measured encoder delay

Encoder profile	Chunk size (s)	Delay (s)
Baseline Fast	1	1.49
	2	1.74
	5	1.78
	7	1.76
	10	1.79
Baseline medium	1	1.54
	10	2.11
Main fast	1	1.73
	10	1.94
Main ultrafast	1	1.36
	10	1.19

The encoder delay in testbed environment is roughly between 1.5 and 2 s. We recognize that obtained results could differ for another encoder type, running in a different environment. Therefore, we stress that the results are relevant for particular hardware/software configuration of our testbed and cannot be generalized in a straightforward way to other types of encoders available on the market.

Packaging Delay

The packager delay was measured with various chunk lengths set on the packager (1–10 s), and with different values of the lookahead parameter (1, 2, 4, 6 fragments). The results are presented in Table 2 and Fig. 9 (subset of the results with *lookahead* = 2). The measured delay is compared with the D_{pack} theoretical value from Eq. (1). One can observe that measured packager delay can be quite well approximated by formula (1).

Table 2 Measured packager delay

Chunk length (s)	Lookahead	Measured packager delay (s)	Theoretical value of D_{pack} (Eq. 1) (s)
1	2	5.51	3.5
2	2	7.86	7
5	2	18.22	17.5
7	2	25.44	24.5
10	2	34.61	35
2	1	6.46	5
5	1	13.02	12.5
7	1	19.64	17.5
10	1	24.61	25
2	4	12.66	11
10	4	57.41	55
5	4	28.82	27.5
2	6	16.26	15

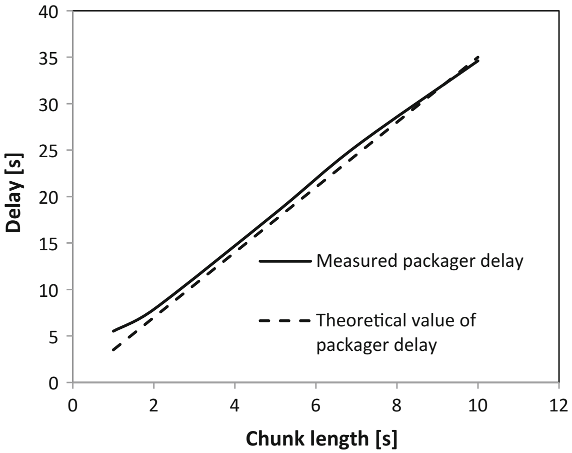


Fig. 9 Measured packager delay, with lookahead = 2

CDN Delay

Measurements were done with different values of chunk size on the packager (1, 2, 5, 7, 10 s) and with a fixed value of *lookahead* = 2. The results of measurements arepresented in Table 3. We can observe that CDN does not introduce a significant delay, especially when chunks are short.

Video Player Delay

E2e delay was measured without a CDN. The delay of encoder and packager was eliminated by subtracting the results of previous measurements performed with the same parameters setting.

Three series of tests were conducted in order to evaluate the impact on the video player delay of 3 parameters: *buffer*, *offset* and *backoff*.

In the first series of experiments, the delay was measured with different values of buffer length in the video player. The values of the two other player parameters were fixed to *backoff* = 6 s, *offset* = 7 s. Note that values *buffer* = 5 s, *backoff* = 6 s and *offset* = 7 s are considered as the default values in the Microsoft Smooth Streaming protocol. Two values are reported as result of experiments (see Table 4 and Fig. 10):

- "Start delay" corresponds to the stream startup time. It was measured with a stop watch, as the time between clicking "play" on the player and the actual display of the video. Reported value is an average over 5 repetitions of each experiment.
- "Player delay" corresponds to the observed difference between watched video and actual "live" position. Reported values are an average and a minimum value over 5 repetitions of each experiment.

One can observe that the player delay is quite well predicted using formula (6). Figure 10b shows the measured playback startup delay. We can observe that it is well approximated by formula (7).

Table 3 Measured CDN delay

Chunk length (s)	Delay with CDN (s)	Delay w/out CDN (s)	CDN delay (s)
1	6	6.6	0.6
2	8.8	9	0.2
5	14.4	15.8	1.4
7	18.6	22	3.4
10	26.2	30.8	4.6

Table 4 Measured player delay as function of buffer length

Player parameters (s)			Startup delay (s)		Player delay (s)		Theoretical D_{play} (Eq. 6) (s)
Buffer	Backoff	Offset	Avg	D_{play_start} (Eq. 7)	Avg	Min	
3	6	7	1.26	0	14.06	13.26	13
5	6	7	1.75	0	14.26	13.26	13
7	6	7	2.56	0	15.46	15.26	13
10	6	7	3.67	3	18.26	15.26	16
13	6	7	4.10	6	17.46	16.26	19
15	6	7	6.68	8	19.26	18.26	21
17	6	7	7.73	10	20.26	19.26	23
18	6	7	10.81	11	23.46	23.26	24
20	6	7	11.73	13	25.66	24.26	26
25	6	7	15.59	18	28.46	27.26	31
30	6	7	23.00	23	36.46	36.26	36

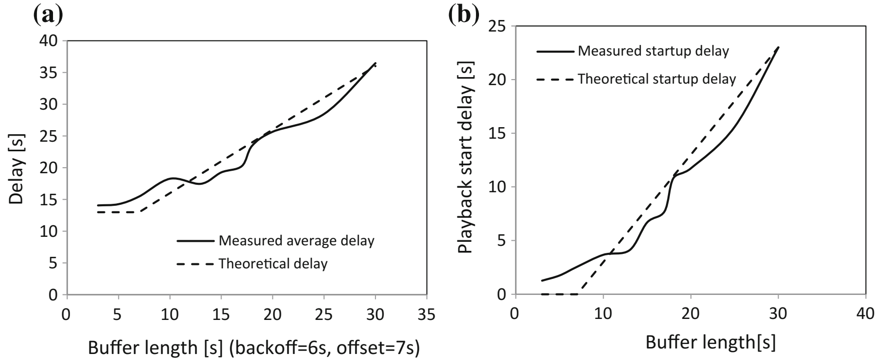


Fig. 10 **a** player delay and, **b** startup delay as function of player buffer length

In the second set of experiments, the player *offset* parameter varied, with fixed *buffer* length equal to 5 s, and fixed *backoff* equal to 6 s. The results are presented in Fig. 11a.

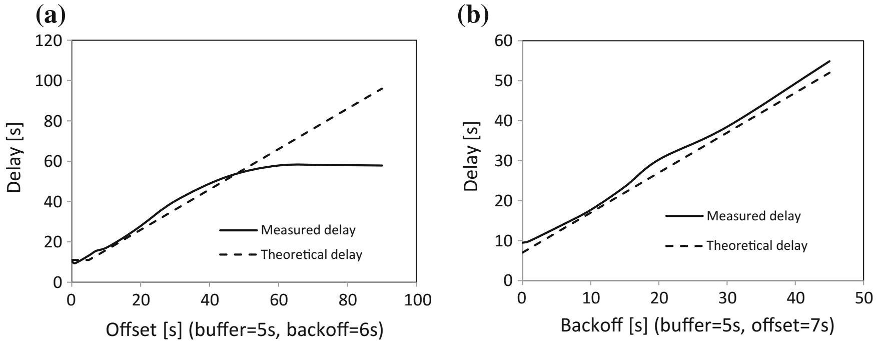


Fig. 11 Measured player delay as function of player **a** buffer offset, **b** buffer backoff

Note that the playback startup delay does not significantly depend on value of *offset* parameter because, in this particular case, the buffer is usually smaller than the offset (except in the two first measurements).

Once again, results confirm the validity of formula (6) for predicting latency of the player.

When *offset* is greater than 60 s, chunks that should be retrieved are out of the range of the advertised window, which means that the player wants to download chunks that are too old and do not exists anymore on the server. Thus, formula (6) does not apply.

In the last set of experiments, the player *backoff* time was varied in the range 0–90 s, with constant *buffer* = 5 s and *offset* = 7 s. The results are presented in Fig. 11b. Once again, the results confirm that the delay introduced by the player can be correctly estimated by Eq. (6).

5 Experimental Measurements with MPEG-DASH Testbed

5.1 Testbed Description

The delay measurements in an MPEG-DASH-based architecture were performed in a testbed setup by Thomson Video Networks and depicted in Fig. 12.

The transcoding + origin server device, a VS7000 product from Thomson Video Networks, is fed by an IP multicast containing a H264 specific source with a time code burned in the video.

- At Measurement Point #1 (MP #1), a PC with VLC player is placed to watch the input video (an IP multicast).
- At MP #2, a PC with VLC player is used to check the output video (an IP multicast).

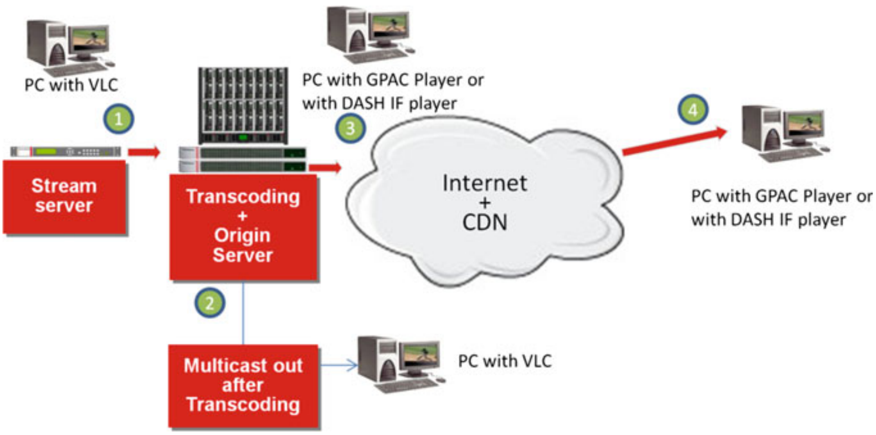

PC with VLC
1
Stream server
Transcoding + Origin Server
PC with GPAC Player or with DASH IF player
3
Internet + CDN
4
PC with GPAC Player or with DASH IF player
2
Multicast out after Transcoding
PC with VLC

Fig. 12 The basic MPEG-DASH testbed used for delay measurements

- At MP #3, a PC with GPAC player [9] or DASH IF player [10] is used to watch the DASH video directly at transcoder output.
- At MP #4, a PC with GPAC player or DASH IF player is placed to watch the DASH video output from a commercial CDN vendor, to simulate what would be the experience of the end user.

The VS7000 transcoder is able to output a transcoded H264 video (MP #2) at the same time as an MPEG DASH output (MP #3).

For the measurements, a snapshot (taken with a camera) of the PC screens at two measurement points is done (see a snapshot example in Fig. 13).

The measured end to end delay is just the difference between the time codes of the two screens. We did several measurements described hereafter by making variations on the DASH segment duration. The segment duration was set at 1, 2, 3, 6, and 10 s. We have tried 2 different DASH players, namely the GPAC player and the DASH IF player. Also with the GPAC player we tried two latency modes of the player (“normal” and “low latency”).

5.2 Measurement Results

Results with GPAC Player—Normal Mode

The GPAC player provides a configuration parameter called “delay mode” which was expected to impact the buffering and thus the delay observed by the end user. The measurements presented in this chapter were performed with two different settings of this parameter. Results of the measurement performed with GPAC player configured with “delay mode” set to “normal” mode are synthesized in Table 5.

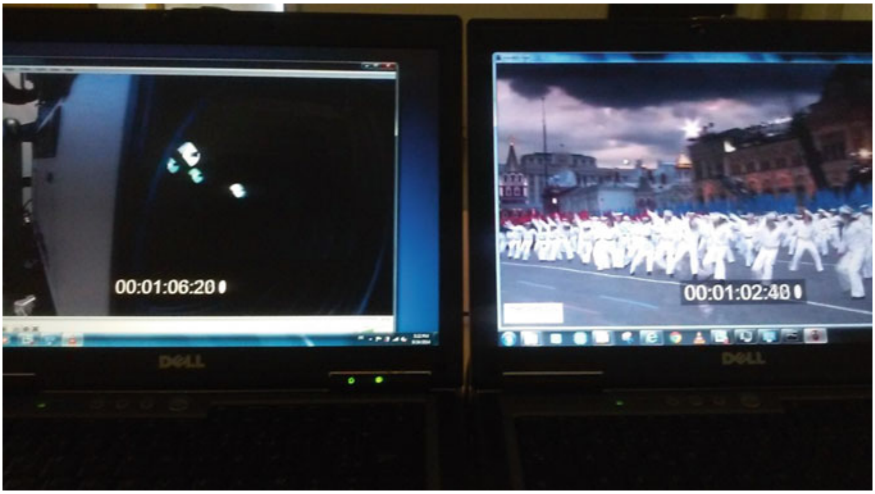


Fig. 13 A snapshot example

Table 5 Results with GPAC player—normal mode

Seg. duration (s)	Delay (s) at Enc out (MP2 − MP1)	Delay (s) at DASH out (MP3 − MP1)	Delay (s) at CDN out (MP4 − MP1)
1	6	–	10.2
2	6	10.4	10.2
3	6	12.2	11.1
6	6	16	19.1
10	6	24.5	25.8

We can observe that the transcoding delay measured on this testbed is always around 6 s and does not vary with the segment duration. The delay at DASH output and at CDN output increases clearly with the segment duration. The extra delay that the CDN can bring is not observable with these measurements. The minimum e2e that we can observe is 10 s when the segment duration does not exceed 2 s.

We noticed that the measurement of the delay on a DASH output brings an uncertainty of at least one segment. It means that the longer the segment is, the higher the uncertainty on e2e delay will be. It explains why we can have sometimes delays shorter with the CDN than without.

Results with GPAC Player—Low Latency Mode

Results of the measurements performed with GPAC player configured in "*low latency*" mode are synthesized in Table 6.

Table 6 Results with GPAC player—low latency mode

Seg. duration (s)	Delay (s) at Enc out (MP2 − MP1)	Delay (s) at DASH out (MP3 − MP1)	Delay (s) at CDN out (MP4 − MP1)
1	6	10	10.3
2	6	9.2	11.9
6	6	18.4	18.5
10	6	23.6	27.6

The conclusion is that we do not observe any significant change between the two latency modes of GPAC. Also, the delay of the CDN is still not observable with these measurements.

Results with DASH IF Player

The results of measurement performed with DASH IF player are synthesized in Table 7. This player does not provide configuration parameters related with buffering latency.

The conclusion is that we can observe an additional average delay of about 8.5 s on the DASH IF player compared to the GPAC player. This shows that the impact of the player is significant on the e2e delay.

Analysis of the Results

The transcoding time for H264 format is around 6 s, whatever the segment duration. The encoder is not the major contributor regarding the global 20–60 s e2e latency currently observed in the field.

The segment duration is a parameter that has a significant impact on the latency. Using a segment duration of 2 s allows getting an end-to-end delay close to 10 s with the best player. Using segment duration of 10 s, an e2e delay of around 25 s is observed with the same player.

The CDN delay does not seem to be a major contributor to the e2e delay. It is hidden by the delay introduced by the player which is the major contributor to the e2e latency. Using two different DASH players, GPAC and DASH IF, all the other parameters being the same, we observed an additional average delay of about 8.5 s with the DASH IF player. We did not observe any significant difference between the “low latency” and the “normal latency” mode of the GPAC player.

Table 7 Results with DASH IF player

Seg. duration (s)	Delay (s) at Enc out (MP2 − MP1)	Delay (s) at CDN out (MP4 − MP1)	Difference with GPAC (s)
1	6	19.7	+9.5
2	6	18.1	+7.9
3	6	22.2	+11.1
6	6	24.7	+5.6
10	6	34.4	+8.6

6 Improvement Proposals

We have identified several ways to improve the e2e delay. They are detailed hereafter.

End to End Delay Improvement Related to the Headend Architecture
A target for the OTT live service is to obtain an e2e delay as close as possible as the e2e delay observed with a broadcast network, a DTH (Direct To the Home) or DTT (Digital Terrestrial Television) network for instance. This is a way to improve the QoE for delay-sensitive services. With current architectures, the live program is acquired from a broadcast network (typically a satellite DTH network). Delay due to the broadcast network (encoding + multiplexing + RF transmission + decoding) is added to the OTT latency, which is not optimal.

It is much better for delay-sensitive OTT services to implement the live OTT headend at the same location as the broadcast headend in order to access the baseband uncompressed signal. With this approach, both encoding processes can be parallelized. Figure 14 illustrates this new architecture, compared to the current approach.

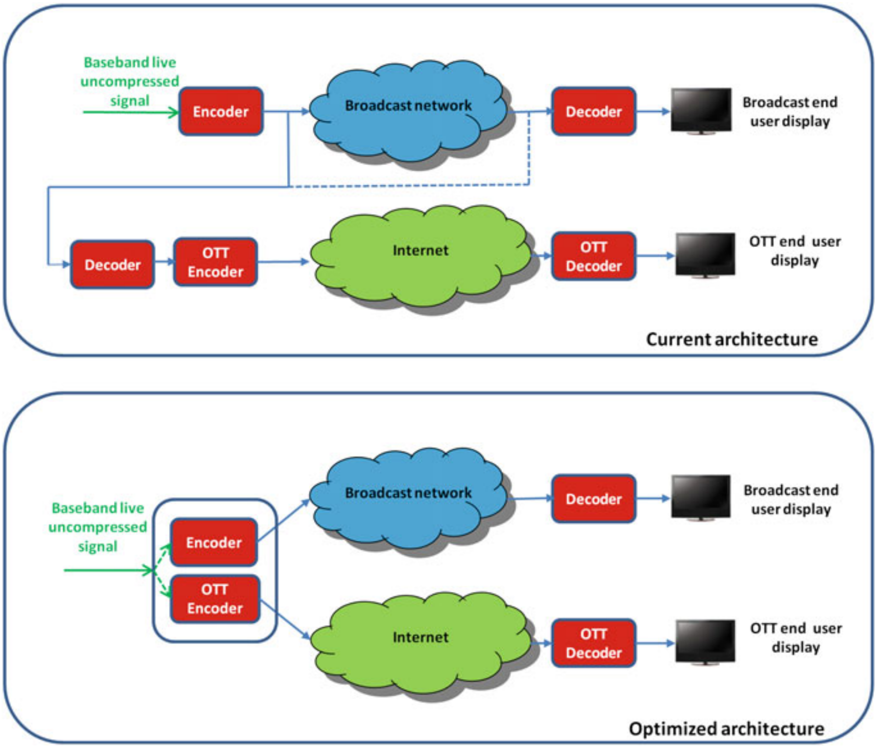


Fig. 14 Difference between current and optimized architecture

In the OTT delivery chain, this optimized architecture allows to remove the first encoding/decoding process, saving 4 to 5 s in the e2e latency, depending on the features and settings of this first encoding/decoding process.

End to End Delay Improvement with a Low-Delay Encoder
As exposed previously, with optimized player and segment duration, we can reach an e2e delay close to 10 s. So under these conditions the 6 s of encoding/transcoding delay is not negligible. Thomson Video Networks has developed an ultra-low delay mode for its OTT encoder.

It is mainly thanks to a reduction of the video encoding pipeline and of the video encoder buffer size that the encoding delay can be improved. With this approach, the encoding delay can be reduced by approximately 2 s.

End to End Delay Improvement When Reducing the Segment Duration
It was exposed previously that there is a direct link between the e2e latency and the segments duration.

The obvious solution to reduce the e2e latency is therefore to reduce the segment duration. However, the price to pay is a visible video quality degradation when this duration goes down to 1–2 s.

This is mainly due to the fact that, currently, OTT video coding is based on a "Closed Group Of Picture (GOP)" approach: Every segment begins with an Instantaneous Decoding Refresh (IDR) frame. The insertion of this IDR frame with a short recurring period has two negative effects. First, it increases the bitrate, or reduces the picture quality if we make the bitrate constant. Second, a "beating effect" is introduced at the recurring period because of the higher quality of the IDR frame.

It will be therefore very interesting to explore an "Open GOP" encoding mechanism for OTT delivery: IDR frames will not be systematically inserted at the beginning of a segment. It will however require some adaptations both at encoder and decoder side.

7 Future Works

Two areas are especially worth further investigation as next steps of work presented in this paper. As mentioned in Sect. 6, so-called "open GOP" approach is considered by authors as promising approach to further decrease the delay introduced by the OTT encoder. The impact on encoder and decoder implementation is to be evaluated, in relation to on-going MPEG-DASH standardization efforts.

On the other hand, further experiments are planned to evaluate the outcome of the study out of the lab, in real-life scenarios. Main contribution of this study, that is evaluation of delay introduced by OTT infrastructure (encoding-packaging-buffering) holds true in any realistic setting, but nevertheless it could be interesting to investigate how the proposed optimizations interplay with TCP protocol dynamics and impact on user-perceived QoE in the presence of network congestion, e.g. for OTT TV user with his tablet connected to WiFi in the home, sharing bandwidth with a connected TV, or mobile phone user in public 4G network in a crowded area.

8 Conclusions

The paper has presented the analysis and measurements of user-perceived delay in Live TV service delivered over the Internet using adaptive HTTP streaming technique.

The following elements of content delivery architecture have been identified as contributors to the e2e latency: video transcoding in the headend, packaging (applying adaptive streaming format), delivery over a CDN and buffering in the terminal. Buffering in the terminal appears as the major contributor to this delay but improvements in the headend architecture and in the video transcoder can also contribute to the reduction of the e2e latency. Presented results are of analytical as well of experimental type and may have practical importance for video service providers as hints for designing service architectures and setting key system parameters, taking into account both technical constraints and user Quality of Experience.

Measurements showed that an e2e delay of about 10 s can be obtained right now when applying some optimizations. This is clearly an improvement when compared to the 20–60 s range currently observed in the field. Shorter delays can even be reached with additional research work both at encoder and decoder side (use of "Open GOP" encoding mechanism).

References

1. Stockhammer, T.: Dynamic adaptive streaming over HTTP: standards and design principles. In: ACM MMSys'11. http://dx.doi.org/10.1145/1943552.1943572
2. Merkuria, R., Cesar, P., Bulterman, D.: Digital TV: The effect of delay when watching football. In: EuroITV'12, 10th European Conference on Interactive TV and Video, Berlin (2012). http://dx.doi.org/10.1145/2325616.2325632
3. Bouzakaria, N., Concolato, C., Le Feuvre, J.: Overhead and performance of low latency live streaming using MPEG-DASH. In: The 5th International Conference on Information, Intelligence, Systems and Applications, IISA 2014, Chania (2014)
4. http://www.streamingmedia.com/SponsoredContent/5800-Solving-sync-Synchronized-Live-OTT.htm. Accessed 29 Feb 2016
5. http://www.v-net.tv/what-it-means-for-tv-when-you-can-deliver-live-ott-streams-in-perfect-time-with-broadcast-signals. Accessed 29 Feb 2016
6. Dąbrowski, M., Kołodyński, R., Zieliński, W.: Analysis of video delay in Internet TV service over adaptive HTTP streaming. In: 4th International Symposium on Frontiers in Network Applications, Network Systems and Web Services, SoFAST-WS 2015 (FedCSIS2015), Łodź, Poland, 13–16 Sep 2015

7. EUREKA/CELTIC NOTTS. http://projects.celticplus.eu/notts/
8. Microsoft SmoothStreaming. http://msdn.microsoft.com/en-us/library/microsoft.web.media.smoothstreaming.smoothstreamingmediaelement.liveplaybackoffset(v=vs.95).aspx. Accessed 26 Jun 2015
9. GPAC, multimedia player with MPEG-DASH support. https://gpac.wp.mines-telecom.fr/player/
10. DASH-IF, a reference MPEG-DASH client. http://dashif.org/reference/players/javascript/1.4.0/samples/dash-if-reference-player/

Power Aware MOM for Telemetry-Oriented Applications—Levee Monitoring Use Case

Tomasz Szydlo, Piotr Nawrocki, Robert Brzoza-Woch and Krzysztof Zielinski

Abstract The paper discusses the problem of the message-oriented middleware utilization in telemetry systems. The authors provide a survey and practical measurements of common data transmission protocols for telemetry applications and wireless sensing. Based on that survey the authors propose concepts of message aggregation mechanisms to improve power consumption of the data transmission channel. As the entry point, the authors assume the utilization of the MQTT protocol. The concepts described in this paper have been successfully implemented in a smart levee monitoring system.

1 Introduction

The purpose of telemetry systems is to transparently convey measurement information from a remotely located sensor to receiving equipment for further processing and visualization. Development and miniaturization of electronic devices has allowed for the high penetration of telemetry solutions in the surrounding world in order to increase the quality of life. In typical telemetry solutions, remote stations are powered from external power sources and use industrial communication protocols such as Modbus to gather data from these devices to the central system. This work is an extended version of paper [1] presented at the Federated Conference on Computer Science and Information Systems, Warsaw, Poland, 2014.

T. Szydlo · P. Nawrocki (✉) · R. Brzoza-Woch · K. Zielinski
AGH University of Science and Technology, al. A. Mickiewicza 30,
30-059 Kraków, Poland
e-mail: piotr.nawrocki@agh.edu.pl

T. Szydlo
e-mail: tomasz.szydlo@agh.edu.pl

R. Brzoza-Woch
e-mail: robert.brzoza@agh.edu.pl

K. Zielinski
e-mail: kz@agh.edu.pl

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_16

Currently, an increasing number of telemetry devices are designed to be powered by energy harvesting thus they must be power efficient and they might temporarily go asleep to preserve power [2–4]. Because of the differences between these types of devices, the legacy polling protocols for communication might not be effective. To achieve the desired functionality, we need two components: (1) an adequate communication channel and (2) a suitable communication protocol. The choice of the communication channel technology is described further in this article. In the case of the communication protocols, they should (1) leverage the power usage characteristic of the used communication technology, (2) handle the sleepy nodes and (3) provide high level addressing of nodes.

We think that the requirements for communication protocol might be fulfilled by the message oriented communication. Sending messages across channels decreases the complexity of the end application, thereby allowing the developer of the application to focus on true application functionality instead of the intricate needs of communication protocols. *Message-oriented middleware* (MOM) [5] allows application modules to be distributed over heterogeneous platforms and reduces the complexity of developing applications that span multiple operating systems and network protocols. The middleware creates a distributed communications layer that insulates the application developer from the details of the various operating systems and network interfaces. Message-oriented middleware may provide reliable asynchronous communication mechanisms that might be used to carry i.e. measurement data or other remote communication messages. We have studied and tested different communication technologies and methods. After analysis of the different MOM protocols such as AMQP (Advanced Message Queuing Protocol), MQTT or MQTT-SN (MQTT For Sensor Networks) we have decided to choose for further research the MQTT and MQTT-SN protocols.

In the current telemetry solutions we distinguish two categories of communication channels. The *external communication* concerns data transmission between any telemetry station and the Internet. The second category is the *internal communication* which may be utilized within the telemetry system, but may be unable to transmit data directly to the global network. To implement the internal communication mechanisms we utilize mesh networking hardware and protocols. In a situation where there is no possibility of the external communication, telemetry station can connect to the other stations through the internal communication. In case of no external communication availability, the data from sensor networks can be transmitted over the mesh network until a telemetry station with the external communication available is found.

In the paper we are analysing communication technologies for internal and external communication and then we are comparing the energy efficiency of XBee and GPRS (General Packet Radio Service) technology. Then we propose the concept of adaptive message aggregation method for MQTT-SN protocol that optimizes power used by GPRS wireless connection during data transmission for the external communication. The research (presented later in the paper) showed that sending data using short IP packets consumes much more energy than using longer packets. Because of the fact that messages containing measurements are relatively small, we propose the concept of adaptive data aggregation prior to sending via GPRS.

The research presented in this paper is a part of ISMOP [6] research project which objectives span construction of an artificial levee, design of wireless sensors for levee instrumentation, development of a sensor communication infrastructure, and a software platform for execution management, data management [7] and decision support [8]. Scientific and industrial consortium in the ISMOP project conducts research on a comprehensive monitoring system enabling evaluation of current and forecasted state of flood levees. This paper focuses on issues related to the organization of data acquired from the sensors located in the levees in order to optimize the power consumed by GPRS modem during data transmission to the central system for later analysis.

The paper is organized as follows. Section 2 discusses the motivating scenario, where Sect. 3 presents the related work. Section 4 contains a description of preliminary tests of the MOM (based on the MQTT protocol) power and energy requirements. Section 5 presents the concept of power-aware adaptive message aggregation, which is then evaluated in the use case in Sect. 6. Finally, the paper is summarized and further research steps are presented.

2 Motivating Scenario

Recently, the importance of sensor network for monitoring various areas, objects or devices, has significantly increased. One of the areas in which telemetry and sensor network begin to fulfill a major role is monitoring systems for hydrologic engineering facilities in particular dams and flood levees [9, 10]. The overall concept of hydrological monitoring facilities was the starting point for the assumptions and implementation of the ISMOP research project which will result in guidelines for creating a telemetry system that enables continuous monitoring of levees. Research addresses the collection of massive measurement data in continuous mode, optimized transmission methodology, interpretation and analysis of monitored data with computer simulation and finally providing visualized results for the relevant authorities.

The condition of a levee can be determined by measuring its internal temperature and pore pressure in multiple places using a large number of sensors. A threat of burst can be estimated from the gathered temperature data. A rapid change of temperature detected in one or more sensors may be a sign of a leak which can further become a dangerous burst. An example scenario of a telemetry system containing sensors placed inside the levees and the central data collection system is shown in Fig. 1.

The role of the telemetry station is to acquire data from sensor networks which contain information about levee condition and to transmit these data to the central station. The data is generated by the sensor network for an *epoch*, which results in bursts of data each time the epoch changes. Apart from that, telemetry station also sends periodically information about its current condition such as battery level, CPU

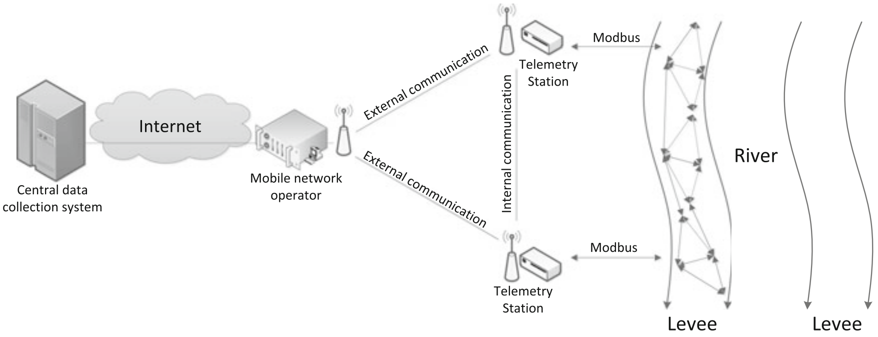


Fig. 1 Telemetry system for flood control levees

usage and others. The data from sensor networks, due to their importance, should be reliably delivered while the condition information may be transmitted on the best effort basis.

3 Related Work

Energy consumption of a wireless transmission device greatly depends on such factors as chosen communication standard, protocols used, and amount of transmitted data. Providing a medium-range or wide-area network connectivity requires a different approach. It is not a demanding task provided that a network infrastructure is available with appropriate SLA (Service Level Agreement) guaranties [11]. However, in remote areas a cellular connection is a common solution for industrial telemetry systems. Typical activities on a smartphone platform (sending a message, making a voice call, transmission over GPRS, etc.) are evaluated in [12]. An in-depth analysis o energy requirements for GPRS and UMTS services is provided in [13, 14].

High-level protocols over cellular network also have an impact on overall energy requirements of a system [15]. A review of various middleware protocols for telemetry applications can be found in [16]. Message-oriented Middleware is widely used as a communication layer for a variety of information systems which require event-driven message and data exchange, and more loose coupling than e.g. remote procedure calls. Examples of commonly utilized technologies for MOM are:

- Java Message Service (JMS) [17];
- Data Distribution Service [18];
- Extensible Messaging and Presence Protocol (XMPP) [19];
- MQTT and its variation, MQTT-SN [20].

These technologies provide several other functionalities such as transaction management, broker clustering, additional message paradigms including point-to-point, publish/subscribe and request-response. Nevertheless, only MQTT has been designed especially for transferring telemetry-style binary data from the pervasive devices with limited computational resources. It should be noted that utilizing the MQTT protocol over a standard TCP (Transmission Control Protocol) connection may provide redundant message delivery guaranties. As TCP is intended to provide a reliable link and has built-in retransmission mechanisms, setting the MQTT's QoS (Quality of Service) parameter to 1 or 2 provides another (redundant) layer of persistence. In contrast, those higher levels of QoS seem very useful in MQTT-SN variation which by design uses UDP (User Datagram Protocol) datagrams. In our research we have chosen MQTT-SN messaging protocol (formerly MQTT-S [21]) because it is promising due to its simplicity. MQTT-SN clients can be implemented in resource-constrained hardware (embedded systems), and there are available plenty of its implementations.

However, it is difficult to find any power efficiency considerations for MQTT-SN. By far many solutions dedicated to the MOM technology have been optimized to limit the data transfer and save energy. The MQTT-SN is an example of protocol that was optimized in terms of quantity of data to be transmitted. It results from using short-distance wireless protocols for sensor-based data transmission, including the IEEE 802.15.4 protocol. All these issues, not previously mentioned in MOM solutions, and particularly in the MQTT-SN protocol, have been analyzed in this article and relevant solutions have been suggested.

4 Power Aware MOM for Telemetry-Oriented System

There are several communication technologies that can be used in the telemetry-oriented applications. However monitoring levees in the hazardous weather conditions is not a trivial task and therefore needs technologies with particular properties. External communication should provide Internet connectivity with a good coverage in the rural areas. Internal communication should provide mesh connectivity as it will be used to transfer measured data to the other stations during the temporal failures of the external communication. Table 1 summarizes the available communication technologies that might be used.

Based on the aforementioned requirements, we have selected the XBee communication protocol for internal communication and GPRS for external communication. The tests described in this section concern power and energy requirements of the MOM based on the MQTT protocol implemented over these two technologies. Figure 2 shows the power consumed by the control-measurement station during four test *bursts*. Each burst consists of (a) reading data from the wireless sensor network edge router (b) processing the data, and (c) transmitting data in three different ways. The reading and processing procedures use the same underlying algorithms for each of the three cases. As can be noticed, data transmission is the main contributing factor for the overall power consumption footprint.

The XBee module's transmit operation required very little power (≈0.2 W during transmission) compared to the GPRS connection (≈3 W). It makes the XBee

Table 1 Communication technologies for telemetry applications

Technology	Topology	Range	Frequency	Applicability	Comments
GSM (GPRS/3G/LTE)	Star	omnipresent	850 MHz, 900 MHz, 1800 MHz, 1900 MHz	External	GPRS is a legacy technology with high area coverage
WiFi	Star	100 m to few km with high gain antenna	2.4 GHz	External/Internal	can be used for external communication when open HotSpots are available
XBee	Star, tree, mesh	Up to 10 km	2.4 GHz, 868 MHz, 900 MHz	Internal	proven technology for sensor networks
6lowPAN	Tree	100 m of direct link	868 MHz, 2.4 GHz	internal	IPv6 based technology
LoRA	Star	up to 20 km	868 MHz, 915 MHz	Internal	low power and low speed data transmissions over long distances

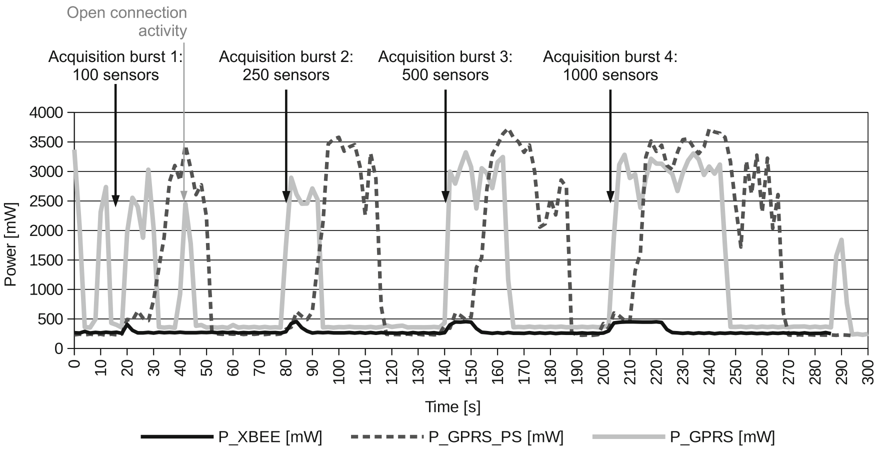


Fig. 2 Relation of power requirements in the function of time for four sample bursts

communication very well suited for the low-power data transmission in the control-measurement station and it does not require any sophisticated power saving mechanism. Despite XBee offers potentially economical station-to-station connectivity, it has no ability to transmit data to the Internet.

In contrast, the GPRS modem's power consumption is significant, but it has the advantage of providing connection to the Internet. In this case power saving mechanisms are justifiable. Further in this section we describe basic yet versatile method of implementing the GPRS power saving mechanisms at very low level. The mechanisms include (a) aggregating data to be transmitted (b) optionally completely disabling the power of the GPRS modem for a period of time. The latter requires the cold-start procedure to be performed each time the GPRS has to transmit data, but it saves the most power during the idle state. Then we compare the two methods.

The gathered information allowed us to determine the relationship between data aggregation time and energy that needs to be provided (harvested) in a period of time. To achieve this, we created a simplified energy model of the transmission subsystem with and without power saving (PS) feature.

First, we define the following symbols:

- T_{AG}—the data aggregation time (transmission interval), i.e. how long we wait and collect samples until transmission occurs (as one *sample* we mean one burst of data from all sensors);
- T_{SA}—sampling period, i.e. time interval between subsequent samples;
- T_1—one sample transmission time;
- T_{TOT}—total measurement period;
- E_{TOT_PS}—total energy required to transmit all data gathered during T_{TOT} with PS feature enabled (GPRS modem is powered down between transmissions);
- E_{TOT_ON}—total energy required to transmit all data gathered during T_{TOT} without the PS feature (GPRS modem is always powered and keeps the TCP connection);
- E_{TX_PS}, E_{TX_ON}—energy required for one transmission of aggregated samples with and without PS feature.

According to our model, we express total energy E_{TOT} for the two cases with the following equations:

$$E_{TOT\_PS} = E_{TX\_PS} \cdot N_{TX} \tag{1}$$

and

$$E_{TOT\_ON} = E_{TX\_ON} \cdot N_{TX}, \tag{2}$$

where N_{TX} is the number of transmissions that we need to perform during the total measurement period. It can be determined by the formula:

$$N_{TX} = \frac{T_{TOT}}{T_{AG}}. \tag{3}$$

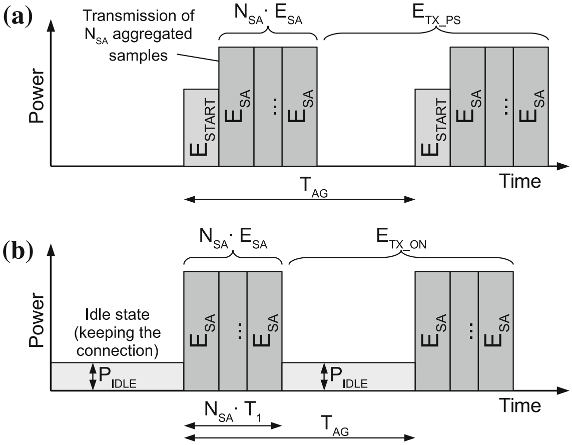
The number N_{TX} decreases with the increasing aggregation period.

Formulas for the energy required to perform one aggregated transmission (E_{TX_PS}, E_{TX_ON}) depend on the selected power scheme.

Refer to Fig. 3a. For the power-saving scheme, the E_{TX_PS} can be expressed as a sum the energy required to perform the GPRS modem start-up procedure E_{START} (one time) and the energy required to transmit one sample (E_{SA}) multiplied by the number of samples (N_{SA}):

$$E_{TX\_PS} = E_{START} + N_{SA} \cdot E_{SA} \tag{4}$$

Fig. 3 Schematic representation of the two power schemes: power saving (**a**) and always-on (**b**)



The number of samples to transmit (N_{SA}) is equal to the ratio of the aggregation time to the one sample period:

$$N_{SA} = \frac{T_{AG}}{T_{SA}} \tag{5}$$

Using Eqs. 1, 3 and 4 we get:

$$E_{TOT\_PS} = E_{START} \cdot \frac{T_{TOT}}{T_{SA}} + E_{SA} \cdot \frac{T_{TOT}}{T_{SA}}. \tag{6}$$

Refer to Fig. 3b. For the always-on scheme, we express the energy needed for the one aggregated transmission as a sum of energy needed to keep the connection active and the no-overhead data transmission energy:

$$E_{TX\_ON} = (T_{AG} - N_{SA} \cdot T_1) \cdot P_{IDLE} + E_{SA} \cdot N_{SA} \tag{7}$$

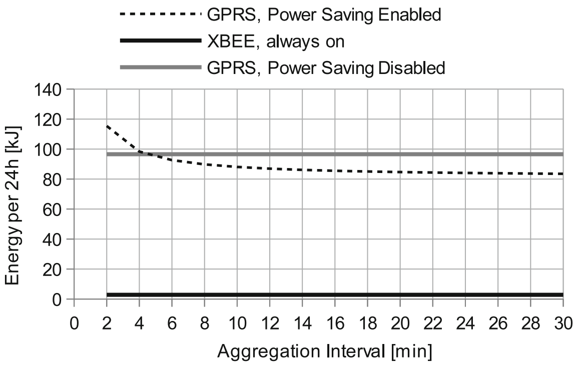
The first element is expressed as the idle power (P_{IDLE}) multiplied by the time in which we need to keep the connection opened and we not yet transmit data.

Then, from (2), (3) and (7), we get:

$$E_{TOT\_ON} = P_{IDLE} \cdot T_{TOT}(1 - N_{SA} \cdot \frac{T_1}{T_{AG}}) + E_{SA} \cdot \frac{T_{TOT}}{T_{SA}} \tag{8}$$

Figure 4 shows a sample relationship between the aggregation time T_{AG} and total energy required in a period of time with and without the power saving feature (E_{TOT_PS} and E_{TOT_ON}). The input values are based on the actual working implementation: $T_{TOT} = 24$ h, $T_1 = 42$ s, $T_{SA} = 2$ min, $E_{START} = 47$ J, $E_{SA} = 113$ J, $P_{IDLE} = 0,273$ W. Depending on the input values which represent given system characteristics, the E_{TOT_PS} and E_{TOT_ON} traces may vary. The E_{TOT_ON} trace is constant,

Fig. 4 Relation between the aggregation time and total energy required by the GPRS transmission subsystem of the prototype telemetry station



because the amount of data to be transmitted and idle time do not change, i.e. the transmission state to idle state ratio are constant and, depending on the aggregation interval (T_{AG}), transmission and idle periods are only differently organized. In contrast, the energy requirements with the PS feature enabled decreases with the growing number of the aggregated samples thanks to reduction of the number of cold start sequences of the GPRS modem.

5 Adaptive Message Aggregation for GPRS Connectivity

The main goal of the research was to decrease the amount of energy necessary to send the data using MOM over GPRS connectivity. The results of the base research aimed to analyze how much energy is used by GPRS modem as a function of packet data size is depicted in Fig. 5.

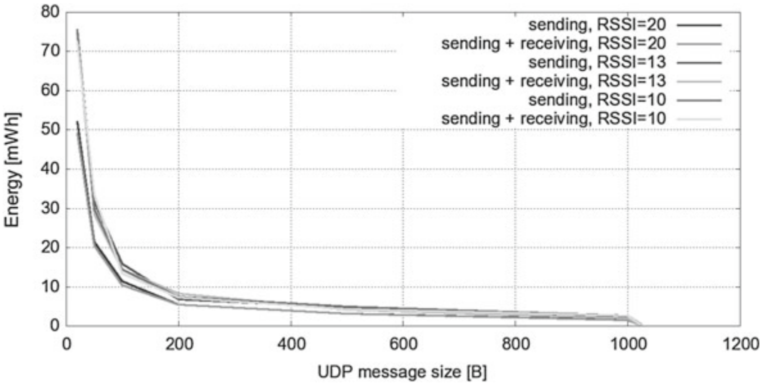


Fig. 5 Power necessary to send 10 kB of data using GPRS communication as a function of packet size

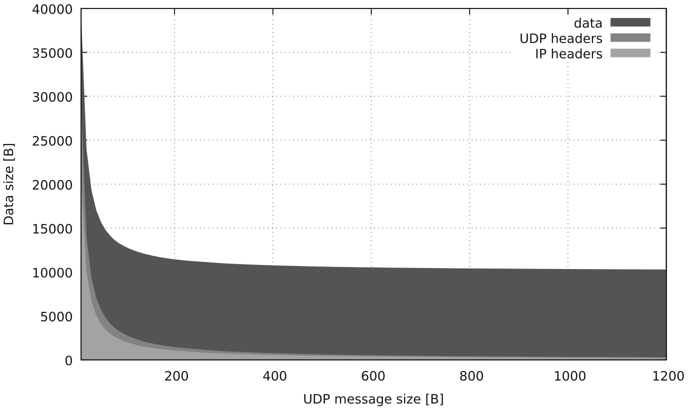


Fig. 6 Overall data necessary to send 10kB of data as a function of packet size

The nonlinearity in the power consumption is caused by two factors: overhead of the appropriate headers of TCP/IP protocol stack and purely technical considerations related to the physical communication with the GPRS modem in embedded devices (e.g. the time of data preparation, inter frame gaps and others). Figure 6 shows the overall data size that are necessary to send 10 KB of data payload. The overhead is related to the headers of UDP and IP protocols for each packet. Consequently, from an energy consumption point of view, it is much better for fixed amount of data to send it using as large packet size as possible.

On the other hand, the amount of measurement data that need to be transmitted is usually small i.e. typically 20 B. The previous measurements show that sending such small packets would be inefficient. Based on these data, we propose the concept of data aggregation before transmission.

Our concept, as presented in Fig. 7, can be applied to MQTT messages with QoS 0 and 1. In QoS 0, messages are not acknowledged, so client may aggregate several messages before sending them. In the second case, with QoS 1, all the messages had to be acknowledged so, we propose also aggregating the acknowledgment packets on the broker side.

We have assumed that larger, important data to send appear in the bursts, while less important data are sent on regular basis in small chunks. This is dictated by the fact that underlying sensor network (installed in the levee) wakes up in time intervals to preserve power. The naive approach to data aggregation is presented in Fig. 8. The main idea is that new messages are not send immediately but first copied to the buffer B with fixed length L and then sent as an aggregated packet. There are two conditions that decide of sending data: buffer is overflowed or buffer timeout T^I has ended. During the *period 1* and *3* aggregated messages are sent because of the timeout condition, while during the *period 2* aggregated message is sent because of the overflow condition. The drawback of the method is that in the *period 3*, the messages that belong to the burst are sent with longer delay then previous ones because

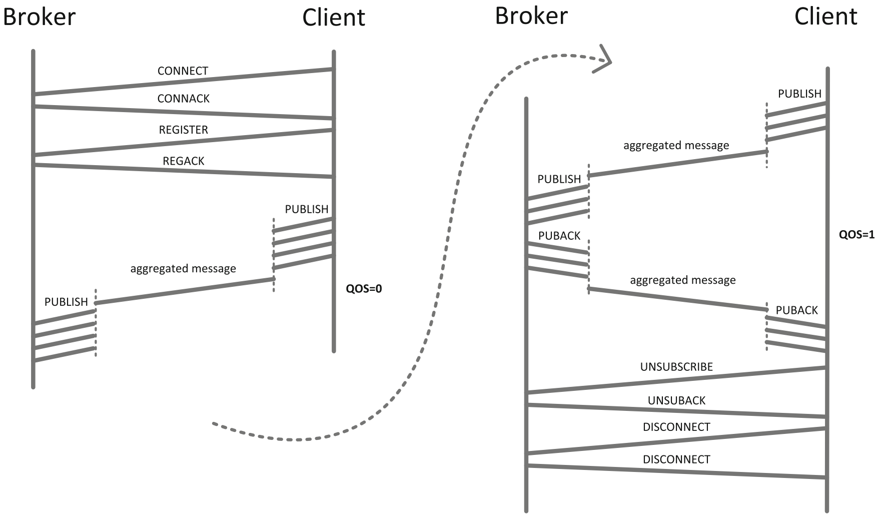


Fig. 7 Message aggregation concept for MQTT-SN

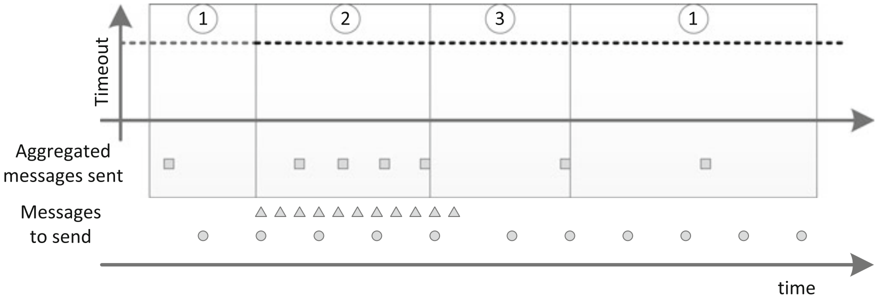


Fig. 8 Naive approach to data aggregation (time periods are marked with *numbers*, telemetry data are represented by *small grey circles*, *triangles* represent bursts of measurement data, packed and transmitted data are depicted as *squares*)

overflow condition did not occurred. Such a situation is unwanted if the data has to be analyzed in the real-time.

In our method, we propose adaptive timeout calculation that adjust itself to the frequency of incoming data. The main concept is that buffer overflow situation decreases the buffer timeout meaning that messages should be send faster, while decreasing the frequency of incoming new data recovers the timeout to its previous value. Such a policy results in the situation that messages belonging to the data burst are received by the broker in the burst as well. The concept is depicted in Fig. 9. During the *period 3*, the aggregated message is send earlier than in naive approach to the aggregation.

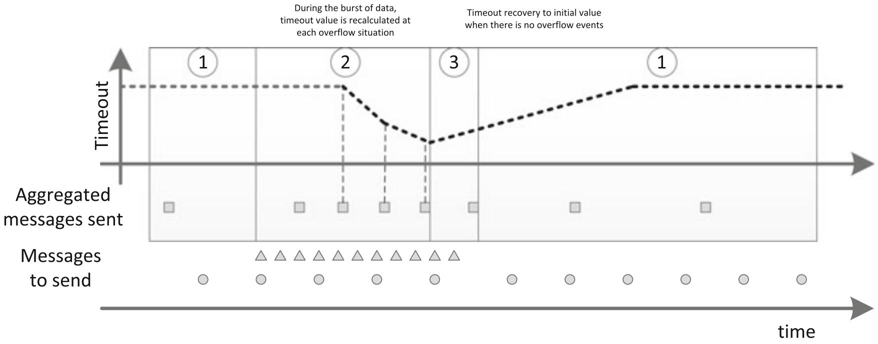

During the burst of data, timeout value is recalculated at each overflow situation
Timeout recovery to initial value when there is no overflow events
1
2
3
1
Timeout
Aggregated messages sent
Messages to send
time

Fig. 9 Adaptive approach to data aggregation (time periods are marked with *numbers*, telemetry data are represented by *small grey circles*, *triangles* represent bursts of measurement data, packed and transmitted data are depicted as *squares*)

As can be noticed, the presented adaptive message aggregation does not generate any additional overhead. On the contrary, it contributes to reducing the overhead introduced by the TCP/IP protocol stack when transmitting small chunks of data. Another aspect is the latency of data arrival. It results directly from the fact, that data needs to be packed in a buffer. In our solution, the latency is controlled with the variable timeout value. The timeout automatically decreases as more data arrives. This can be utilized to keep latency and overhead ratio at reasonable levels.

More formally, the algorithm is composed of the two parts and might be presented as follows. The input to the algorithm is provided by four values: T^I is the initial timeout value, T^R is the recovery time to the initial value, the factor α, and a buffer length L. In the MOM client there is global timer T that represents actual timeout value—at time t this value is denoted as T_t. When the aggregated message buffer B of length L is created at time t, it has assigned timeout that equals T_t. Length of the buffers is constant.

First part of the algorithm is responsible for recovering (i.e. increasing) timeout T to the initial value of T^I and is formulated as follow: for each time k, the timeout T_{k+1} is calculated using the Eq. 9, where ΔT is the time step.

$$T_{k+1} = \min(T_k + \frac{T^I}{T^R}\Delta T, T^I) \tag{9}$$

The second part of the algorithm is responsible for decreasing timeout T to the value that is similar to the time of sending overflowed buffers when data burst is observed. The Eq. 10 is used only when the overflow of the buffer is observed. Value d in the equation is the time from the last overflow event.

$$T_{k+1} = \min(\alpha T_k + (1-\alpha)d, T^I) \tag{10}$$

Having in mind, that data usually comes from remote telemetry stations to the central point, we propose to use adaptive aggregation method on the client side to aggregate messages, and to use naive aggregation approach on the broker side to aggregate acknowledgments. The evaluation results of the proposed algorithm are presented in the next section.

6 Evaluation

We have evaluated the proposed concept on the scenario similar to the one presented in the previous section. We have assumed that data from levee monitoring sensors are gathered and sent in two stages:

- at the beginning, for QoS 1 in MQTT/MQTT-SN, 1000 PUBLISH messages with a length of 20 B are sent and received confirmation of these messages (PUBACK);
- later, for QoS 0 in MQTT/MQTT-SN, in 12 min epoch and for every 30 s PUBLISH messages with a length of 20 B are transmitted.

During tests we used a popular GPRS modem (SIM900D) and, in order to verify the results obtained, we also used an industrial GPRS Modem (Wavecom Fastrack Supreme 20). In order to develop test software we extend implementation of MQTT-SN— Eclipse Mosquitto [22] (which we call *A-MQTT-SN*) to support adaptation.

Above presented testing scenario was carried out for three cases using:

- MQTT protocol (Eclipse Mosquitto);
- MQTT-SN protocol (Eclipse Mosquitto);
- A-MQTT-SN protocol with adaptation for sent and received data (message type PUBLISH and PUBACK).

The adaptive aggregation algorithm for A-MQTT-SN was initiated with values: $TI = 120$ s, $TR = 240$ s, $\alpha = 0.5$, and $L = 1000$ B. The naive aggregation algorithm was initiated with values: $T_I = 2$ s and $L = 250$ B. The values are application-specific, and should be tailored for different conditions, such as: amount of transmitted data, real-time boundaries and the maximal accepted latency by the application.

The measurements were made with a custom multichannel current and voltage sensing module and tailored for energy measurements of various embedded devices. Data for all of the presented tests was acquired from the GPRS modems (Class 10) connecting to public GSM network with a throughput of 25 Kb/s (2 timeslots in uplink direction).

The result of these tests is shown in the following figures:

- for MQTT protocol (using TCP and PPP protocols)—Fig. 10;
- for MQTT-SN protocol (using UDP, PPP protocols and AT commands on the GPRS modem)—Fig. 11;
- for A-MQTT-SN protocol (using UDP, PPP protocols and AT commands on the GPRS modem)—Fig. 12.

The above results show that the power consumption of a GPRS modem for data transmission is higher for MQTT and MQTT-SN than A-MQTT-SN protocol. In our

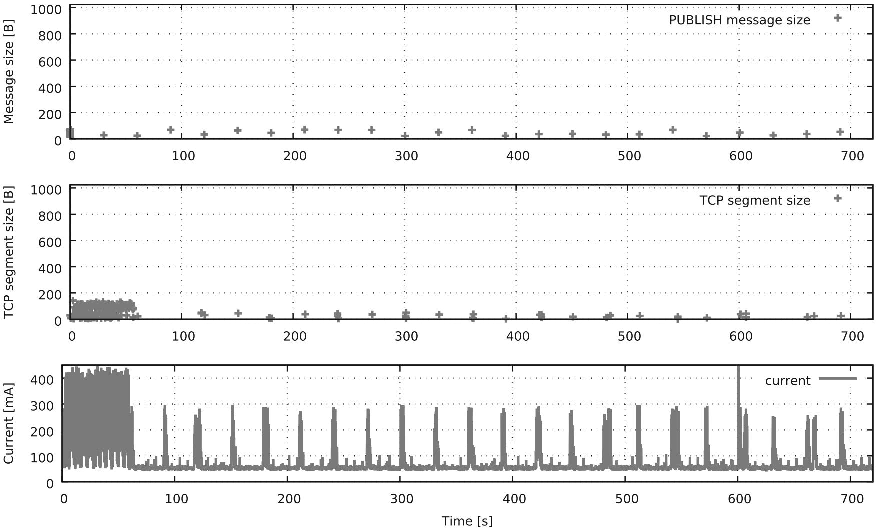


Fig. 10 The current consumption for the MQTT protocol

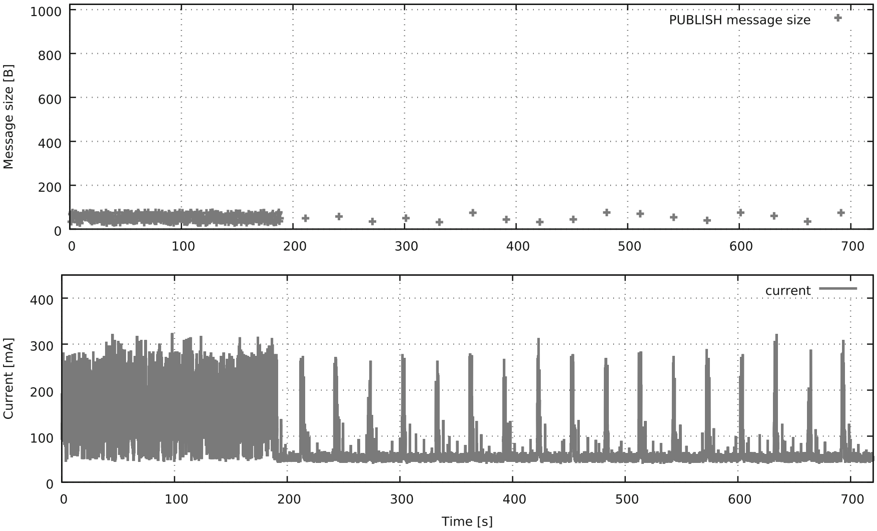


Fig. 11 The current consumption for the MQTT-SN protocol

opinion, the higher value of power consumption for MQTT protocol is the result of using TCP and its complexity (call setup, retransmissions). When we transmit MQTT messages with QoS 0, there should be no retransmissions at the MQTT protocol level. However, the retransmissions may actually occur at the TCP protocol level. In the second case the increased energy consumption of the GPRS modem is related

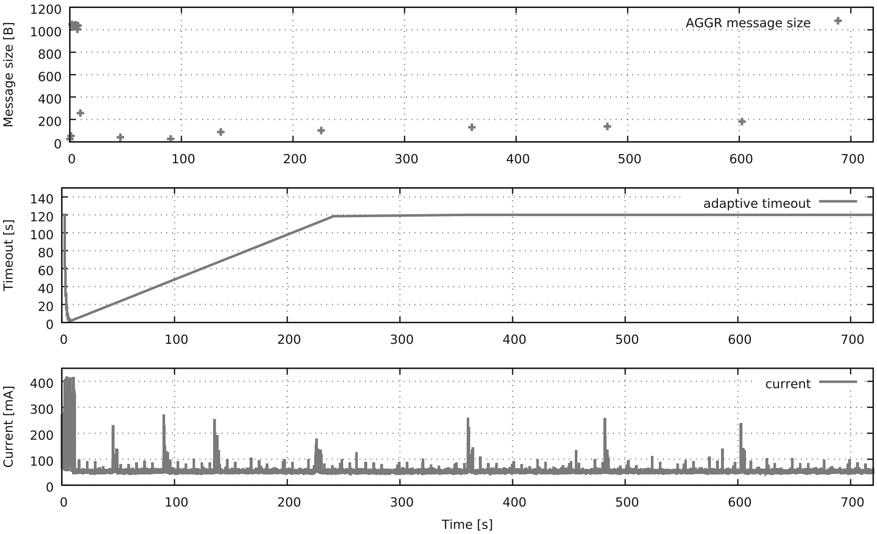
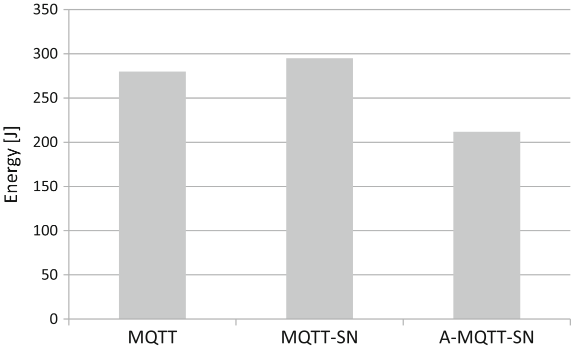


Fig. 12 The power consumption for the A-MQTT-SN protocol

Fig. 13 Energy consumption for MQTT, MQTT-SN and A-MQTT-SN protocols



to the size of transmitted data chunks with the MQTT-SN over UDP. Small chunks of data result in an increased overhead related to the packets' headers. The developed A-MQTT-SN protocol variation aggregates data and significantly decreases the number of headers that need to be transmitted. This, in turn, provides the best energy efficiency (Fig. 13).

7 Summary and Future Work

The paper discusses the problem of sending the sensor data from and between remote telemetry stations. We have analyzed various communication technologies and protocols for both internal and external communication and decided to use XBee and GPRS connectivity. The paper also proposed the extensions to the communication protocol that adjust its behavior to the GPRS connectivity profile in order to decrease the data transmission-related energy consumption.

The motivating scenario presented in the paper is only one of the possible applications of our concept. The solutions might be successfully applied to e.g. multilayer telemetry solutions where due to the sleepy nodes, data have to be pushed rarely but efficiently.

Acknowledgements The research presented in this paper was partially supported by the National Centre for Research and Development (NCBiR) under Grant No. PBS1/B9/18/2013 and by the Polish Ministry of Science and Higher Education under AGH University of Science and Technology Grant 11.11.230.124 (statutory project).

References

1. Szydlo, T., Nawrocki, P., Brzoza-Woch, R., Zielinski, K.: Power aware MOM for telemetry-oriented applications using gprs-enabled embedded devices - levee monitoring use case. In: Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Warsaw, Poland, 7–10 Sept 2014, pp. 1059–1064 (2014). doi:10.15439/2014F252
2. Szydlo, T., Brzoza-Woch, R.: Predictive power consumption adaptation for future generation embedded devices powered by energy harvesting sources. Microprocess. Microsyst. Embed. Hardw. Des. **39**(4–5), 250–258 (2015). doi:10.1016/j.micpro.2015.05.001
3. Szydlo, T., Gut, S., Puto, B.: Smart Applications: Discovering and interacting with constrained resources IPv6 enabled devices. Przeglad Elektrotechniczny, 221–226 (2013)
4. Szydlo, T., Suder, P., Bibro, J.: Message oriented communication for IPV6 enabled pervasive devices. Comput. Sci. **14**(4) (2013). doi:10.7494/csci.2013.14.4.667
5. Curry, E.: Message-oriented middleware. In: Mahmoud, Q.H. (ed.) Middleware for Communications, Chap. 1, pp. 1–28. Wiley, Chichester, England (2004). doi:10.1002/0470862084.ch1
6. ISMOP Project (2013). www.ismop.edu.pl. Accessed 30 Nov 2015
7. Piórkowski, A., Leśniak, A.: Using data stream management systems in the design of monitoring system for flood embankments. Studia Informatica **35**(2), 297–310 (2014)
8. Chuchro, M., Lupa, M., Pięta, A., Piórkowski, A., Leśniak, A.: A concept of time windows length selection in stream databases in the context of sensor networks monitoring. In: New Trends in Databases and Information Systems, Proceedings of 18th East-European Conference on Advances in Databases and Information Systems (in print). Advances in Intelligent Systems and Computing. Springer (2015)
9. Balis, B., Bartynski, T., Bubak, M., Dyk, G., Gubala, T., Kasztelnik, M.: A development and execution environment for early warning systems for natural disasters. In: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 575–582 (2013). doi:10.1109/CCGrid.2013.101
10. Balis, B., Kasztelnik, M., Bubak, M., Bartynski, T., Gubaa, T., Nowakowski, P., Broekhuijsen, J.: The urbanflood common information space for early warning systems. Proc. Comput. Sci. **4**(0), 96–105 (2011). http://dx.doi.org/10.1016/j.procs.2011.04.011. Proceedings of the International Conference on Computational Science, ICCS 2011
11. Kosinski, J., Nawrocki, P., Radziszowski, D., Zielinski, K., Zielinski, S., Przybylski, G., Wnek, P.: SLA monitoring and management framework for telecommunication services. In: Bi, J., Chin, K., Dini, C., Lehmann, L., Pheanis, D.C. (eds.) Fourth International Conference on

Networking and Services, 2008. ICNS 2008, pp. 170–175. IEEE Computer Society (2008). doi:10.1109/ICNS.2008.31

12. Perrucci, G.P., Fitzek, F.H., Widmer, J.: Survey on energy consumption entities on the smartphone platform. In: Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd, pp. 1–6. IEEE (2011). doi:10.1109/VETECS.2011.5956528
13. Sikora, A., Yunitasari, A., Dold, M.: GPRS and UMTS services for ultra low energy M2M-communication. In: 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), vol. 1, pp. 494–498. IEEE (2013). doi:10.1109/IDAACS.2013.6662734
14. Pauls, F., Krone, S., Nitzold, W., Fettweis, G., Flores, C.: Evaluation of efficient modes of operation of GSM/GPRS modules for M2M communications. In: Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th, pp. 1–6. IEEE (2013). doi:10.1109/VTCFall.2013.6692200
15. Alonso, E.J.V.: Exploiting energy awareness in mobile. Communication (2013). doi:10.3384/lic.diva-98656
16. Azzara, A., Bocchino, S., Pagano, P., Pellerano, G., Petracca, M.: Middleware solutions in WSN: the IoT oriented approach in the ICSI project. In: 2013 21st International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1–6. IEEE (2013). doi:10.1109/SoftCOM.2013.6671886
17. Java Message Service Specification. https://jcp.org/en/jsr/detail?id=343. Accessed 30 Nov 2015
18. Data Distribution Service ver. 1.2 documentation. http://www.omg.org/spec/DDS/1.2. Accessed 30 Nov 2015
19. Extensible Messaging and Presence Protocol documentation. http://xmpp.org/xsf/press/2004-10-04.shtml. Accessed 30 Nov 2015
20. MQ Telemetry Transport (MQTT) documentation. http://mqtt.org/documentation. Accessed 30 Nov 2015
21. Hunkeler, U., Truong, H.L., Stanford-Clark, A.: MQTT-SA publish/subscribe protocol for wireless sensor networks. In: 3rd International Conference on Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008, pp. 791–798. IEEE (2008). doi:10.1109/COMSWA.2008.4554519
22. Mosquitto technology project. http://projects.eclipse.org/projects/technology.mosquitto. Accessed 30 Nov 2015

Monitoring Drip Irrigation System Using Wireless Sensor Networks

I. Bennis, H. Fouchal, O. Zytoune and D. Aboutajdine

Abstract Recently, adopting an optimized irrigation system has become a necessity due to the lack of the world water resource. Moreover, many researchers have treated this issue to improve the irrigation system by coupling the novel technologies from the information and communication field with the agricultural practices. The Wireless Sensor and Actuators Networks (WSANs) present a great example of this fusion. In this paper, we present a model architecture for a drip irrigation system using the WSANs. Our model includes the soil moisture, temperature and pressure sensors to monitor the irrigation operations. Specifically, we study the case where a system malfunction occurs, as when the pipes burst or the emitters are blocked. Furthermore, we differentiate two main traffic levels for the information transmitted by the WSAN, and by using an adequate priority-based routing protocol, we can achieve high QoS performances for the priority information. We have performed extensive simulations through TOSSIM simulators. The results show that our solution gives better performances in terms of delay and packet delivery ratio. Also we have realized a real test-bed to investigate the effectiveness of our approach. The experimentation results show considerable gain compared to other state-of-the-art protocol.

Keywords WSANs · Drip irrigation · TOSSIM · TelosB · Routing protocol

I. Bennis (✉) · H. Fouchal
Université de Reims Champagne-Ardenne, Reims, France
e-mail: ismail.bennis@univ-reims.fr

H. Fouchal
e-mail: hacene.fouchal@univ-reims.fr

O. Zytoune
Université Ibn Tofail, Kenitra, Morocco
e-mail: zytoune@univ-ibntofail.ac.ma

D. Aboutajdine
Université Mohammed V Rabat, Rabat, Morocco
e-mail: aboutaj@fsr.ac.ma

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent
Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_17

1 Introduction

During the last decade, the Precision Agriculture (PA) has emerged as novel trend to enhance the agricultural practices. The principal aim of the PA is to monitor the spatio-temporal characteristics of the agricultural parcel [1]. In that way, the crops yield can be optimized while the natural, financial and energetic resources can be preserved. However, since the monitored agricultural regions are generally scattered and suffer from variable environmental conditions, the need for accurate and real-time collected information is more pronounced. Also, the classical solution as the satellite imagery, aircraft or other systems based on the map cannot be supported by all farmers due to their heavy cost. To overcome this limitation, the Wireless Sensor Networks (WSNs) was introduced into the agricultural environment [2].

Technically, the sensor nodes are deployed into the farmland. They start to collect environmental information and monitor soil characteristics. Then, they cooperate according to designed protocols to communicate the collected information to a central node. After that, this information is processed and treated to make an eventual decision.

The WSN have been explored in different ways for the agriculture field. As example, in [3] the authors have used four nodes types: soil, environmental, water and gateway to monitor the water content, temperature and soil salinity at a farm located in Spain. The security aspect is another example of how can the WSNs improve the agricultural yield. In fact, crops are negatively affected by human or animals intruders. Also, the production process is still insufficiently controlled which leads to a potential product loss. To overcome this point, the video-surveillance nodes can be used to detect and identify intruders as well as to better take care of the production process [4]. The detection can be performed by the Passive Infrared sensor (PIR), when the identification process can be done by CMOS camera.

Another kind of WSN was proposed focusing in the underground communication. In fact, in order to monitor the underground parameters such as soil moisture or water and mineral content, the terrestrial sensors nodes were usually connected by cable to a wireless transceiver on the ground [5]. However, such way can highly influence the farming activities. In addition, the sensor nodes can be affected by geographical and meteorological factors [6]. To overcome this shortcoming, the entire node (including the transceiver part) is buried at a specific depth. This network is called Wireless underground sensor networks (WUSN) [5]. According to [5], the main advantages of the WUSN against terrestrial WSN can be resumed to the following points: concealment, ease of deployment, data timeliness, and coverage density. Nevertheless, even if the adoption of the WUSN is a promising solution, ensuring a reliable wireless underground communication is almost a new topic for researchers in the precision agriculture field which implies novel challenges concerning the underground channel properties.

One of the most important applications of the WSNs in the PA was the control of the irrigation system. The interest comes naturally from saving water. For this aim, many researches were conducted to enhance the irrigation control system by coupling novel technologies with the agricultural practices. Among irrigation strategies, the drip irrigation system was considered as the most efficient policy to save water use. Moreover, combining this strategy with WSNs leads us to have a great benefit from the farmlands. However, the irrigation system reliability needs more attention, mainly in the case of general or partial dysfunction. For this aim, we present in this paper a model architecture for a drip irrigation system using WSANs. Our model includes the soil moisture, temperature and pressure sensors to monitor the irrigation operations. Specially, we take into consideration the case when a dysfunction of the system occurs, as when the pipes are broken or the emitters are blocked. Also, we differentiate two main traffic levels for the information transmitted by the WSAN. Furthermore, based on our previous work [7], we can achieve a high QoS performance through an adequate priority-based routing protocol. The aim was to ensure an efficient and real-time communication between the different nodes type and the sink. We note that this work is an extended version of [8].

The remainder of this paper is organized as follows: in Sect. 2, we review some related works designed for an efficient irrigation system. In Sect. 3, we discuss our designed drip irrigation system with a description of the used priority-based routing protocol. The simulation and the experimental results for our proposed scheme are given in Sect. 4. Finally, in Sect. 6, we draw the conclusion and give some perspectives.

2 Related Work

To the best of our knowledge, monitoring the dysfunction of the drip irrigation system using the WSNs with an adequate priority-based routing protocol was never suggested before in the specialized literature. Therefore, in this section we summarize some related works for the irrigation system control.

In [9], the authors propose an energy efficient method for the wireless sensor communication used in an automated irrigation system. This method is based on the Time Division Multiple Accesses (TDMA) scheduling that allows nodes to turn ON/OFF their radio according to scheduled slots. The main advantage of such scheme is saving the node's energy and reducing radio interference. Also, the authors give a comparison between two methods to transmit the collected data to the sink node; namely the direct communication method and the data fusion method. For each method, the energy consumed and the data throughput are studied over the NS2 simulator.

To optimize water use in agricultural context, the authors propose in [10] an automated irrigation system based in the WSNs technology. The developed system is composed of two kinds of sensors to collect soil-moisture and temperature information. The sensors are placed in the root zone of the plants. Also, a gateway was used to gather sensor information, triggers actuators, and transmits data to a web application. To control the water quantity, the authors had programmed into a micro-controller an algorithm with threshold values of temperature and soil moisture. Concerning the energy, photo-voltaic panels are used to power the system. The entire system can be controlled through a web page which help to program an irrigation schedule and performs a data inspection.

In [11], the authors present practical irrigation management system using a deployed WSN. This system includes a remote monitoring mechanism through a GPRS module to send SMS message containing land characteristic such as soil temperature and soil moisture, or the network performances such as packet delivery ratio, RSSI or the nodes energy level. The main contribution of this paper is to design and implement a low-cost efficient irrigation management system that combines sensors and actuators in a wireless sensor/actuator network. The authors conclude through this study that the deployment of the sensor nodes in the agricultural field is a critical issue. Furthermore, they suggest that the distance between sensor nodes has to be as short as possible in order to enhance the effectiveness of the system. However, the main weakness of this study is that the authors employ only five sensors for the experiment.

We conclude for all referred works, that the authors don't take into consideration the case of irrigation system dysfunction. Also they don't use the pressure sensor to monitor the irrigation flow rate. In addition, no priority-based protocol is designed to distinguish the importance of the communicated information. In the following section we present our proposed drip irrigation system that can overtake the dysfunction case.

3 DIS: Drip Irrigation System

Recent practices in precision agriculture include two main micro irrigation methods which promote interesting water efficiency. The first method is the drip irrigation. It allows water to be dripped to the plants roots through pipes containing several emitters. This irrigation system is composed of the following components: water source (generally is a tank) which is connected with a main tube called main pipeline. Several pipes are connected to this main pipe using manual or electrical valves that control the water flow. The pipes go through the field and distribute water for each plant.

The second method is the sprinkler irrigation which delivers water through a pressurized pipe network to the nozzles of sprinkler which spray the water into the air [12]. However, this method is less efficient than the drip one, since more water is losing due to evaporation and runoff. Therefore we choose the drip strategy for our design.

Our proposed model is a closed-loop model. As defined in [13], a system can be categorized as a closed-loop model if the response of the system is monitored and used to adjust the control. We notice that our proposed model is designed for a site-specific irrigation where the crops are characterized by a spatio-temporal variation of the irrigation requirements. The variability comes from the soil type, crop type, crop and meteorological conditions [13]. The main purpose of our design is to handle the dysfunctional situation of the drip installation. As discussed in [4], the crops are negatively affected by human or animals intruders. This is more critical in the case of drip irrigation installation. In fact, the pipes can be broken by rangers or by accident which can cause water waste and plants damage. Also the pipe emitters can be blocked due to environmental condition (sludge, sand) which can causes plant stress. To overtake these shortcomings, the water flow rate into the drip installation must be monitored. For this aim, our proposed system includes the following sensors and actuators:

- Soil moisture sensor: It is used to optimize irrigation and to warn of plant stress by controlling some parameters such as the electrical conductivity of soil or the underground volumetric water content (VWC). Measuring the soil moisture can help the farmers to manage their irrigation systems more efficiently by using less water to grow a crop and increasing quality and yields.
- Temperature sensor: It is used to monitor the ambient temperature. It can be analog or digital and help farmer to adjust their irrigation schedule according the temperature measured to avoid risk of evaporation.
- Pressure sensor: It is used to measure a pressure of gases or liquids and change it into a quantity that can be processed electronically. It generates a signal as a function of the pressure imposed. In irrigation applications, this kind of sensors helps to monitor the abnormal pressure of pipe installation. In such cases, by means of communication module (Zigbee/802.15.4), a message can be transmitted to the corresponding solenoid valve or the master valve (which controls the main pipe) to shut down the system. A very low pressure value can be synonymous of a broken pipe or failure to open valves. Having a high pressure value can indicate that a valve is not closed correctly or some emitters are blocked.
- Solenoid valve: It is an electromechanical valve to use with liquid or gas controlled by running or stopping an electrical current through a solenoid, which is a coil of wire, thus changing the state of the valve [14]. Combined with a Zigbee module, the valve can be controlled through wireless communication. Concerning the energy issue, the valve can have an external energy sources as solar panel.
- Sink node: It corresponds to the gateway of the system. All sensor nodes in the topology need to forward their gathered information to the sink node to be processed. Also, through this node, a request commands are generated to corresponding actuators or sensors.

An illustration of drip irrigation system with a deployment of the WSANs is shown in Fig. 1.

3.1 Deployment Strategy

Deploying the sensor nodes to monitor a farmland is a crucial issue. In fact, many parameters must be considered to choose the most beneficial deployment, as the crops characteristics, the micro meteorological parameters, the sensors and nodes

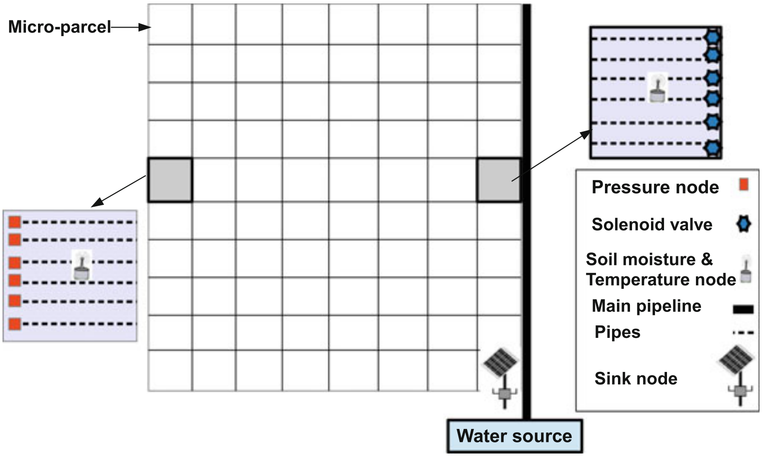


Fig. 1 Drip irrigation system layout

specification and obviously the farmer's budget. According to a generic guide proposed in [15] the coverage of the sensor nodes in agricultural WSN must be dense. By this way, all the required measurements can be gathered to have reliable knowledge of the monitored area. The authors in this guide argue that for 100 m^2 of the field's size, at least 80–90 nodes are needed. In addition to have an adequate number of nodes, the topology formation must be determined. Among start, tree, or grid topology, the right choice depends to field's size and the plants formation. However, for middle or high surface, the grid topology remains the most suitable.

Based on the above discussion, we choose the grid topology for our drip irrigation design. We divide the field area into several equal micro parcel as suggested in [16]. The size of the parcel must be a trade-off between monitoring quality required, the communication coverage and the deployment cost. In the middle of each parcel we fix a soil moisture and temperature node. We make the assumption that the soil moisture and the temperature remain the same inside the parcel.

3.2 Communication Strategy

In Fig. 2 we present a flowchart of the communication between all actors in the designed drip irrigation system. The sensor nodes gather the temperature and the soil moisture from the farmland periodically. According to the value obtained, the sensor nodes decide to send the information to the sink or not. At the sink node, the abnormal information is processed and an eventual decision is taken to adjust the irrigation schedule according to the plant requirement. The same irrigation schedule is transmitted to the pressure nodes to be awakened at the same time of irrigation process. Once the actuators receive an action from the sink, they control their cor-

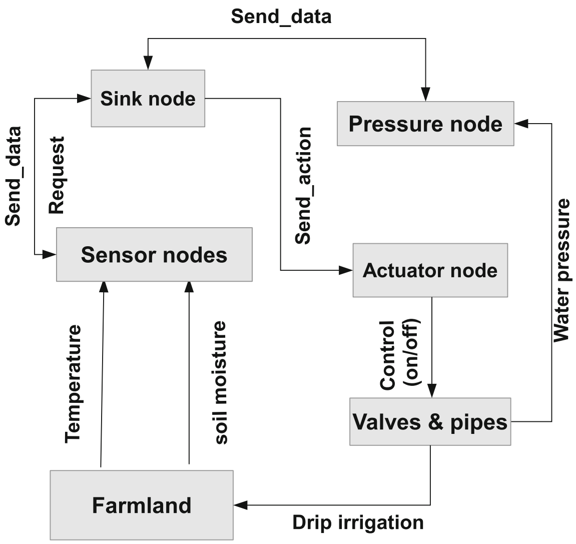


Fig. 2 Drip irrigation system communication

responding valves to be opened or closed. If the valves are opened, the water flow goes through the pipes and the pressure nodes start sensing. If any abnormal pressure value is gathered, an alert message is transmitted to the sink node to shut down the irrigation process and request an external human verification of the pipe installation.

We consider that the sensor nodes communicate only with the sink node through a multi-hop protocol. Also, the actuators receive only actions from the sink. We assume also that the sink node can request some information from the sensor nodes at any time.

As discussed in [17], we present the node's workflow in our system. When the program starts, the sensor nodes are initialized and enter into low power consumption mode to wait for being awakened. At this mode, the processor is in the idle state, but the SPI (Serial Peripheral Interface) port and interrupt system will still being ready to accept system interrupt request. When the scheduled time is reached, the system will transmit signal of acquisition request. Afterwards, the sensor nodes will enter into work mode to collect data. If the collected data value are above or less than predefined thresholds, then the nodes send this data to the sink node. After finishing sending data, the system will return to low power consumption mode. If the collected data don't give a relevant information, the nodes return directly to the idle state. The collecting data process can also be triggered if the nodes receive a special interrupt request from the sink node. In this case, even if the nodes are in low power consumption mode, the MCU will be awakened and enter into work mode. After interrupt returning, the system will return to the idle state again. The figure Fig. 3 summarizes the node's workflow.

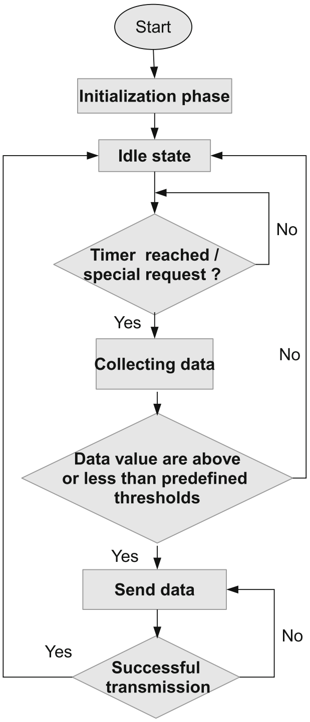


Fig. 3 Node's workflow

3.3 *Priority-Based DIS*

3.3.1 Priority-Based Protocol

As we have discussed, we have two main traffic types gathered from sensors. The first one related to information gathered from temperature and the soil moisture sensors. We classify this traffic type as normal traffic since no need for an urgent intervention is required. The second traffic type is related to information gathered from pressure sensors. We classify this traffic type as priority traffic due to the need for an emergency resolution of the detected problem (shut off the main valve, require human intervention, etc.). Now, in the case when both traffics are active simultaneously, it is clear that the reliability and the timeliness of the priority traffic is more requested than those of the normal traffic.

However, in the wireless context, there are many troubles that can occur due to the sharing of the same communication medium. Among these problems we cite the interference, the exposed and the hidden problem [18]. The other problem that must be considered is the effect of the carrier sense range on communication performances as we have discussed in our previous work [7]. So appropriate routing process must be applied to avoid any trouble between multiple sources and to satisfy the requested QoS for each traffic. In what follows, we will describe how the paths with different traffic priorities can be constructed.

3.3.2 Protocol Description

Our aim is to design a routing protocol that can allow the priority source node (namely the pressure node) to construct an efficient routing path while avoiding the carrier sense range effect. In this work we make the assumption that nodes are aware of their positions and the position of the sink node. In the following, we give a short description of how the paths are constructed according to our approach.

When a priority source node seeks to communicate with the destination, it sets up a route discovery process by sending a forward agent to construct a short multi-hop path. The choice of the next hop node is based on the geographic information available at each node. For each selected node *i*, the node state is changed from free to busy, and a Hello message is broadcasted to all neighbors in the communication range to notify the new state of the node *i*. Every neighbor node *j* of the node *i* becomes a banish node, that means it cannot be selected for any path during the current communication. After that, each node *j* broadcasts in its turn a hello message in their neighborhood. Now, if a normal source node needs to forward data to the sink, it constructs the routing path by respecting the following rule: the next hop must not be blocked, and must not be a banish node or having a banish node in its neighborhood. A node is in a blocked state when the destination is unreachable through this node. To avoid a blocking situation when a node cannot reach the destination, we use the same principle as in [19], called the step-back method. The same method is used by the agent when the selected next hop has a banish node in its neighborhood. Once the destination is reached, the forward agent becomes a backward agent and an optimized reverse path is traversed. At each intermediate node, the agent records the valid next hop into the routing table, after that, it chooses from the reverse path the nearest neighbor to the current node. The same procedure is repeated until reaching the source node. When the communication is ended, all the nodes involved in the communication process reset their state and become ready for further transmissions.

4 Simulation and Result Analysis

We choose using the simulator TOSSIM (Tiny SIMulator) [20] since it allows performing simulations for a large-scale sensor networks while taking into account the physical and link-layer characteristics of WSNs. In addition, the same code used for simulation can be also used for any motes running under TinyOS. In order to evaluate the CSA-MGR protocol over the TOSSIM simulator, it was necessary to add the link reliability to the next hop choosing process. In fact, TOSSIM provides a realistic simulation of the wireless link between nodes. Also, it allows adding the noise, which increases or decreases the link gain. In such setting, considering only the geographic position allows in most cases choosing the next hop with a poor link gain. To avoid such situation, we adjust our protocol as follows:

- Each node record in his internal memory the number of received hello packet from each neighbor during the discovery period.
- Once the discovery period is finished, each node computes his link reliability against all its neighbors. The link reliability is computed as ratio between the number of received hello packet and the number of packet that should be received. As example, if a node *i* generates four hello packets per second during 10 s, then the number of packet that should be received is 40. So, if a node *j* neighbour of *i* receive only 30 packets, then the link reliability from *j* to *i* is 75 %.
- At the routing decision, the next hop chosen is the one that has the link reliability with a ratio above or equal to a predefined threshold. In our simulation, the threshold used is 70 %.

We call this modified version of the CSA-MGR protocol a TinyCSA. We note that in TinyCSA, each source build only one path towards the destination.

4.1 Working Environment

We implemented the tiny version of the CSA-MGR in nesC for the TinyOS operating system. The protocol occupies 7,289 Bytes of RAM, and 25,466B of program memory. We compared the performance of TinyCSA with two state-of-the-art routing protocols, more specifically, TPGF [19] for geographic category, and TinyHop [21] for topologic category. Also, we compare TinyCSA with a version of geographic routing protocol that is based only on the link quality to construct paths without considering the carrier sense approach. We called this version Link-aware protocol.

In TOSSIM, the specific behavior of the wireless link depends on two elements: the radio characteristics and the environment where the nodes are placed. Hence, in order to obtain better simulations, the parameters of both elements should be provided.

The nodes environment forms what we call the communication channel. This channel is modeled using the log-normal path loss model [22]. This model has the following parameters:

- PATH_LOSS_EXPONENT: rate at which signal decays.
- SHADOWING_STANDARD_DEVIATION: randomness of received signal due to multipath.
- D_0: reference distance (usually 1 meter). D_0 also determines the minimum distance allowed between any pair of nodes.
- PL_D_0: power decay in dB for the reference distance D_0.

In our simulation, we choose the above parameters to simulate an agricultural field as given in [22]. Once these parameters are chosen, the tool provided in [22] allows generating a file containing the gain of each link in the topology. In such way, the connectivity map of the network is defined. We make the assumption that the links between nodes are symmetric, i.e. the gain between nodes *i* and *j* is the same as the gain between nodes *j* and *i*.

In addition to the radio propagation model discussed above, TOSSIM also simulates the Radio Frequency noise and interference a node can hear. It uses the Closest Pattern Matching (CPM) algorithm [23]. CPM takes a noise trace as input and generates a statistical model from it. This model can capture bursts of interference and other correlated phenomena which greatly improves the quality of the RF simulation. The noise trace used in our simulation is a shorter version of 200 lines of the Meyer Library given by the Stanford University.

Our scenario is as follows: we deployed 100 nodes in a 100 m * 100 m grid, with an inter-node distance of 10 m. All nodes start a broadcasting period of four packets per second during 10 s to exchange their position information. After that, two sources generate traffic sequentially toward the sink. The first source starts sending immediately after the end of broadcasting period, the second source starts sending one second after. Each source generates **X** packets per second, where:

$$X \in \{8, 16, 24, 32\}.$$

The two sources are located in the middle of the first column, and the sink is located in the middle of the last column. In such way, the constructed paths may have up to 9 hops. The metrics we used for our simulation are the delay and the PDR (Packet Delivery Ratio). The simulation time is 100 s and the results presented in what follow are average of 20 simulations.

Table 1 summarizes the parameters used for simulation.

Table 1 Main configuration parameters for TOSSIM simulation

Parameters	Value
Propagation model	Shadowing
Path loss exponent	4.7
Shadowing deviation (dB)	3.2
Reference distance (m)	2
PL_D_0 (dB)	55.4
Packet size (Byte)	52
MAC layer	IEEE 802.15.4
Frequency (GHz)	2.4

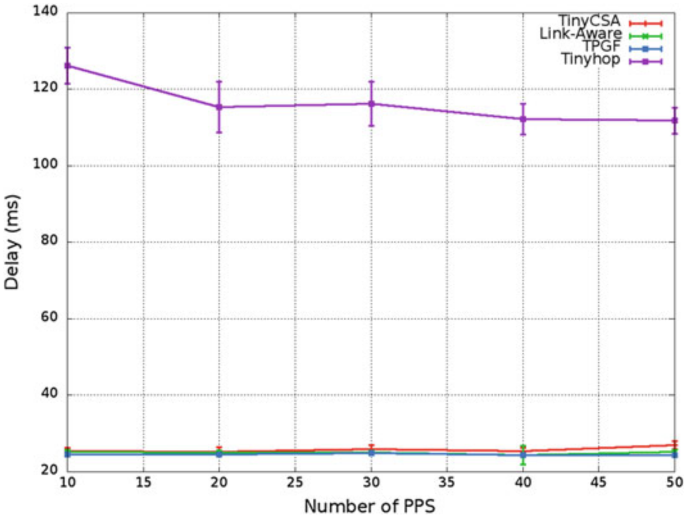


Fig. 4 Average delay versus PPS

4.2 Result Analysis

In Fig. 4, the delay for all protocols is depicted. As a first observation, we can see that the TinyHop has the higher delay compared to the other protocols and this for all values of the number of Packet Per Second (PPS). It is due to the protocol conception, in fact TinyHop behaves as the AODV protocol with in addition an acknowledgement mechanism for each control and data packet. However, enabling acknowledgement for data packet may increase the delay since high data packet retransmission can occur due to collision or inability to access to the channel. This fact is more pronounced when the packet per second transmitted is significant. Concerning TinyCSA, TPGF and Link-Aware protocols, they perform nearly the same delay as we can see in the zoom figure Fig. 5. The delay difference between the three protocols never exceeds 3 ms.

In Fig. 6, the PDR for all protocols is depicted. As a first observation, we can see that the PDR decreases as the number of PPS increases. We see clearly that TinyCSA achieves the high PDR compared to the other protocols. It is expected since the constructed paths for both sources avoid the carrier sense effect. The Link-Aware has the second best PDR result. It can be explained by the fact that the paths are constructed while considering the link reliability as explained above. TPGF protocol performs a low PDR result due to the greedy forwarding approach. In fact, during the path construction process, the next hop chosen is always the one who is closest to the sink. However, choosing such next hop may have, in most cases, a poor link gain with the current node. Which means having poor communication performances. Concerning the TinyHop protocol, it has the lowest PDR result. As explained for the

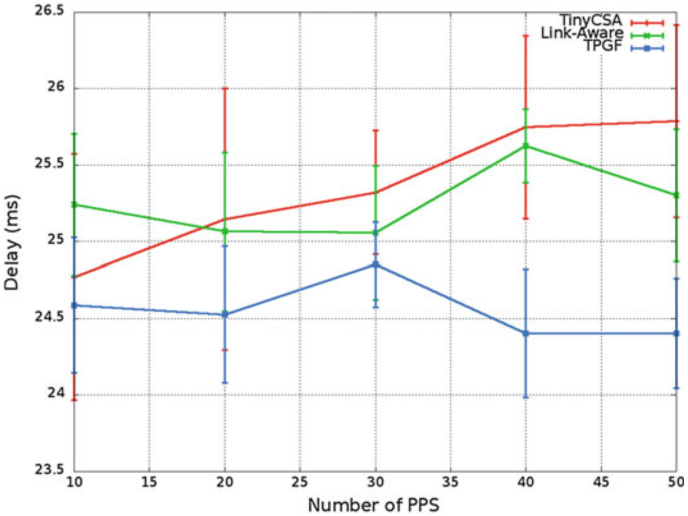


Fig. 5 Zoom average delay versus PPS

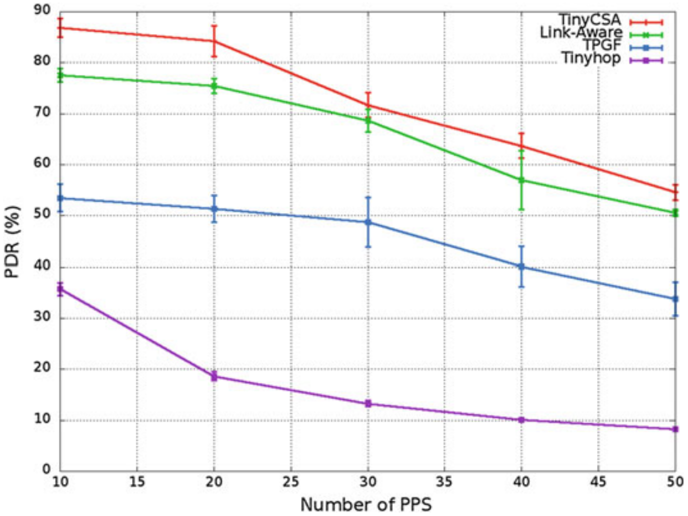


Fig. 6 Average PDR versus PPS

delay metric, TinyHop performs an acknowledgement mechanism for the data packet transfer. However, with significant PPS, the likelihood of data retransmission is more important due to the concurrency to access to the communication channel. Hence, when the number of retransmission of the same packet exceeds three, the packet is discarded.

5 Experimentation

We have implemented the TinyCSA on a TelosB mote running TinyOS operating system. In this section, we discuss design decisions and issues regarding the real-world implementation.

5.1 Experimental Setup

The TinyCSA routing protocol is realized in real test bed consisting of 17 Crossbow TelosB motes deployed as shown in Fig. 7. The code size of TinyCSA protocol under TelosB mote is 23,224 bytes of flash memory and 1210 bytes of RAM. The motes use a 2.4 GHz IEEE 802.15.4 radio interface, where every channel supports up to 250 kbps data rates. The topology is composed of two sources, (mote 1 and 2), 14 intermediate motes and one sink (mote 17). We assign the coordinates of each mote during the compilation phase. We present in Fig. 8 the coordinates system adopted in our experimentation. The inter-node distance is approximately 10 cm since the radio power of the motes is defined at its minimum (i.e. CC2420_DEF_RFPOWER is set to 1, giving a transmit power less than −25 dBm).

We use the lower level of the transmit power for two reasons. First, to set up a small deployed network within a single room or even on a small area of the floor. Second, to allow that each mote has only as neighbors, the motes located directly in front of it. As example, the mote 3 in Fig. 7 may have the motes 7 and 8 as valid neighbors (with a good gain). We note that we have tried several nodes location while testing the next transmission power level in order to get the final connectivity map. Other attempts were resulting in poor connectivity or sometimes too good connec-

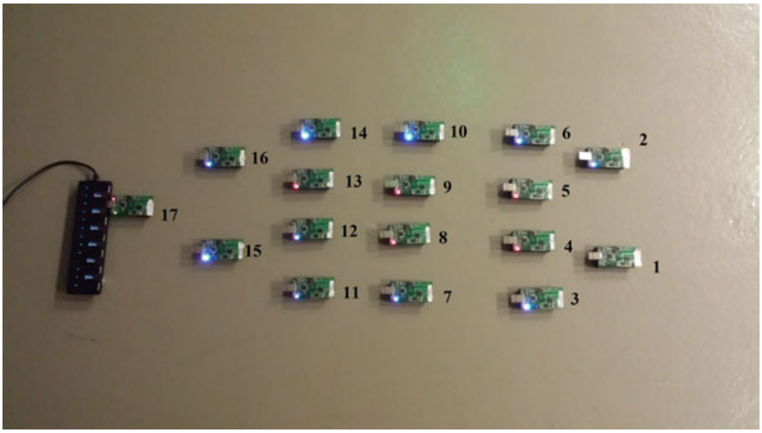


Fig. 7 Experimentation topology

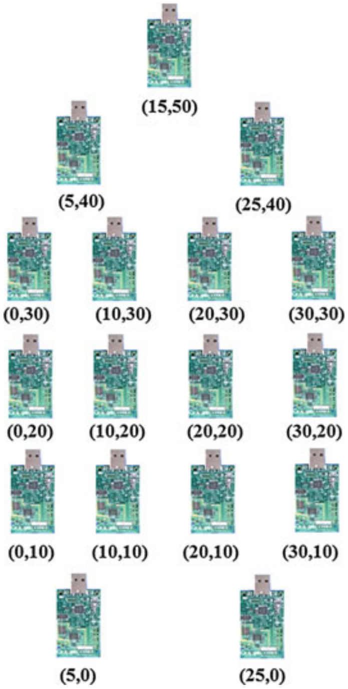


Fig. 8 The motes coordinates

tivity. We present in Table 2 an example of connectivity map obtained during our experimentation when the link reliability is above 70 %.

The sink mote is powered via USB cable connected to a main PC running Ubuntu 14.04 with AMD Processor and 4 GBytes of RAM. All motes are programmed through the USB port. We use a USB hub consisting of 13 ports and with a shell script we can compile many motes at the same time. We use an additional mote as a remote control to ensure that all motes in the network start at the same time. This mote is programmed with a nesC code allowing broadcasting a special packet with the high transmit power (i.e. CC2420_DEF_RFPOWER is set to 31). Each mote that receives this special packet makes a reset system. We debugged our work on the central PC by allowing motes to send debug messages through USB interfaces. Although, even if the test-bed setting is for a small-scale sensor network, the experiment with this configuration is valuable as it reveals the protocol behaviour in a real hardware setting.

By means of the cutecom tool [24], we record in a trace file the test bed performance in term of packet delivery ratio and average packet delay from the source to the destination. The results of the TinyCSA are compared with those of the Link-Aware protocol.

Table 2 Connectivity map of the experimentation

Mote ID	Reachable neighbor with link reliability >70 %
1	3,4
2	5,6
3	1,4,7,8
4	1,3,5,7,8,9
5	2,4,6,8,9,10
6	2,5,9,10
7	3,4,8,11,12
8	3,4,5,7,9,11,12,13
9	4,5,6,8,10,12,13,14
10	5,9,13,14
11	7,8,12,15
12	7,8,11,15
13	8,9,10,12,14,16
14	9,10,13,16
15	11,12,17
16	13,14,17
17	15,16

5.2 *Experimental Scenario*

The experimentation scenario is as follows: After receiving the reset command from the remote control, all motes start a discovery period of 15s. Once finished, the mote number 2 start sending periodic data packet. After one second, the mote number 1 start also sending data. The data rate used in our experimentation is **X** packets per second, where:

$$X \in \{5, 10, 15, 25, 30\}.$$

Due to the limit number of mote in our possession, we make the first constructed path as static path. In fact, since the TinyCSA is a geographic protocol, the mote 2 will choose the middle motes in the network to construct path. As example, the path $2- > 5- > 9- > 12- > 15- > 17$ may be constructed. In such case, the second source will note find enough valid motes to construct path due to the banish state. To avoid such situation, we fix the path from mote 2 to mote 17 to be as follows: $2- > 6- > 10- > 14- > 16- > 17$.

We compare TinyCSA with only the Link-aware protocol since the TPGF and TinyHop have shown poor results for the simulation scenario. We note that the same configuration is applied for the Link-Aware protocol. The experimentation time is 50 s, after that, all motes are reset automatically by means of scheduled task. We repeat the experimentation 10 times and the average results are presented in figures below.

5.3 Experimental results

We start our analyse by Fig. 9. We can see that TinyCSA has the lowest delay compared to the Link-aware protocol. The difference is about 15 ms. We can see also that the delay increases as the number of packet per second increases. Compared to the simulation result obtained by TOSSIM, we see clearly that the end-to-end delay in the real test bed is higher compared to the simulation case. It can be explained by the fact that the processing delay for the TelosB microprocessor is very slow compared to the laptop processing in the simulation. The microprocessor for TelosB runs at 8 MHz while the processor in simulation runs at 2400 MHz which is nearly 300 times faster. In addition, this difference can be due to communication links failures in wireless networks which causes retransmissions of the packet at the MAC layer and thereby increasing the average delay.

In Fig. 10, the PDR for both protocols is depicted. As first remark, we can see that the PDR decreases as the number of PPS increases. It is expected since more the data rate is important, more the motes fail to accede to the channel due to concurrency, which increase the likelihood of packet lost. Furthermore, we see clearly that the TinyCSA protocol has the highest average PDR for all PPS values. The gain compared to the Link-Aware protocol is up to 15 %. Compared to the simulation result, we can see that both protocols have nearly the same average PDR in real test bed as well as in simulation scenario.

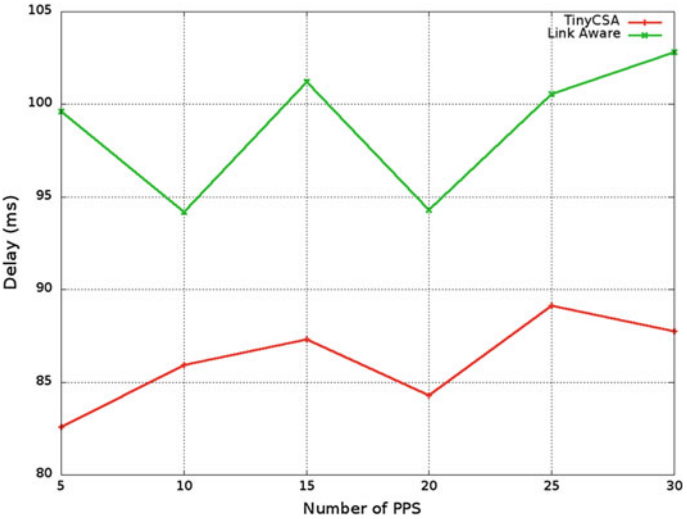


Fig. 9 Average delay versus PPS

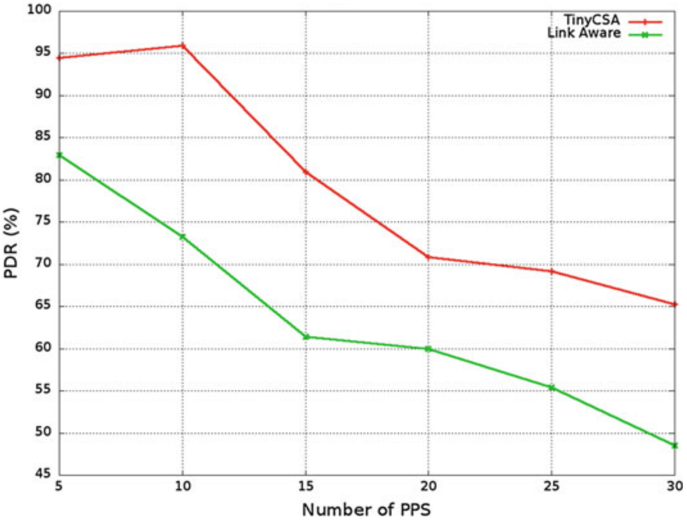


Fig. 10 Average PDR versus PPS

6 Conclusion

In this paper, we have presented a model architecture for a drip irrigation system using the WSANs. Our model includes the soil moisture, temperature and pressure sensors to monitor the irrigation operations. Specially, we take into account the case where a system malfunction occurs, as when the pipes are broken or the emitters are blocked. Also, we differentiate two main traffic levels for the information transmitted by the WSAN, and based on the CSA-MGR protocol, we achieve a high QoS performance.

We have performed extensive simulations through TOSSIM simulator. The results prove that our solution gives better performances in terms of delay, PDR for the priority traffic. Also we have realized a real test-bed to investigate the effectiveness of our approach. The experimentation results show considerable gain compared to other state-of-the-art protocol. As future work, we intend to move towards real implementation in agricultural fields.

References

1. Camilli, A., Cugnasca, C.E., Saraiva, A.M., Hirakawa, A.R., Correa, P.L.P.: From wireless sensors to field mapping: anatomy of an application for precision agriculture. Comput. Electron. Agric. **58**(1), 25–36 (2007). Precision Agriculture in Latin America
2. Wang, N., Zhang, N., Wang, M.: Wireless sensors in agriculture and food industry recent development and future perspective. Comput. Electron. Agric. **50**(1), 1–14 (2006)

3. Riquelme, J.A.L., Soto, F., Suardiaz, J., Sanchez, P., Iborra, A., Vera, J.A.: Wireless sensor networks for precision horticulture in Southern Spain. Comput. Electron. Agric. **68**(1), 25–35 (2009)
4. Garcia-Sanchez, A.-J., Garcia-Sanchez, F., Garcia-Haro, J.: Wireless sensor network deployment for integrating video-surveillance and data-monitoring in precision agriculture over distributed crops. Comput. Electron. Agric. **75**(2), 288–303 (2011)
5. Akyildiz, I.F., Stuntebeck, E.P.: Wireless underground sensor networks: research challenges. Ad Hoc Netw. **4**(6), 669–686 (2006)
6. Xiaoqing, Y., Pute, W., Han, W., Zhang, Z.: A survey on wireless sensor network infrastructure for agriculture. Comput. Stand. Interfaces **35**(1), 59–64 (2013)
7. Bennis, I., Fouchal, H., Zytoune, O., Aboutajdine, D.: Carrier sense range effect on multipath routing performances in wireless sensor networks. In: Federated Conference on Computer Science and Information Systems, pp. 1087–1092 (2014)
8. Bennis, I., Fouchal, H., Zytoune, O., Aboutajdine, D.: Drip irrigation system using wireless sensor networks. In: 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1297–1302 (2015)
9. Sudha, M.N., Valarmathi, M.L., Babu, A.S.: Energy efficient data transmission in automatic irrigation system using wireless sensor networks. Comput. Electron. Agric. **78**(2), 215–221 (2011)
10. Gutierrez, J., Villa-Medina, J.F., Nieto-Garibay, A., Porta-Gandara, M.A.: Automated irrigation system using a wireless sensor network and gprs module. IEEE Trans. Instrum. Measur. **63**(1), 166–176 (2014)
11. Mafuta, M., Zennaro, M., Bagula, A.B., Ault, G.W., Gombachika, H.S.H., Chadza, T.: Successful deployment of a wireless sensor network for precision agriculture in malawi. IJDSN **2013** (2013)
12. Ali, H.: Practices of Irrigation & On-farm Water Management: Volume 2. Springer, New York (2011)
13. McCarthy, A.C., Hancock, N.H., Raine, S.R.: Advanced process control of irrigation: the current state and an analysis to aid future development. Irrigation Sci. **31**(3), 183–192 (2013)
14. Haller, J.J., Glaudel, S.P., Volz, G.J.: System and method of operating a solenoid valve at minimum power levels, 26 Sept 2013. US Patent App. 13/900,683
15. Mampentzidou, I., Karapistoli, E., Economides, A.A.: Basic guidelines for deploying wireless sensor networks in agriculture. In: 2012 4th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), pp. 864–869 (2012)
16. An, W., Ci, S., Luo, H., Wu, D., Adamchuk, V., Sharif, H., Wang, X., Tang, H.: Effective sensor deployment based on field information coverage in precision agriculture. Wirel. Commun. Mob. Comput. **15**(12), 1606–1620 (2013)
17. Song, J., Zhu, Y., Dong, F.: Automatic monitoring system for coal mine safety based on wireless sensor network. In: Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC), 2011, vol. 2, pp. 933–936 (2011)
18. Wang, L., Kaishun, W., Hamdi, M.: Combating hidden and exposed terminal problems in wireless networks. IEEE Trans. Wirel. Commun. **11**(11), 4204–4213 (2012)
19. Shu, L., Zhang, Y., Yang, L.T., Wang, Y., Hauswirth, M., Xiong, N.: TPGF: geographic routing in wireless multimedia sensor networks. Telecommun. Syst. **44**(1–2), 79–95 (2010)
20. http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM/. Accessed July 2015
21. Simón Carbajo, R., Huggard, M., McGoldrick, C.: An end-to-end routing protocol for peer-to-peer communication in wireless sensor networks. In: Proceedings of the 6th Workshop on Middleware for Network Eccentric and Mobile Applications, MiNEMA'08, pp. 5–9 (2008). ACM, New York, NY, USA
22. http://www.tinyos.net/tinyos-2.x/doc/html/tutorial/usc-topologies.html/. Accessed July 2015
23. Lee, H.J., Cerpa, A., Levis, P.: Improving wireless simulation through noise modeling. In: 6th International Symposium on Information Processing in Sensor Networks, 2007. IPSN 2007, pp. 21–30 (2007)
24. http://cutecom.sourceforge.net/. Accessed July 2015

BARBEI: A New Adaptive Battery Aware and Reliable Beacon Enabled Technique for IEEE 802.15.4 MAC

Marwa Salayma, Ahmed Al-Dubai, Imed Romdhani and Muneer Bani Yassein

Abstract The IEEE 802.15.4 standard supports both physical and Media Access Control (MAC) layers of low rate Wireless Sensor Networks (WSNs). However, this standard considers only the ideal linear power consumption, omitting the fact that the diffusion principle in batteries shows the nonlinear process when it releases a charge. Thus, this study proposes a technique that improves the performance of the IEEE 802.15.4 standard by allowing its MAC to exploit the nonlinear processes of the battery to prolong the WSN lifetime. To the best of our knowledge, the proposed technique is the first that considers both the battery status and network latency status through a cross layer model. The performance of the new algorithm has been examined and compared against that of the legacy IEEE 802.15.4 MAC algorithm through extensive simulation experiments. The results show that the new technique reduces significantly the energy consumption and the average End-to-End delay as well.

Keywords WSN · IEEE 802.15.4 · MAC · Rakhmatov · Battery aware · Delay · Duty cycle

M. Salayma (✉) · A. Al-Dubai · I. Romdhani
School of Computing, Edinburgh Napier University, Edinburgh, UK
e-mail: M.Salayma@napier.ac.uk

A. Al-Dubai
e-mail: A.AL-Dubai@napier.ac.uk

I. Romdhani
e-mail: I.Romdhani@napier.ac.uk

M.B. Yassein
Department of Computer Science, Jordan University of Science and Technology,
Irbid, Jordan
e-mail: masadeh@just.edu.jo

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent
Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_18

1 Introduction

This work is an extended version of [1]. In Wireless Sensor Networks (WSNs), energy conservation is one of the main concerns challenging the Cutting-Edge standards and protocols [2, 3]. Most existing studies focus on the design of WSN energy efficient algorithms and standards. The standard IEEE 802.15.4 has emerged for WSNs in which the legacy operations are based on the principle that the Power-Operated battery is ideal and linear. However, the diffusion principle in batteries shows the nonlinear process when it releases a charge [4]. Hence, we can prolong the network lifetime by designing optimized algorithms that reflect the battery characteristics. Within this context, this paper proposes a Cross-Layer algorithm to improve the performance of beacon enabled IEEE 802.15.4 network by allowing a Personal Area Network Coordinator (PANc) to tune its MAC behavior adaptively according to both the current remaining battery capacity and the network status. In order to gain a better understanding of the proposed technique, it is important to have an insight at the three concepts that make up our proposed protocol, and which explain the importance of considering such approach when it comes to improving the WSN lifetime. For this purpose, the following three subsections present a general overview about WSNs, IEEE 802.15.4 and the battery. The rest of the paper is organized as follows: Sect. 2 summarises some of the literature work which is closely related to the paper topic. Section 3 presents the motivation behind tackling this work, while Sect. 4 illustrates the methodology adopted for achieving our protocol. Section 5 depicts the details of the proposed approach. In Sect. 6 we evaluate and discuss the performance of our proposed protocol. Section 7 concludes the paper and outlines future work.

1.1 Wireless Sensor Networks (WSNs)

Recently, most wired sensors are being replaced with wireless ones creating the emerging era of WSNs. WSNs consist of sensing devices that can communicate with each other and with the surrounding environment via the wireless communication medium [2, 3]. Yet, a huge number of sensor nodes are often scattered in unreachable areas, and WSNs are often battery powered and cannot be easily recharged. Thus, energy conservation is one of the main concerns in the area of WSN. Many studies that focus on designing WSN energy efficient algorithms and standards, based on IEEE 802.15.4, have emerged recently [5].

1.2 IEEE 802.15.4 Standard

The IEEE 802.15.4 standard supports both physical and MAC layers of low rate WSNs. IEEE 802.15.4 MAC supports two types of devices, namely, Full Functional Devices (FFDs) and Reduced Functional Devices (RFDs). Both FFDs and RFDs communicate with each other, forming two types of topologies, star and peer to peer topologies. The IEEE 802.15.4 MAC operates either in beacon enabled or beaconless modes. In the beacon enabled mode, the FFD broadcasts regular beacon frames to synchronise nodes when they need to access the channel. The time between two successive beacons is referred to as the Beacon Interval (BI), which is divided virtually into 16 equal sized slots. BI duration is specified by the Beacon Order parameter (BO) according to the following formula [6].

$$\mathrm{BI} = \mathrm{aBaseSuperframeDuration} * 2^{\mathrm{BO}} \tag{1}$$

Nodes can use the channel during the whole BI period or can sleep for some time portions depending on Superframe Order (SO) parameter. This parameter decides the Superframe Duration (SD) active session according to the following formula.

$$\mathrm{SD} = \mathrm{aBaseSuperframeDuration} * 2^{\mathrm{SO}} \tag{2}$$

where $0 \leq \mathrm{SO} \leq \mathrm{BO} \leq 14$.

According to (3), the value of aBaseSuperframeDuration depends on the slot duration. For 2.4 GHz Radio Frequency (RF) band, the value of each slot duration equals 60 symbols where each symbol equals 16 µs.

$$\mathrm{aBaseSuperframeDuration} = \mathrm{aBaseslotDuration} * \mathrm{total\ slots} \tag{3}$$

All these concepts can actually be indicated through the duty cycle (D). This is the percentage of time the node is awake from the whole time between the two successive beacons. D is mathematically expressed as [7, 8]:

$$\mathrm{D} = \mathrm{SD}/\mathrm{BI} * 100\,[\%] \tag{4}$$

When a node needs to access the medium, it has to locate the beginning of the next time slot in order to compete for the channel based on the Carrier Sense Multiple Access/Collision Avoidance algorithm (CSMA/CA). This time portion is referred to as the Contention Access Period (CAP) [6–8]. Furthermore, the standard gives PANc the authority to assign a number of slots to some nodes exclusively. The optional period which includes those slots is referred to as the Contention Free period (CFP). The time period that consists of CAP and CFP is called the active period and the time portion that remains in the superframe (if enabled) is called the inactive period. IEEE 802.15.4 superframe structure is depicted in Fig. 1.

The lengths of the discussed periods are assigned through the beacon frame, which is transmitted in the first time slot (slot 0). Due to the complicated issues of

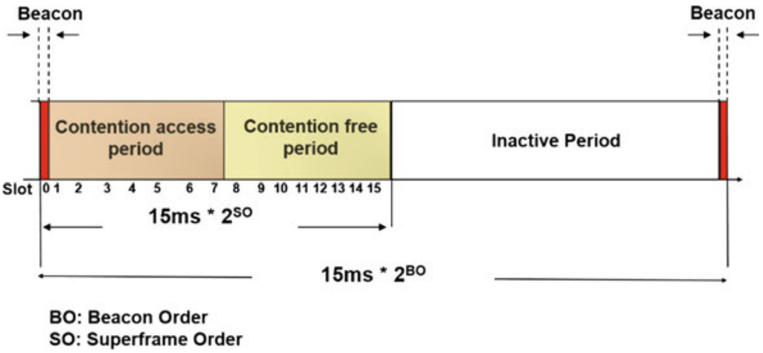


Fig. 1 IEEE 802.15.4 MAC superframe structure

the inactive period, most beacon enabled IEEE 802.15.4 studies are limited to star One-Hop topology. Similarly, this paper considers the same assumption.

BO and SO values control the performance of beacon enabled IEEE 802.15.4. Small BO values increase beacon overhead, which in return, drain more power in a short period of time. Small SO values, on the other hand, decrease nodes active time, while increasing the sleep time period, which will increase delay and adversely affect throughput [7, 8]. Beacon overhead, collision and packets retransmission are all reasons for early battery charge depletion. In order to maximise node lifetime, we need to increase battery lifetime. To achieve that, we need to analyse the battery behaviour and study how it copes with IEEE 802.15.4 operations. To do so, let us first have a general idea about the battery operations and what is going on inside the battery.

1.3 Battery Interior

The battery is comprised of multiple Electro-chemical cells. Figure 2 shows the interior side of a battery cell. Each cell consists of two electrodes; the negative anode and the positive cathode which are separated by an electrolyte. In order to provide energy, the Electro-chemical materials around the anode diffuse towards the electrolyte and are accepted by the cathode. During the recharging process, the Electro-chemical process works vice versa [9]. However, if the battery is allowed to relax for some time portion, it is able to gain some of its lost charges [4, 9]. This phenomena is called battery recovery effect and explained in more details in the following subsection.

Battery recovery effect. This effect is best explained through Fig. 3. Figure 3a shows how the battery looks like when it is fully charged. The small orange balls resemble the Electro-active species around the electrodes. Battery delivers power based on the Electro-chemical reactions that occur between the electrodes and the

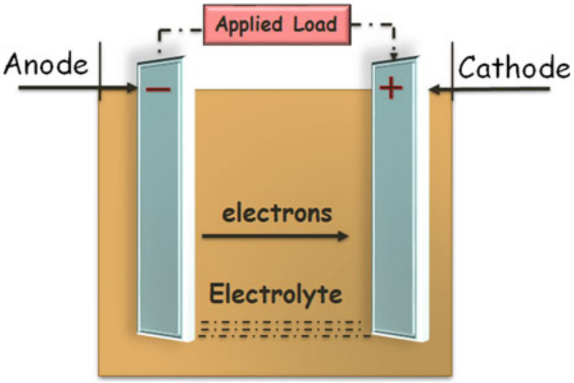


Fig. 2 Battery cell interior

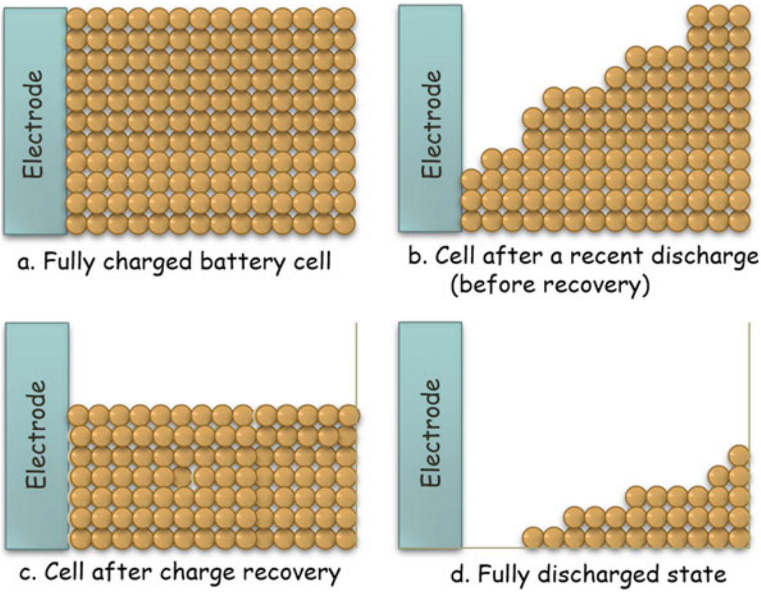

Electrode
a. Fully charged battery cell
Electrode
b. Cell after a recent discharge (before recovery)
Electrode
c. Cell after charge recovery
Electrode
d. Fully discharged state

Fig. 3 Battery recovery effect

active material around the electrodes [4, 9, 10]. Figure 3b depicts the battery cell during the discharging process. Continuous discharging causes continuous depletion of active chemical mass near the electrode. The continuous Electro-chemical reactions cause concentration gradient inside the battery cell. Figure 3c shows that allowing the battery to relax for some time portion gives the chemical material an opportunity to diffuse towards the electrodes by organising themselves more uniformly around the electrode, which allows the battery to replenish some of the lost charges. This nonlinear behaviour is called the battery recovery effect and the idle time is used as recovery time [10]. Figure 3d depicts the final state of the battery when it is about to fully discharged. Accordingly, one can conclude that battery lifetime depends on the usage pattern of the battery [4], and so if we want to estimate battery lifetime, battery recovery effect must be included in our consideration.

Battery linearity. IEEE 802.15.4 standard, as many of other protocols and standards, does not consider battery recovery effect in its communication mechanisms. It does however, consider what is called the "ideal unrealistic" behaviour of the battery. The ideal battery behaviour assumes that the voltage stays constant over time until the moment it is completely discharged, which is the moment the voltage drops to zero, whereas the capacity is the same for all loads that the battery generates [4, 9]. In other word, in its mechanisms, IEEE 802.15.4 considers the linear relationship between the lifetime of the battery and the capacity of the battery as what is shown in (5).

$$C = I \times L \tag{5}$$

Where L is the lifetime of the battery (hour), I is the constant load provided (Ampere) and C is the total battery capacity (Ampere-Hour) [4]. One can notice from (5) that there is no assumption or consideration regarding to the usage pattern of the battery and which results in battery recovery effect. Yet, (5) tells us that the lifetime of the battery can be easily calculated by dividing the total battery capacity (C) over the discharging current (I). Accordingly, the total energy E in the ideal case can be calculated as follows [11].

$$E = V \times C \tag{6}$$

Where E (Watt-Hour) is the provided energy and V is the voltage (volt), C is the total capacity of the battery. However, the realty is totally different. In fact, due to battery recovery effect, the diffusion process of chemical materials around the electrode shows a nonlinear process when the battery releases charges [10]. Battery recovery effect is not only the case, actually there is another contributor to battery nonlinearity, which is the rate capacity effect. This effect says that the battery voltage drops gradually during the discharging process and the capacity is lower for high discharge currents because the voltage drops faster with high current loads applied [4, 9]. Thus, due to rate capacity effect and battery recovery effects, (5), (6) do not hold if one want to consider battery real nonlinear characteristics. Accordingly, for the aim of prolonging network lifetime, battery recovery effect can be exploited by the IEEE 802.15.4 standard by allowing it to duty cycle the nodes adaptively according to the internal status of the node battery. In this regard, we propose an adaptive and cross layer approach that improves the beacon enabled IEEE 802.15.4 performance by allowing the MAC layer to tune its parameters according to the battery behaviour of the coordinator as well as the network status in a star topology. The contribution of this paper is fourfold:

- The real behaviour of the battery in a beacon enabled IEEE 802.15.4 MAC is investigated by considering battery nonlinearity by analysing the diffusion of chemical reactions in the battery following Rakhmatov model.
- The gain of the battery recovery effect according to what sleep period can increase battery life time of the beacon enabled IEEE 802.15.4 is analysed.
- A Cross-Layer and adaptive battery aware beacon enabled IEEE 802.15.4 MAC that tunes synchronization time according to current battery status is proposed.
- The network reliability is considered by checking network delay and tune nodes active period accordingly.

2 Related Work

Recently, there has been significant amount of studies that addressed the Electro-chemical behaviour of batteries. Li et al. [12] proposed an analytical model that computes the life time of a low duty cycled star sensor network. In their model they considered nonlinearities of Lithium-Ion battery following Rakhmatov model [9] and they aimed to minimize the total energy consumption of the Lithium-Ion battery by finding the optimal idle and sleep period while guaranteeing energy efficiency, reliability and reasonable latency. In their proposal, they considered the Trade-Off between energy that is dissipated in sending frequent preambles and the period thereby sensors stay idle waiting for the preamble. Experimental results show that the proposed method can provide the optimal sleep or channel check intervals that maximize the lifetime of the network while guaranteeing a little latency and high reliability. However, this model targets only a simplified work mechanism of MAC protocol, without giving details of battery nonlinearity effects on the proposed protocol on their network.

Li et al. [10] presented three battery aware algorithms that reduce power consumption and extend battery lifetime. Each one of the proposed schemes is targeted towards a specific application type, which are the hard Real-Time applications, the soft Real-Time applications, and the periodic applications. Li et al. [10] follow Rakhmatov model to depict both battery and recovery effect and nonlinearity. Simulation results demonstrated that the three Battery-Friendly algorithms perform better in extending lifetime of Battery-Operated sensor nodes as they reduce battery charge consumption. Actually, the work stated in [10] adopts the accurate form of battery model that was already proposed in [13]. Li et al. [13] proposed three battery friendly transmission policies for extending battery lifetime by considering certain delay constraints. In the first scheme, multiple packets are combined in order to exploit battery charge recovery effect during longer idle periods. The second mechanism is battery efficient as it draws smoother and lower current by adopting modified lazy packet scheduling. The third scheme is a hybrid of the two mentioned schemes. The three schemes are simulated for a wireless network with internet traffic and are compared according to average delay and battery performance.

Results revealed that the more packets are combined in the first scheme, the better the battery performance; this is achieved at the expense of a larger buffer size and higher system delay. The second scheme is more battery efficient and has lower runtime complexity. The last scheme proved to have lowest battery charge consumption at the expense of higher average packet delay. Delay analysis shows that as the packet arrival rate reduces, the average delay per packet of the first scheme varies slightly, while it increases in the second and third schemes. Thus, packet delay is the main drawback of these schemes.

Chau et al.

[14] studied the gain at which the battery recovery effect prolongs commercial sensors lifetime empirically. This effect has also been studied analytically corroborated by simulation. The outcome of [14] revealed that there is a saturation threshold at which the battery recovery will contribute in improving the behaviour of the battery. Authors in [14] proposed a distributed battery aware duty cycle protocol and measured the battery runtime under both deterministic and randomised schedules. The authors in [14] studied also the Trade-Off between both delay and harnessing the recovery effect. The same authors proposed a more Energy-Efficient algorithm, that is aware of battery recovery effect by extending the Pseudo-random duty cycling scheme proposed in [14] by a forced sleep. In addition, the authors in [15] achieved analytical results that predict the average delay in sensor networks by setting the sleep duration of the RF transceiver at the saturation threshold of the battery, which can take the maximal advantage of the Duration-Dependent battery recovery effect.

Casilari et al. [16] proposed an analytical model that forecasts the minimum, mean and maximum battery lifetime of a WSN by allowing it to work under different traffic loads, data rates and probability of packet loss. This is done by an experimental characterization of activity cycles battery consumption in commercial motes that follows the 802.15.4/ZigBee stack and also by measuring the current that is drained from the power source under different 802.15.4 communication operations [16]. The characterization considers the different operations required by 802.15.4 protocol and takes into consideration the delay introduced by the CSMA/CA algorithm applied by the 802.15.4 MAC layer. The model has also been extended to cope with the extra consumption that the node Re-association requires when a packet loss occurs.

Mario et al. [17] proposed an adaptive and Cross-Layer Energy-Aware module for Energy-Efficient and reliable data collection targeted towards IEEE 802.15.4/ZigBee WSNs. The proposed module captures the packet delivery ratio at the application and configures the MAC layer parameters, which are backoff window size and the number of retransmissions according to the traffic conditions in order to minimize the power consumption. [17] Addressed the Trade-Off between energy efficiency and reliability, while satisfying the application requirements. To achieve that, [17] proposed an Adaptive Access Parameters Tuning (ADAPT) method that consumes low energy and low latency for both Single-Hop and Multi-hop networking scenarios while meeting the Application-Specific reliability requirements under a wide range of operating conditions. Simulation results showed that ADAPT is very Energy-Efficient, with Near-Optimal performance.

Nasralah et al. [18] proposed two battery aware MAC protocols for prolonging WSN life time. Authors in [18] consider the remaining voltage as a criteria through which nodes decide the duration of their backoff period in CSMA/CA mechanism. To predict battery capacity, Authors in [18] follow Markov stochastic model [18] compares the results with legacy CSMA/CA mechanism in terms of energy consumption, throughput, percentage of collision and the recovered energy. The proposed mechanisms outperform the legacy CSMA/CA in terms of the tested metrics. Experimental results in [18] revealed that the battery recovery has greater effect in batteries with higher capacity. However, the procedure only targets the backoff period in CSMA/CA algorithm in IEEE 802.15.4 MAC without giving details about the implemented topology. So it is better to be considered as a new battery aware backoff algorithm.

Khan et al. [19] suggested two critical Wireless Body Area Network (WBAN) design issues of an implanted node, which are the energy consumption and transmission power profile. They analysed the lifetime of a WBAN node's battery for different physiological data collection, MAC protocols and transmission channel conditions. In order to simulate battery behavior and transmission channel, the authors followed Peukert's Law and a KiBaM battery models through MATLAB based simulation. They examined battery life cycle of different WBAN sensors, such as heart rate, ECG and body temperature since they transmit packets using different time schedules at different sample rate. For each physiological data, they used three different MAC protocols to measure the battery lifetime. The simulation model initially consisted of battery models and MAC protocols, then the authors extended the model to examine the power transmission profile of an implanted node by following two different transmitter powers to obtain the PER profiles. The retransmission power requirement depends on the followed MAC protocol [19].

3 Motivation

According to the reviewed literature, it is realised that despite there are some efforts that consider nonlinear behaviour of the battery in WSN, still there is no a standardised one. IEEE 802.15.4 for example, provides mechanisms to achieve an Ultra-low power WSN, yet it assumes battery ideal characteristics. However, battery recovery effect tells us that minimising energy consumption of a system does not necessarily mean maximising battery lifetime, yet due to battery nonlinear effects, saving energy depends so much on the usage pattern of the battery.

Battery recovery effect can be exploited by the standard to help in a proper duty cycling of nodes, which will increase both battery and network life time. This can be achieved by proposing optimization algorithms designed according to the battery's characteristics, e.g., by inserting a relaxation time between the two packets for greater charge capacity to prolong the lifetime of network.

4 Methodology

In order to exploit battery characteristics in our protocol, we need to study its Electro-chemical behaviour, which can be analysed empirically or through models. Empirical analysis is time consuming and requires expensive prototyping and measurement for each alternative. Therefore, battery behaviour under various conditions of charge/discharge can be predicted through models [4, 9, 10, 13–15]. Models for energy consumption and performance estimation of each system component are described in the following Sub-section.

4.1 Battery Model

There are different models that describe the battery discharge processes. Each model type has a varying degree of accuracy and complexity. Those models can be classified as low level Electro-chemical models and high level mathematical models. Electro-chemical models are the least flexible and the most computation intensive, so they are sophisticated models to use for battery modeling and they are the most accurate ones. On the other hand, electrical circuit models, analytical models and the stochastic models can be easily configured for different types of batteries. Electrical circuit models still computationally intensive and they ignore the effects of charge recovery during idle periods. The stochastic models can be used efficiently for simulation and are capable of modeling rate capacity and recovery effects [4, 12]. Analytical models are computationally efficient, but limited in the discharge effects they model. One of these models is Sarma and Rakhmatov model which is an abstraction of a real battery [9] that we used in this work. Rakhmatov model is chosen for estimating the real residual battery capacity at a specific time, because it is the simplest accurate analytical model. Other models require solving complex Partial Differential Equations (PDEs) which are difficult to optimize [4, 9]. For the model to adequately mimic real behaviour of the batteries, one can utilise this formula:

$$\alpha = I\left[L + 2\sum_{m=1}^{\infty}\frac{1 - e^{-\beta^2 m^2 L}}{\beta^2 m^2}\right] \tag{7}$$

where I is the applied load and L is battery lifetime, α is the capacity of the battery when it is fully charged, β refers to battery materials diffusion around the electrolyte and measures the nonlinearity of the battery as it tells us how fast the diffusion process can keep up with the rate of discharge. The value of α is a battery related parameter and its value is decided by manufacture of battery designer. Formula (7) indicates that the total capacity of the battery is the sum of two terms, the linear ideal behaviour plus the nonlinear behaviour. As long as β value is large, the battery behaviour becomes closer to battery ideal effect. When β goes to infinity, the battery works in its ideal situation (linear behavior). This means that the higher the value of β, the better the battery performs. The value of β is estimated from the data sheet of the battery [9, 12]. For example, the data sheet of a battery might model rated capacity (in Ah) versus discharge current (in hour). Thus, before one can use the proposed model, the parameters need to be estimated from experimental data for the modeled battery. Simple experiments with constant loads are sufficient for estimation purposes. However, choosing the optimised values for both α and β is beyond the scope of this paper and for more details about it please refer to [9].

It is important to note that the current is discharged according to different transceiver activities (transmission, receive and idle), each has its own time duration. Thus, the load can be depicted in the form of consecutive N constant current values $I_1, I_2, I_3, \ldots, I_N$, where I_K is the current of activity k which took place at time t_k in the duration of $\Delta k = t_{k+1} - t_k$ [9, 10]. Accordingly, battery capacity when it is fully charged can be depicted as follows:

$$\alpha = \sum_{k=1}^{N} I_k \Delta_k + \sum_{k=1}^{N} 2 I_k \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 (L - t_k - \Delta_k)} - e^{-\beta^2 m^2 (L - t_k)}}{\beta^2 m^2} \tag{8}$$

In order to calculate the remaining capacity at a specific time unit, we need first to calculate the amount of charge consumed after performing M activities (charge lost from the battery) which is denoted by σ as follows [9, 10]:

$$\sigma(t) = \sum_{k=1}^{M} I_k \Delta_k + \sum_{k=1}^{M} 2 I_k \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 t} \left(e^{\beta^2 m^2 \Delta_k} - 1 \right)}{\beta^2 m^2} e^{-\beta^2 m^2 t_k} \tag{9}$$

According to (8) and (9), the residual capacity at a specific time t (the available charge) is presented here:

$$\alpha(t) = \alpha - \sigma(t) \tag{10}$$

One might conclude from Fig. 3 that in order to gain the best of the battery recovery effect, one should relax the battery as much as possible. However, this is not the case. The following subsection clarifies that there is a threshold at which battery recovery can be exploited, which if it is not considered, will cause extra charge consumption from the battery.

4.2 Battery Recovery Threshold (Influence of Injecting Sleep Period)

Figure 4 clarifies the effect of injecting an inactive state for achieving the battery recovery effect purpose. Figure 4a shows that if the battery is not given an opportunity to relax, then it will not recover charges and there will be continuous

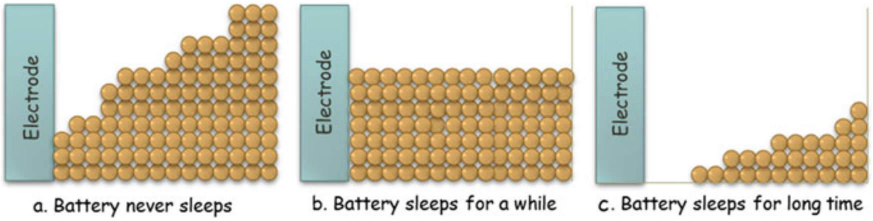

Electrode
Electrode
Electrode
a. Battery never sleeps
b. Battery sleeps for a while
c. Battery sleeps for long time

Fig. 4 Effect of sleep period on battery recovery effect

increase in the node energy consumption, which will be at the expense of decreased delay and the increased throughput. However, if the battery is allowed to sleep for some time portion, it will replenish some of its lost charge, which will help in minimising node energy consumption. This will be at the expense of the decreased throughput and the increased latency. But in fact, allowing the battery to sleep for long time period, will result in a very short duty cycle, which will not allow nodes to do their operations at the current superframe, so nodes will differ transmitting their packets to the next superframe at which they will transmit their packets altogether. This will cause collisions and subsequent transmissions. Consequently, there will be a multiply loss in the battery charges, and there will be extra increase in energy consumption accompanied with a decreased throughput as well as an increased delay in receiving data packets. Therefore, one must be careful about the amount of time the battery is required to sleep for the recovery effect purpose, because long sleep periods leads to unwanted consequences. Hence, before injecting battery recovery factor in improving the IEEE 802.15.4 performance, we need to decide for how long exactly the battery is required to sleep. In other words, we need to analyse the rate at which sleep period can maximise battery lifetime.

To achieve that, in [1] we simulated a star topology of eight nodes (seven clients and one PANc) configured with IEEE 802.15.4 MAC. The supeframe was allowed to offer different active periods (CAP) with different sleep time portions. We applied the Rakhmatov Model to reflect the battery nonlinearity. We added a relaxation time between two packets in order to gain more capacity charge. This was achieved by increasing the value of BO which allowed nodes duty cycle to decrease gradually, each time we analysed nodes residual capacity. Simulation parameters are depicted in Table 1 and the achieved results are depicted in Fig. 5.

It can be noticed from Fig. 5. that, for all the tested active periods apart from 1966.08 ms, increasing the sleep time portion by 50 % (and hence, decreasing the duty cycle) increases the total residual capacity. On the other hand, allowing node sleep more than 50 % decreases the total residual capacity. That is because the sleep time portion allows the chemical charge diffuses around the electrode which enables the battery heal and regain some of its charges because of the principle of the battery recovery effect. Moreover, it can be noticed that the effect of battery recovery increases as the active period increases, this is because by giving a node a longer active time, it will have enough time to do its activities and thus avoid other

Table 1 QualNet 5.2 simulation parameters

Parameter	Value
Physical and MAC model	IEEE 802.15.4
Area	50 m × 50 m
Number of nodes	8
Transmission range	10 m
Simulation time	1000 s
Energy model	MICAZ
Battery type	Duracell AA
Battery model	Rhakhmatov
Traffic, arrival rate	CBR, 1 s
Payload size	50 byte
BO values	2, 3, 6, 7
SO values	2, 3, 4, 6, 7

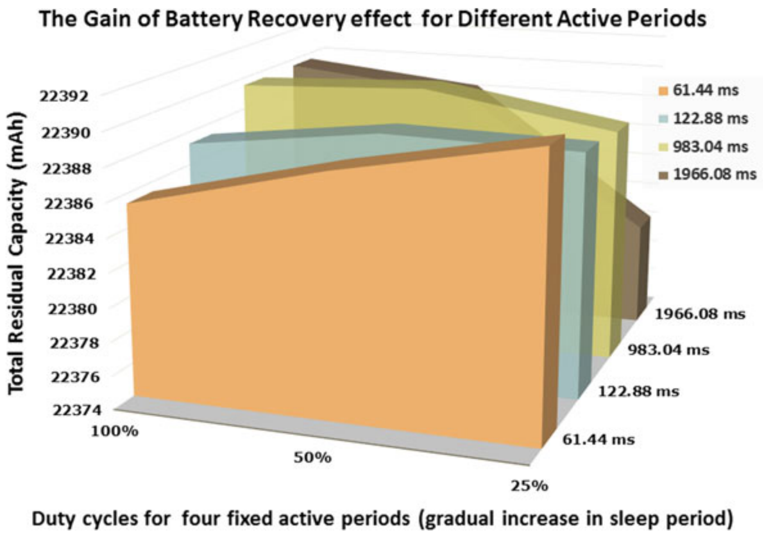


Fig. 5 Residual battery capacity for different duty cycles with different active periods

unnecessary operations such as, retransmission, which will save battery energy due to the increased residual capacity. This explains why for 61.44 ms active period, as the sleep time increases, total battery capacity increases for the three tested duty cycles. For this short period, a node does not have adequate time to perform its activities at all. Instead, it will keep differing its activities to the next superframe. Consequently, as all nodes will try to transmit together, this will cause frequent collisions and retransmissions which adversely affect network performance. It is therefore better for the node to sleep than to stay active [1].

As a conclusion, and based on our conducted topology, the experimental findings show that, in order to exploit battery recovery effect, BO value is needed to be increased only by one as this will allow the node to operate within 50 % duty cycle.

5 The Proposed Technique

Our findings in Sect. 4.2 call for proposing a more Energy-Efficient duty cycling scheme by setting the sleep duration of the coordinator RF transceiver at the saturation threshold of the battery, which can take the maximal advantage of the Duration-Dependent battery recovery effect. In this way, the MAC parameters can be tuned according to the current residual battery capacity. Nevertheless, improving battery performance should not be at the expense of other performance metrics. Packets End-to-End delay can be estimated at higher layers of the protocol stack, while energy consumption and battery behavior are evaluated at the lower layers.

The proposed algorithm: Battery Aware and Reliable Beacon Enabled IEEE 802.15.4 (BARBEI) works as follows: before sending a new beacon frame, PANc asks the physical layer for its total residual capacity. If the new residual capacity is worse than the previous one, to give the battery an opportunity to recover at this time moment, PANc increments the value of BO, otherwise it does nothing. At the same time, PANc also checks the number of received packets at the application layer, for example, if the number is five, and if the new average End-to-End delay is worse than the previous one, then it increments the value of SO, otherwise it does nothing [1]. We notice that while BO values can range from 0 to 14, in our algorithm, BO value is not allowed to exceed the value 8, this is because according to the findings of [7, 8], values of BO higher than 8 result in a very long BI which leads to a dramatic drop in network overall performance. For more details about this, please refer to [7, 8]. Therefore, the overall IEEE 802.15.4 protocol stack is needed to be considered for the MAC layer to adaptively tune its parameters according to the actual needs. In other words, friendly battery technique should be able to adapt to the actual network operating status. Through this approach, both physical and application layers cooperate with MAC layer in order to prolong network lifetime by preserving energy battery charge at the physical layer, while considering average end to end delay status at the application layer.

6 Performance Evaluation

Using QualNet 5.2 Simulator, the performance of the new proposed approach is evaluated by conducting a comparison against the legacy IEEE 802.15.4 performance in terms of total energy consumption, total battery residual capacity, average End-to-End delay and throughput. Evaluation process is applied on a star topology of seven RFDs with 7 Constant Bit Rate (CBR) traffic applications working over 1000 s simulation period.

```
Algorithm: Battery Aware and  Reliable Beacon Enabled
IEEE 802.15.4 (BARBEI)

Objective: Tune MAC supe-frame structure parameters
according to battery nonlinear behavior and network status.
Input: FFD node f, seven RFD nodes $r_1$-$r_7$
Output: New superframe structure with updated BO and SO values
Phase 1: Tune BO value according to f residual capacity.

if    f.send(BEACON) and f.check (RESIDUAL_CAPACITY
        (t(BEACON)) <  RESIDUAL_CAPACITY (t (BEACON-1))))
      and   (BEACON. BO! = 8 )
                 then BO ← BO+1
endif
Phase 2: Tune SO value according to $r_1$-$r_7$ average end to end
delay
for $r_1$ to $r_7$
          if r.check(DATA_PACKETS.num %5)=true
                 then r. calculate (DELAY)
          endif
endfor
if   f.check(AVERAGE_DELAY.new > AVERAGE_DELAY.prevouis)
      and   (BEACON.SO! = BEACON.BO)
                then SO ←SO+1
endif
```

Data rate is fixed for all nodes and the chosen packet interval is 1 s for a 50 bytes packet size. 16 scenarios are tested, each one with different BO: SO combination to cover different duty cycles behaviour. Each time the new algorithm performance is compared against the original IEEE 802.15.4 MAC algorithm [1]. Each case is repeated 10 times. Simulation parameters are the same as those presented in Table 1, but with more BO values considered. The following subsections illustrate the results achieved for the four metrics.

Total Energy Consumption (mWh). According to Fig. 6, it is apparent that the new algorithm decreases energy consumption regardless of the values in BO: SO combination. This is because the new algorithm tunes the MAC BO parameter according to battery residual charge. BO value is incremented if the current residual capacity is less than the previous one allowing the inactive period to increase. This offers node more time to sleep, which in turn allows PANc battery gain some of charge according to battery recovery effect. Consequently, battery capacity

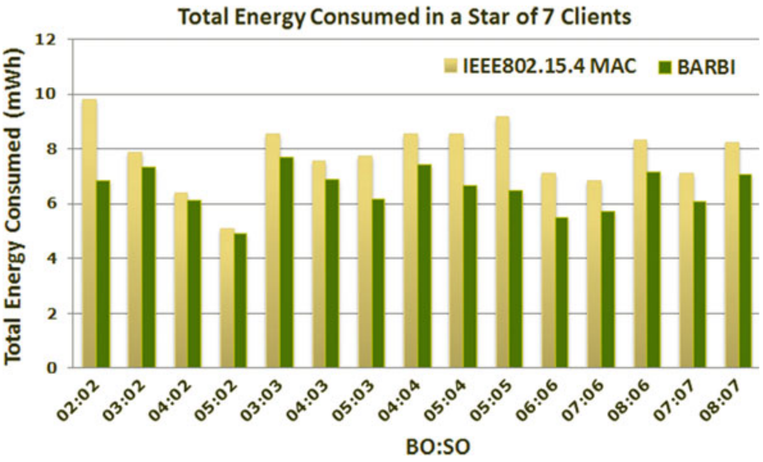


Fig. 6 Total energy consumption for 7 RFDs in a star topology

increases providing more energy according to (6). Moreover, increasing BO decreases beacon overhead which will decrease energy consumption, this effect is obviously noticed in BO: SO combination with small BO values, such as BO = 2. In addition, energy is saved because the new algorithm avoids packets collision and retransmissions as it increases SO according to the application layer status. This gives nodes more time to transmit packets in the increased active period [1].

Residual Battery Capacity (mAh). Figure 7 reveals that despite the duty cycle or BO: SO values, the residual capacity in a network that follows our algorithm is higher than for the network that follows the legacy IEEE 802.15.4 MAC. This is

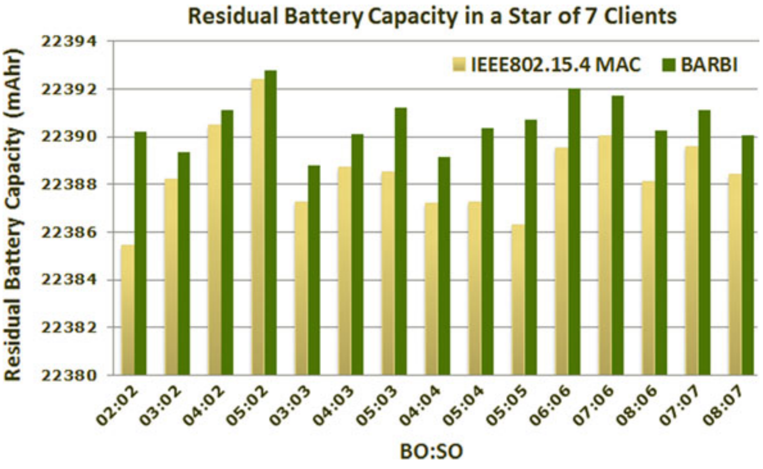


Fig. 7 Residual battery capacity for 7 RFDs in a star topology

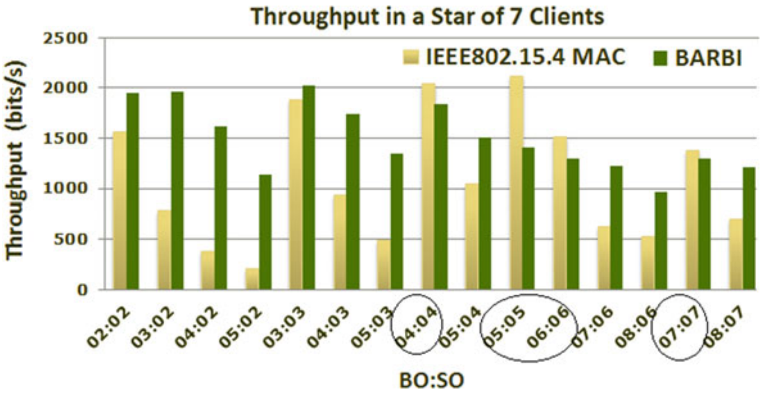


Fig. 8 Throughput for 7 RFDs in a star topology

because PANc exploits battery recovery effect by incrementing BO value according to battery status. This allows PANc battery to heal and gain some of its charge which will increase battery residual capacity. Network nodes are also given more time to sleep as BO increases. This saves residual battery capacity for all network nodes [1].

Throughput (bits/s). Figure 8 depicts that the new algorithm increases the throughput at most of BO:SO values. This is mostly obvious in combinations with small BO: SO values, such as 3:2, 4:2, and 5:2. In these scenarios, following the legacy MAC algorithm, the active period is too short causing a node to differ packet transmission to the next superframe which will cause collision and hence adversely affecting network throughput.

However, as the new algorithm allows the SO values to increase according to network performance, this will give more time for RFDs to complete their packet transmissions successfully, and consequently will improve network throughput. For combinations with 100 % duty cycle, such as 4:4, 5:5 6:6 and 7:7, the legacy MAC outperforms our algorithm. This is because a node in these situations will have full active period to perform its activities which will increase the throughput. However, increasing the inactive period according to battery status lowers the duty cycle which consequently decreases the throughput [1].

Average End-To-End Delay (s). Figure 9 shows that the average End-to-End delay for our algorithm performs better only for combinations with large BO values, such as BO = 6, 7 or 8, because nodes have enough time to do their work and there is no need to increase SO value which avoids the increase of delay. Unfortunately, average End-to-End delay performs worse for the new algorithm for BO: SO combinations with small values such as SO = 2, 3, 4 and 5. For small BO:SO values, the delay is always bad, and there will be frequent increments in BO and SO values allowing node to operate in consequent 50 % duty cycles, which will increase delay [1].

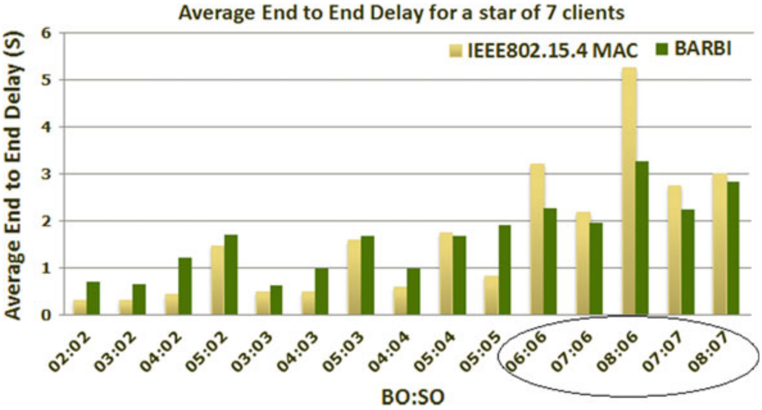


Fig. 9 Average End-to-End delay for 7 RFDs in a star topology

7 Conclusion

The performance of IEEE 802.15.4 standard can be improved by adopting Battery-Friendly algorithms for packets transmission. This can be achieved by designing battery aware duty cycling approaches that exploit battery nonlinearity of recovery effects. However, there is a threshold at which battery recovery effect can be exploited. The proposed adaptive Cross-Layer and battery aware approach improves energy efficiency and power consumption for all possible duty cycle applications that the beacon enabled IEEE 802.15.4 standard offers. For an improved overall performance in terms of energy, average End-to-End delay and throughput, the new approach can be best followed in applications that work in low duty cycle with long active periods. As a future work, not only the PANc is allowed to be aware of its battery behaviour, but also all network nodes will tune their activities according to their residual capacity. This can be achieved by taking the priority as criteria for packets transmission. Node priority will be determined according to its residual capacity. In addition, Cross-Layering facilitates achieving context awareness at higher layers will be proposed.

References

1. Salayma, M., Al-Dubai, A., Romdhani, I., Yassein, M.: Battery aware beacon enabled IEEE 802.15.4: an adaptive and cross-layer approach. In: Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1267–1272. IEEE, Lodz, Poland (2015)
2. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. Commun. Mag. IEEE **40**(8), 102–114 (2002)

3. Selavo, L., Wood, A., Cao, Q., Sookoor, T., Liu, H., Srinivasan, A., Porter, J.: Wireless sensor network for environmental research. In: The 5th International Conference on Embedded Networked Sensor Systems, pp. 103–116. ACM, Sydney, Australia (2007)
4. Jongerden, M.R., Haverkort, B.R.: Battery modeling. Technical report, TR-CTIT-08-01, CTIT (2008)
5. Koubâa, A.: Promoting quality of service in wireless sensor networks. In: Habilitation Qualification in Computer Science, National School of Engineering in Sfax, Tunisia (2011)
6. Ergen, S.C.: ZigBee/IEEE 802.15. 4 Summary. UC Berkeley, 10, 17 Sept (2004)
7. Salayma, M., Mardini, W., Khamayseh, Y., Yassein, M.B.: Optimal Beacon and superframe orders in WSNs. In: The 5th Fifth International Conference on Future Computational Technologies and Applications (IARIA), pp. 49–55. Futurecomputing 2013, Valencia, Spain (2013)
8. Salayma, M., Mardini, W., Khamayseh, Y., Yassein, M.B.: IEEE802. 15.4 performance in various WSNs applications. In: The 7th International Conference on Sensor Technologies and Applications, pp. 103–116. SENSORCOMM 2013, Sydney, Australia (2013)
9. Rakhmatov, D., Vrudhula, S., Wallach, D.: A model for battery lifetime analysis for organizing applications on a pocket computer. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **11**(6), 1019–1030 (2003)
10. Li, H., Yi, C., Li, Y.: Battery-friendly packet transmission algorithms for wireless sensor networks. J. Sensors IEEE **13**(10), 3548–3557 (2013)
11. Linden, D., Reddy, T.B.: Handbook of Batteries (1985)
12. Li, Y., Yin, S., Liu, L., Wei, S., Wang, D.: Battery-aware MAC analytical modeling for extending lifetime of low duty-cycled wireless sensor network. In: The 8th IEEE International Conference on Networking, Architecture and Storage (NAS), pp. 297–301. IEEE (2013)
13. Li, Y., Li, H., Zhang, Y., Qiao, D.: Packet transmission policies for battery operated wireless sensor networks. Front. Comput. Sci. China **4**(3), 365–375 (2010)
14. Chau, C.K., Qin, F., Sayed, S., Wahab, M.H., Yang, Y.: Harnessing battery recovery effect in wireless sensor networks: experiments and analysis. J. Sel. Areas Commun. IEEE **28**(7), 1222–1232 (2010)
15. Chau, T., Wahab, M.H., Qin, F., Wang, Y., Yang, Y.: Battery recovery aware sensor networks. In: The 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, pp. 1–9. IEEE (2009)
16. Casilari, E., Cano-García, J. M., Campos-Garrido, G.: Modeling of current consumption in 802.15. 4/ZigBee sensor motes. Sensors **10**(6), 5443–5468 (2010)
17. Francesco, M. D., Anastasi, G., Conti, M., Das, S.K., Neri, V.: Reliability and energy-efficiency in IEEE 802.15. 4/ZigBee sensor networks: an adaptive and cross-layer approach. J. Sel. Areas Commun. IEEE **29**(8), 1508–1524 (2011)
18. Nasrallah, Y., Guennoun, M., Mouftah, H.T.: Energy-efficient battery-aware MAC protocol for wireless sensor networks. In: IEEE Wireless Communications and Networking Conference (WCNC). IEEE, Paris, France (2012)
19. Khan, J.Y., Yuce, M.R., Harding, B.: Battery Life Cycle and Transmission Power Profile Analysis of a Wireless Body Area Network with Implanted Nodes

A Multisource Energy Harvesting Platform for Wireless Methane Sensor

Saba Akbari, Denis Spirjakin, Vladimir Sleptsov and Alexey Savkin

Abstract Sensors used for detecting combustible gases consume significant amounts of power. Energy management for these sensors can become an important issue when they are used as part of a wireless sensor network. This is because of the fact that wireless sensors are usually powered by batteries. Batteries have a finite lifetime and their replacement can take a considerable amount of time in a gas monitoring application where thousands of sensor nodes are deployed to measure the concentration of flammable gases. Moreover, the battery replacement procedure can turn into a more complicated task if the gas monitoring network is located in a harsh environment. Energy harvesting is a method which can increase the operation time of wireless gas sensor networks. In this article, we present a multisource harvesting circuit for a wireless gas sensor node. As for ambient sources, we have chosen solar and wind energy. Energy from ambient sources is stored in supercapacitors which have a capacity of 400 F. We prove that a catalytic gas sensor can operate for 2 days without batteries by using the developed scheme.

Keywords Multisource energy harvesting · Energy management · Combustible gases · Wireless methane sensor · Catalytic gas sensor

1 Introduction

Fire and toxic gas leakage may have consequences resulting in pecuniary loss or fatality. Hazardous gas monitoring is one of the areas where wireless sensors have been used. However, there are some issues which need to be addressed when applying wireless sensor networks technology to the area of combustible gas monitoring. The devices in wireless sensor networks (WSN) usually use batteries as their power supply [1, 2]. In some applications the battery replacement procedure

S. Akbari (✉) · D. Spirjakin · V. Sleptsov · A. Savkin
Moscow Aviation Institute (National Research University), Moscow, Russia
e-mail: akbarisaba@gmail.com

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_19

can take a considerable amount of time and turn into a more complicated task if the gas monitoring network is located in a harsh environment.

Optical, semiconductor and catalytic sensors are used in wireless sensor networks for combustible gases monitoring [3–5]. Combustible gas sensors consume significant amounts of power [6, 7] and this needs to be taken into account when sensors are used in a WSN.

Tables 1, 2 and 3 illustrate some off-the-shelf components used in a sensor node design. As can be seen from those tables, sensors designed for combustible gas detection consume more power than other components. The operation time of an autonomous wireless sensor device is currently limited by the batteries capacity which is about 3000 mAh, 8000 mAh and 15,000 mAh for the AA, C and D types, respectively. According to the requirements concerning the response time of a gas sensor, it is necessary that the time interval used for combustible gas detection be no more than 120 s [8]. Therefore, it is necessary to optimize the use of batteries.

Table 1 Current consumption of some off-the-shelf gas sensors

Type	Company	Current (mA)
DTK-2 catalytic sensor	NTC-IGD, Russia	55 mA
SGS-21XX semiconductor sensor	Delta, Russia	66.5 mA
TGS2610 semiconductor sensor	FIGARO, Japan	93.3 mA
NAP-66A catalytic sensor	Nemoto, Japan	120 mA
MQ-4 semiconductor sensor	Hanwei Electronics, China	250 mA
Infrared gas sensor	Platform presented in [7]	Sensor current consumption: 125 mA, Lamp current consumption: 115 mA

Table 2 Current consumption of some off-the-shelf transceivers

Type	Company	Current consumption (mA)
CC2500 Transceiver	Texas instruments	Tx: 21.2 mA @ 0 dBm, Rx: 13.3 mA
CC2430 Transceiver	Texas instruments	Tx: 27 mA, Rx: 25 mA
ETRX35x Transceiver	Telegesis	Tx: 31 mA @ +3 dBm, Rx: 25 mA @ 12×10^6 Hz clock speed
TR1000 Transceiver	RF Monolithics	Tx: 12 mA @ 0 dBm, Rx: 3.1 mA
JN5148-001-M00/03 Transceiver	Jennic	Tx: 15 mA @ +2.5 dBm, Rx: 17.5 mA

Table 3 Current consumption of some off-the-shelf microcontrollers

Type	Company	Current consumption (mA)
MSP430F247 Microcontroller	Texas instruments	Active mode: 321 μA @ 3 V/10^6 Hz Low power mode: 1 μA @ 3 V/32,768 Hz
ATmega168P Microcontroller	Atmel	Active mode: 1.8 mA @ 3 V/4 × 10^6 Hz Power-save mode: 0.9 μA @ 3 V/32 kHz
ATxmega32A4 Microcontroller	Atmel	Active mode: 1.1 mA @ 3 V/2 × 10^6 Hz Power-save mode: 0.7 μA @ 3 V/32 × 10^3 Hz
ADuC824 Microcontroller	Analog devices	Active mode: 3 mA @ 3 V/1.5 × 10^6 Hz Power-down mode: 20 μA @ 3 V/32 × 10^3 Hz

A series of methods for reducing energy consumption of wireless sensor nodes have been considered in the literature. These methods include: the technologies used for the fabrication of gas sensors [9], efficient sensing circuit and associated gas measurement procedure [10] as well as data transmission optimization [11]. More detailed information on decreasing power consumption can be found in [12–14].

However, the physical limitations related to the finite capacity of energy storage, e.g., batteries and supercapacitors do not provide the possibility for a complete realization of the concept of ‘perpetual’ operation in wireless sensor networks. At the same time, energy harvesting technology can increase the autonomous operation time of wireless sensor node [13–18]. A compilation of various energy harvesting sources is given in Table 4.

A wireless gas sensor node consists of a microcontroller, transceiver, battery, power management block, energy harvesters and a sensor. However, some gas sensing platforms can contain more than one sensor [19, 20]. This article is an extended version of the paper presented at FedCSIS 2014 [16]. In this article, we present a multisource harvesting circuit for a wireless gas sensor node. As for ambient sources, we have chosen solar and wind energy. Energy from ambient sources is stored in supercapacitors which have a capacity of 400 F. We prove that a catalytic gas sensor can operate for 2 days without batteries by using the developed scheme.

Table 4 Some energy harvesting sources [13, 15, 17]

Energy source	Conditions	Performance
Solar	Outdoors	100 mW/cm^2
Solar	Indoors	100 μW/cm^2
Vibration	1 m/s^2	100 μW/cm^3
RF	Cell phone	0.1 μW/cm^2
Thermal	$\Delta T = 5$ °C	60 μW/cm^2
Wind	Wind (Wind speed: 4.4 m/s, Wind turbine type: Four-blade horizontal axis, diameter: 6.3 cm, Load value ~540 Ω, Electrical power: 7.8 mW)	0.25 mW/cm^2

2 The Proposed Circuit for Multisource Energy Harvesting

The block diagram of the WGSN for methane detection with two harvesters used in our work, is presented in Fig. 1. A more detailed view of the block diagram for analog and digital circuits of the sensor node is illustrated in Fig. 2. Figures 3 and 4 demonstrate the schematic view of the multisource energy harvesting platform.

We have chosen catalytic sensors operating in a Wheatstone bridge circuit [9]. The sensing circuit shown in Fig. 2, consists of an active (R4) and a reference (R5) catalytic sensor. The sensors are arranged in a Wheatstone bridge configuration, where both R1 and R2 have a resistance of 1 kΩ. The resistance of R3 is equal to 1 Ω and is wired in series to the bridge in order to measure the heating current by measuring its voltage drop and applying Ohm's law. The resistance of the active and reference sensors is 12 Ω under normal conditions.

Catalytic sensors are mostly used in the Lower Explosive Limit (LEL) gas concentration range. We use planar catalytic gas sensors produced by NTC-IGD, Russia which has a power consumption of 150 mW [9, 21].

The sensors are fabricated by using the technology of nano-porous gamma anodic alumina membrane with a thickness of 30 μm. Micro-heaters are fabricated

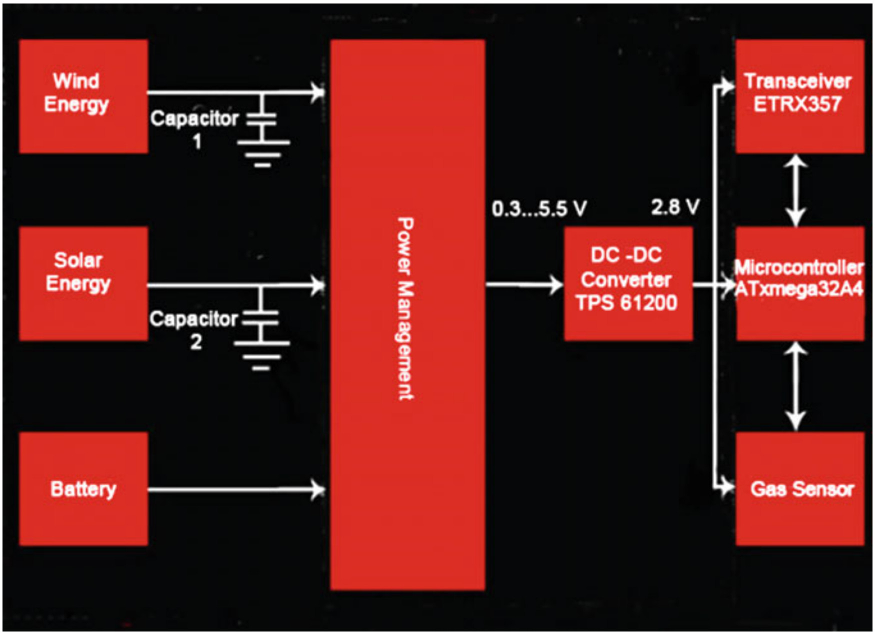


Fig. 1 Block diagram of the wireless gas sensor node (WGSN) for methane detection with a multienergy harvesting mechanism

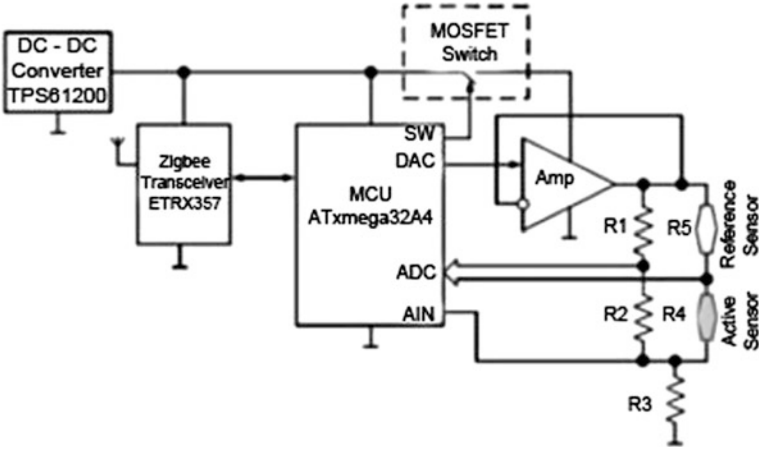


Fig. 2 Schematic view of the wireless gas sensor node

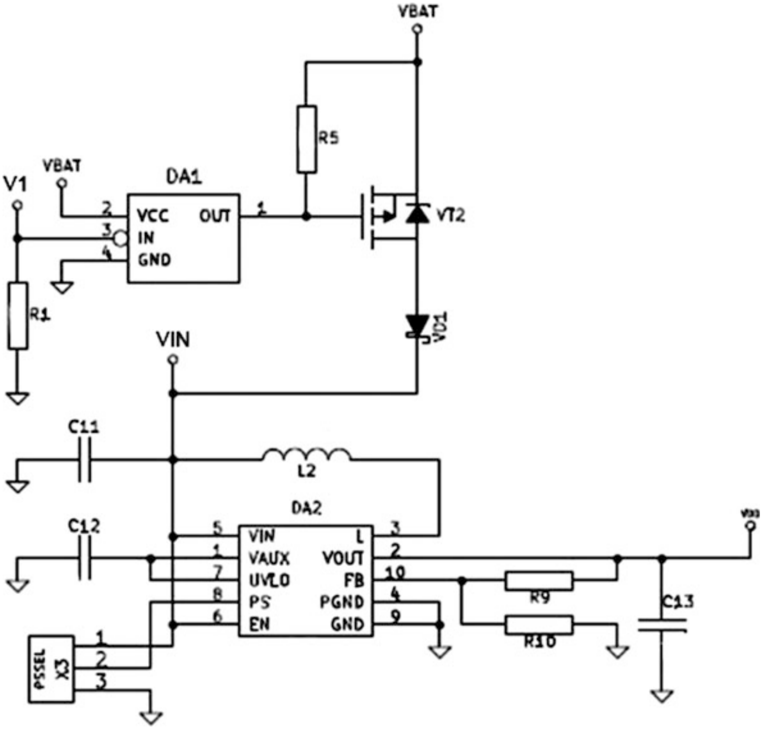


Fig. 3 Schematic view of the DC–DC converter

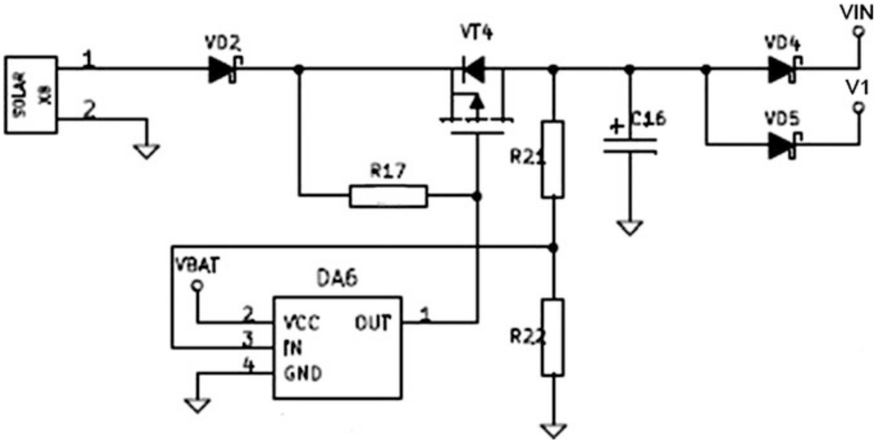


Fig. 4 Schematic view of the solar energy harvester

by magnetron sputtering of a platinum target and covered by a thin film layer of Al_2O_3 to prevent its degradation. The heated area is about 200 × 200 μm^2.

In order to obtain the catalytic activity of the active sensor, the porous gamma alumina membrane is impregnated by catalytic metals (Pd and Pt). The reference sensor has only a micro hotplate covered by a porous gamma alumina oxide material without catalytically active metals and is used to compensate for the environmental factors such as temperature and humidity. The wireless sensor node is based on an ATxmega32A4 microcontroller (MCU). This MCU has been chosen due to its low power consumption (~1 µA in power-save mode), sufficient performance and high-precision 12-bit ADC and DAC.

In order to provide communication across the WGSN, we have used an ETRX357 ZigBee transceiver, which operates in the 2.4 GHz ISM band and has a low sleep current (~0.7 µA).

The wireless gas sensor node is powered by a battery as well as two alternative energy sources (Figs. 1, 3 and 4).

We chose wind and solar energy harvesters in this work since for hazardous gas leak monitoring, for example, in the case of pipelines located above the ground, these sources can be among the first options available.

The solar cell power is approximately 0.77 W at 1.5 AM (short circuit current I_{sc} = 0.35 A and open-circuit voltage U_{oc}= 2.2 V).

At a wind speed of 4.3 m/s, the wind turbine provides a maximum open circuit voltage and short circuit current of 2.2 V and 27 mA. The short circuit current was also measured at wind speeds of 2.5, 3, 3.5, 5, 5.7 and 6 m/s and the amount of current obtained was 3, 6, 15, 24, 27 and 30 mA.

The radius of the blade used in the experiment was 7 cm. The wind speed was measured by using a VOLTCRAFT (BL—30) anemometer and a VOLTCRAFT tachometer (DT—10L) was used for reading the blade's RPM.

Each alternative energy source stores energy in its own supercapacitor, in which case we used two COOPER BUSSMANN—XV3560-2R7407-R—CAP supercapacitors. Each supercapacitor has a nominal capacity of 400 F and a voltage of 2.7 V (The supercapacitor used to store solar energy is marked as C16 in Fig. 4).

Since the main task designated for the gas sensor node is to monitor the methane concentration, it is necessary to provide continuous operation of the sensor node even in cases when the amount of energy delivered by harvesting sources is not enough to power the sensor node.

Therefore, battery is the main source for powering the sensor node. We used a lithium battery (3.6 V) with a capacity of 18,000 mAh. Changing from battery to energy harvesting sources mode takes place when the amount of voltage in supercapacitors is above 900 mV.

Each supercapacitor stores energy independently as a result of which it will be possible to store and use the energy from each source with maximum efficiency. The voltage of supercapacitors shall not exceed the maximum limit, i.e., 2.7 V.

Thus, it is necessary to control the maximum value of each supercapacitor. For the solar harvester, this function is performed by comparator DA6 which can be seen in Fig. 4.

The voltage at the comparator input is compared with the reference voltage (which is 900 mV). In this case, the supercapacitor voltage is divided by a resistive divider (R21–R22) as shown in Fig. 4 so that a voltage of 900 mV in the midpoint of the divider corresponds with the maximum operating voltage of the supercapacitor (2.7 V).

The DC–DC converter (marked as DA2 or TPS 61200 in Figs. 1 and 3) which is integrated in the circuit, operates in the range of 0.3–5.5 V and is used to convert the voltage delivered by the harvesters to the one defined by the scheme (2.8 V). All comparators are powered by battery.

Apart from the methane measurement, the wireless gas sensor nodes also perform self-diagnostics, which includes the monitoring of the batteries voltage level and the sensor heater status.

Both active and reference sensors must be heated up to 450 °C to reach the standby mode temperature at which gas combustion occurs. To meet this requirement, a 2.8 V pulse supply voltage is applied. The heating voltage is adjusted by a built-in Digital-to-Analog Converter (DAC) in the microcontroller and by an output amplifier. The measurement circuit is disabled by a MOSFET switch when it does not perform the sensing of the environment.

Figure 5 shows the sensor current consumption during the heating, measurement and data transmission stages. With the increase of the sensor temperature, the sensor resistance also increases. This, therefore, leads to the decrease of the heating current given that the heating voltage is kept constant. The time necessary to heat the sensor and to enable its operation is approximately 0.7 s.

The response voltage is the voltage between the arms of the Wheatstone circuit (Fig. 2) which is changed in the presence of methane. Then, the MCU processes the gas sensor data, awakes the transceiver and sends the AT command to transmit a unicast to a sink node (the blue portion of Fig. 5).

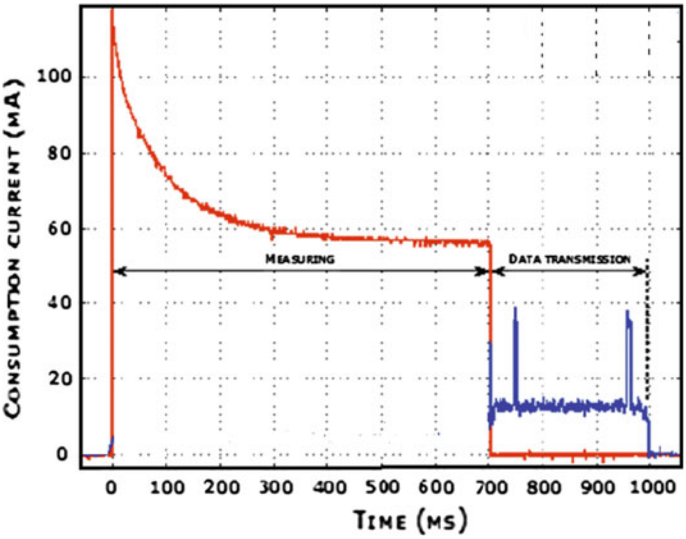


Fig. 5 Current consumption diagram for the wireless gas sensor node

To ensure a successful transmission, the node waits for an acknowledgement and in order to save energy, the measurement is performed only once per 15 s. After a successful transmission, the node immediately is switched to power-save mode.

The current consumption of the node in power-save mode is of several μA. All modules and clocks of the MCU are shut down except for the RTC, which is clocked by an external 32.768 kHz crystal oscillator. The RTC interrupts are used to wake-up the MCU. After waking up, the measurement sequence is repeated. The proposed power circuit algorithm is as follows. At first, both supercapacitors are discharged and the battery powers the sensor node. The voltage of the supercapacitors will increase as the charging process starts. The power management block selects a power supply source for the sensor.

When the voltage across one of the supercapacitors is equal to 900 mV, the sensor node feeding will be switched from battery to a harvesting source, i.e., according to Fig. 3, the value of V1 is compared with the battery voltage (VBAT) at DA1 and the VIN output from a harvester powers the sensor node through the DC–DC converter (which is marked as DA2 in the figure). It is necessary to note that if the average current consumption of the sensor node is less than the one provided by any of the harvesting sources then the supercapacitor will be kept charging until it reaches the maximum voltage.

If the average current of the sensor node is greater than the current provided by the harvesting sources, the supercapacitor will start discharging and the voltage across the supercapacitor decreases.

At voltages less than 900 mV, the sensor node powering mechanism is switched to battery, i.e., according to Fig. 3, the value of V1 is compared with the battery voltage (VBAT) at DA1 and the VBAT output from battery powers the sensor node through the DC–DC converter (which is marked as DA2 in the figure).

In order to provide a stable operation for the sensor node by using an energy harvesting scheme, it is necessary that the power generated by the source be greater than the power consumption of the sensor node. Typically the WGSN is part of a wireless sensor network that includes a coordinator, sensor nodes, actuators and relay nodes.

Wireless gas sensor node measures the concentration of gas and transmits data or command to a coordinator or actuator. In some cases, data transmission can be carried out via relay nodes or other sensor nodes. Since the algorithms of wireless sensor network operation are quite different, we consider a simple network in which coordinator and the WGSN communicate directly. For methane, the lower limit of its explosion is 4.4 % volume. The hazardous situation occurs when methane concentration is higher than 1 % volume. Therefore, the wireless gas sensor node is operating in the following way. Two thresholds concerning the methane detection are defined for the operation of the WGSN. The thresholds can be changed depending on the application being considered. If the methane concentration is less than 0.5 % volume and the measurement process is over, the WGSN goes to sleep mode. If the WGSN has detected methane concentration in the range of 0.5–1.0 % volume, the microcontroller changes the ZigBee module over to transmission mode, generates an alert message at the lower threshold and transmits it to the network coordinator.

As soon as the acknowledgement has been received, the WGSN goes to sleep mode. If the WGSN has detected methane in the range of above 1.0 % volume, the sensor node notifies the coordinator and, for example, sends an alert message directly to the wireless actuator to close a valve (or turn on the ventilation, alarm and so on) [22].

To save energy, the WGSN operates in a periodic mode. The methane measurements are carried out each 15 s, however, this interval could also be varied according to the application. The rest of the time, the WGSN remains in sleep mode.

3 Power Consumption and Operation Time Estimation

3.1 Operation Time Estimation Using a Battery

As can be seen from Tables 1, 2 and 3 the gas sensor is the main energy consumer. The second important consumer is the transceiver. However, the transceiver is in active mode in case of emergency situations only. The other time it is in sleep mode.

A voltage of 2.8 V is mainly used to heat the sensors up to 450 °C, which is the operational temperature. The heating time for the sensors in the Wheatstone circuit is about 0.7 s [10]. The average constant current flowing through during heating is 65 mA that results in about 180 mW of power consumption in continuous measurement mode. We estimate the battery lifetime for gas measurement, which is carried out once per 15 s. Excluding data transmission, which takes place only in case of emergency, the estimated operation time of the wireless gas sensor node is (3.6 V × 18,000 mAh)/(180 mW × 0.7/15 s) = 7714 h (or 321 days). In case of data transmission, the estimated operation time of the wireless gas sensor node will decrease to 7013 h (292 days).

3.2 Operation Time Estimation Using Harvesting Sources

The energy stored in a supercapacitor is given by the following equation:

$$\mathrm{W} = \frac{\mathrm{C}}{2} \cdot \left(\mathrm{V}_{\max}^{2} - \mathrm{V}_{\min}^{2}\right) \tag{1}$$

where $V_{m\mathrm{ax}}$, V_{min} are the maximum (2.2 V) and minimum (0.9 V) values for voltage, and C is the capacity. The result is 806 J or 0.22 Wh.

It can be observed that the power stored in the supercapacitor is almost 295 times less than the lithium battery power, which is almost 65 Wh. However, the energy of the supercapacitor, which is completely charged, is enough for the operation of the wireless gas sensor for 43 and 39 h in case of excluding and including the data transmission stages. Since two supercapacitors are used, the operation time of the WGSN based on energy harvesting will be 86 h.

Figure 6 shows the theoretical and experimental graphs indicating the supercapacitor energy and voltage during the wireless gas sensor node operation provided that the supercapacitor used is completely charged (i.e., 1300 J, 400 F, 2.7 V). According to Fig. 6, the supercapacitor discharge takes approximately 28 h, which is much faster comparing with the theoretical time of 43 h.

This result is due to the actual losses that occur in the conversion of energy, particularly the losses in the DC–DC converter. DC–DC converters have a high efficiency (which is more than 90 %) in case of delivering lower amounts of voltage, but when it is necessary to output higher voltage values, the efficiency of the DC–DC converter deteriorates and this is the issue, which has happened in our case.

It is necessary to note that the maximum voltage of the solar battery and wind generator in this work is less than the maximum amount which can be reached by the supercapacitor (2.7 V). Because of this, both supercapacitors were not fully charged in our case and the operation time of the WGSN will be less.

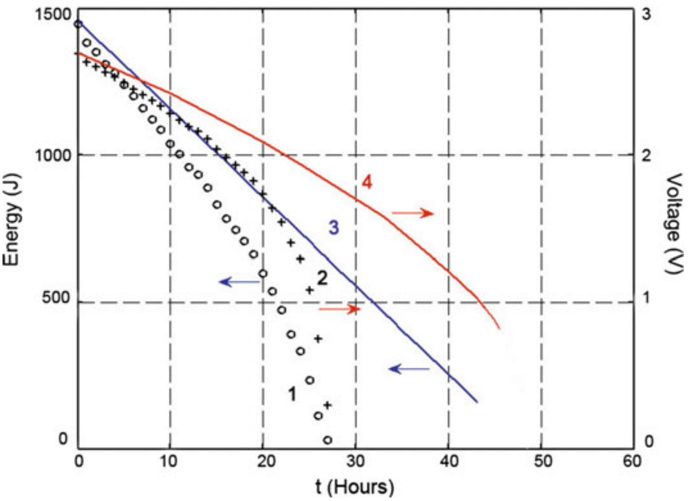


Fig. 6 Theoretical (the *solid line* and *curve*) and experimental graphs (marked by o and +) indicating the supercapacitor energy (*curve 1* and *line 3*) and voltage (*curves 2* and *4*) during the wireless sensor gas node operation

But this is not a significant limitation in comparison with the energy loss during the voltage conversion. The supercapacitor charging time (T) can be evaluated by Eq. 2.

$$T = \frac{C \cdot V}{I} \tag{2}$$

where C is the capacity, V is the charge voltage and I is the charge current. For example, it takes 1080 s (18 min) in order to charge a 400 F supercapacitor to 2.7 V with a 1A current.

Since it is unlikely that solar and wind sources will be operating at their peak powers and taking into account the fact that the open circuit voltage associated with the solar panel and wind turbine is 2.2 V, the charging time of the supercapacitor will increase. With a current of 0.1 A, this time will be 2.5 h in order to charge the supercapacitor designated for the solar harvester. In any case, it is reasonable to store energy during the day to provide power for the operation of the WGSN at night. As the maximum current provided by the small wind turbine is 27 mA which is less than the one provided by its solar counterpart, the charging of the supercapacitor will take more time.

However, the wind blows regardless of the time of the day. If the solar or wind harvesters cannot not provide enough energy the WGSN will operate on the lithium battery. Figure 7 illustrates the devices used during the experiment.

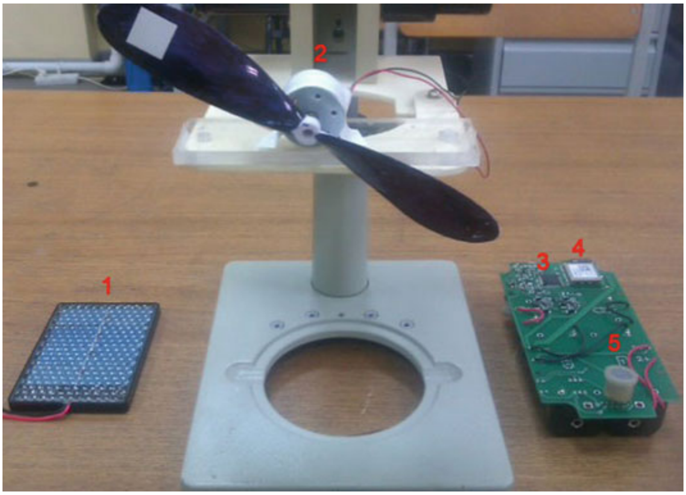


Fig. 7 Devices used during the experiment: *1*. Solar panel *2*. Wind turbine *3*. Microcontroller *4*. Transceiver *5*. Gas sensor

4 Applications

In a large distribution system such as a pipeline infrastructure, employing energy harvesting techniques for powering wireless gas sensors can be a realistic scenario since batteries may not guarantee a continuous sensor operation and this is associated with the their limited lifetime. Moreover, if the system consists of a large number of sensors then replacing batteries can complicate the monitoring process. It is necessary to note that solar energy can be the most abundant source available at the monitoring site. Therefore, implementing maximum power point tracking (MPPT) algorithms for the solar panel can optimize the harvesting process and consequently deliver more power to the senor node.

5 Conclusion

In this article, we presented the circuit design of a wireless gas sensor for CH_4 monitoring supplied with a multisource energy harvesting platform. In order to increase energy efficiency, the sensor conducts measurements once per 15 s. The energy harvesting block provides energy from two ambient sources including solar and wind energy. Each harvesting source stores energy in its own supercapacitor. Whenever the voltage level across any of the supercapacitors reaches 900 mV, the sensor operation will be switched from battery to a harvesting source. It has been also indicated that if each supercapacitor is fully charged; i.e.; if a voltage of 2.7 is achieved, the sensor autonomous operation will be about 43 h and since there are two supercapcitors used, this number turns into 86 h. The charging time of the supercapacitor associated with the solar panel is estimated to be around 18 min, if the charging voltage and the current provided are 2.7 V and 1 A respectively. As mentioned earlier, the maximum current generated by the solar panel in this work is 0.35 A which means that it takes 42 min to charge the supercapacitor. The current obtained at a wind speed of 4.3 m/s was 27 mA and the time needed to charge the supercapacitor connected to the wind harvester is around 9 h.

In case of insufficient energy from ambient sources, the sensor will be powered from battery. If the data transmission stage is excluded, the sensor operation based on battery will be equal to 7714 h which corresponds to 321 days. If data transmission is taken into account, the sensor lifetime will be equal to 7013 h and it means the sensor can operate for 292 days.

Therefore, the proposed energy harvesting circuit can extend the operation time of the sensor node.

Acknowledgements The authors would like to thank Professor Alexander Baranov from Moscow Aviation Institute (National Research University) for his assistance and support in preparing the manuscript. This work was supported by grant No. RFMEFI57714X0133 from the Ministry of Education and Science of Russian Federation.

References

1. Uzoh, P.C., Li, J., Cao, Zh., Kim, J., Nadeem, A., Han, K.: Energy efficient sleep scheduling for wireless sensor networks. In: Wang. G., Zomaya, G., Perez, G.M., Li, K. (eds.) Algorithms and Architectures for Parallel Processing, vol. 9528, pp. 430–444. Springer International Publishing (2015)
2. Somov, A., Baranov, A., Spirjakin, D., Spirjakin, A., Sleptsov, V., Passerone, R.: Deployment and evaluation of a wireless sensor network for methane leak detection. Sensors Actuators A **202**, 217–225 (2013)
3. Brunelli, D., Rossi, M.: Enhancing lifetime of WSN for natural gas leakages detection. Microelectron. J. **45**, 1665–1670 (2014)
4. Samotaev, N.N., Vasiliev, A.A., Podlepetsky, B.I., Sokolov, A.V., Pisliakov, A.V.: The mechanism of the formation of selective response of semiconductor gas sensor in mixture of $CH_4/H_2/CO$ with air. Sensors Actuators B: Chem. **127**, 242–247 (2007)
5. Somov, A., Suchkov, A., Karelin, A., Mironov, S., Baranov, A., Karpova, E.: Compact low power wireless gas sensor node with thermo compensation for ubiquitous deployment. IEEE Trans. Ind. Inf. **11**, 1660–1670 (2015)
6. Spirjakin, D., Baranov, A., Sleptsov, V.: Design of smart dust sensor node for combustible gas leakage monitoring. In: Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1279–1283 (2015)
7. Makeenkov, A., Lapitskiy, I., Somov, A., Baranov, A.: Flammable gases and vapors of flammable liquids: monitoring with infrared sensor node. Sensors Actuators B: Chem. **209**, 1102–1107 (2015)
8. British Standard Institution Staff: Electrical Apparatus for the Detection of Combustible Gases in Domestic Premises. Test Methods and Performance Requirements. British Standard Institution (2000)

9. Karpov, E.E., Karpov, E.F., Suchkov, A., Mironov, S., Baranov, A., Sleptsov, V., Calliari, L.: Energy efficient planar catalytic sensor for methane measurement. Sensors Actuators A **194**, 176–180 (2013)
10. Somov, A., Baranov, A., Spirjakin, D., Passerone, R.: Circuit design and power consumption analysis of wireless gas sensor nodes: one-sensor versus two-sensor approach. IEEE Sensors J. **14**, 2056–2063 (2014)
11. Kumar, A., Hancke, G.P.: Energy efficient environment monitoring system based on the IEEE 802.15.4 standard for low cost requirements. IEEE Sensors J. **14**, 2557–2566 (2014)
12. Baranov, A., Spirjakin, D., Akbari, S., Somov, A.: Optimization of power consumption for gas sensor nodes: a survey. Sensors Actuators A **223**, 279–289 (2015)
13. Magno, M., Boyle, D., Brunelli, D., O'Flynn, B., Popovici, E., Benini, L.: Extended wireless monitoring through intelligent hybrid energy supply. IEEE Trans. Ind. Electron. **61**, 1871–1881 (2014)
14. Zahid Kausar, A.S.M., Reza, A.W., Saleh, M.U., Ramiah, H.: Energizing wireless sensor networks by energy harvesting systems: scopes, challenges and approaches. Renew. Sustain. Energy Rev. **38**, 973–989 (2014)
15. Vullers, R.J.M., van Schaijka, R., Doms, I., Van Hoof, C., Mertens, R.: Micropower energy harvesting. Solid-State Electron. **53**, 684–693 (2009)
16. Akbari, S.: Energy harvesting for wireless sensor networks review. In: Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 987–992 (2014)
17. ÓMathúna, C., O'Donnell, T., Martinez-Catala, R.V., Rohan, J., O'Flynn, B.: Energy scavenging for long-term deployable wireless sensor networks. Talanta **75**, 613–623 (2008)
18. Baranov, A., Spirjakin, D., Akbari, S., Somov, A., Passerone, R.: POCO: 'Perpetual' operation of CO wireless sensor node with hybrid power supply. Sensors Actuators A **238**, 112–121 (2016)
19. Samotaev, N.N., Ivanova, A.V., Oblov, K.Yu., Vasiliev, A.A.: Wireless digital platform for environmental gas monitoring. In: 2015 International Siberian Conference on Control and Communications (SIBCON 2015), pp. 1–4 (2015)
20. Samotaev, N., Ivanova, A., Oblov, K., Soloviev, S., Vasiliev, A.: Wi-Fi wireless digital sensor matrix for environmental gas monitoring. Proc. Eng. **87**, 1294–1297 (2014)
21. Somov, A., Baranov, A., Suchkov, A., Karelin, A., Mironov, S., Karpova, E.: Improving interoperability of catalytic sensors. Sensors Actuators B: Chem. **221**, 1156–1161 (2015)
22. Somov, A., Baranov, A., Spirjakin, D.: A wireless sensor-actuator system for hazardous gases detection and control. Sensors Actuators A **210**, 157–164 (2014)

Author Index

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent
Systems and Computing 461, DOI 10.1007/978-3-319-44354-6